ON BOUNDED RATIONALITY AND COMPUTATIONAL COMPLEXITY

Christos H. Papadimitriou[1] **and Mihalis Yannakakis**[2]

ABSTRACT: *It has been hoped that computational approaches can help resolve some well-known paradoxes in game theory. We prove that if the repeated prisoner's dilemma is played by finite automata with less than exponentially (in the number of rounds) many states, then cooperation can be achieved in equilibrium (while with exponentially many states, defection is the only equilibrium). We furthermore prove a generalization to arbitrary games and all Pareto optimal strategy pairs within the pure individually rational region.*

1. INTRODUCTION

The theory of games [AH], founded by von Neumann and Morgenstern, is supposed to model the behavior of rational economic agents. Very often, however, game theory predicts behavior that can be criticized as unnatural and nonrational —we describe a famous such example, the *prisoner's dilemma*, below. It has been hoped that the situation can be remedied if the model is modified to take into account appropriate notions of complexity. This work is a contribution to this line of research. To describe and motivate the results, we shall first need some definitions and basic facts from game theory.

Background

A *game (in strategic form)* between two players I and II consists of two sets of strategies X and Y, and two functions g_{I} and g_{II} (the *payoffs*), each mapping $X \times Y$ to the integers (the main results apply to real payoffs as well). If I selects strategy $x \in X$ and II selects $y \in Y$, then I receives a payoff of $g_{\mathrm{I}}(x, y)$, while II receives $g_{\mathrm{II}}(x, y)$.

Figure 1. The prisoner's dilemma and matching pennies.

Two simple but subtle games are shown in Figure 1; the rows are the strategies of I, the columns are the strategies of II, and the (x, y)th entry of the table is the pair $(g_{\mathrm{I}}(x, y), g_{\mathrm{II}}(x, y))$. The first game is *prisoner's dilemma*, a famous example proposed by

[1] Department of Computer Science and Engineering, University of California at Berkeley. Research supported by the National Science Foundation.

[2] Bell Laboratories, Murray Hill, NJ 07974.

A. Tucker in the 1940s. I and II are in this case prisoners accused of corruption. Each can either *collaborate* (stay loyal to the other prisoner), or *defect* (agree to testify against the other). The payoffs are consistent with the intuitive outcomes of such a situation. (For a merrier interpretation, imagine that I and II are duopolistic enterprises, strategy C keeps prices high, and strategy D lowers the prices.) The second game in Figure 1 is the game of *matching pennies*. I and II each choose heads or tails; I wins if the outcomes match, otherwise II wins. Notice that this game is *zero-sum*, in that II collects what I loses, and vice-versa; the prisoner's dilemma is not zero-sum.

Call a pair of strategies (x, y) a *pure (Nash) equilibrium* if neither player has an incentive to move away from this strategy; that is, if it holds that for all $x' \in X$, $g_{\mathrm{I}}(x', y) \leq g_{\mathrm{I}}(x, y)$, and also for all $y' \in Y$, $g_{\mathrm{II}}(x, y') \leq g_{\mathrm{II}}(x, y)$. For example, in the prisoner's dilemma, (D, D) is a pure equilibrium, while matching pennies has no pure equilibrium. Intuitively, a pure equilibrium is one possible formulation of rational behavior in this context; it is thus rather disappointing (and quite alarming about the value of games as realistic models of economic behavior) that certain natural games have no equilibrium. We can remedy the situation by an ingenious maneuver: Define a *mixed strategy* for I to be a *probability distribution* π over X, and similarly for Y. If (π, ρ) is a pair of mixed strategies for the two players, the payoff $g_{\mathrm{I}}(\pi, \rho)$ for player I is $g_{\mathrm{I}}(\pi, \rho) = \mathcal{E}_{\pi,\rho} g_{\mathrm{I}}(x, y)$, where $\mathcal{E}$ stands for expectation; similarly $g_{\mathrm{II}}(\pi, \rho) = \mathcal{E}_{\pi,\rho} g_{\mathrm{II}}(x, y)$. A pair (π, ρ) of mixed strategies is an *equilibrium* if for all mixed strategies π' for I, $g_{\mathrm{I}}(\pi', \rho) \leq g_{\mathrm{I}}(\pi, \rho)$, and similarly for II. One of the most fundamental facts in game theory is that *every (finite) game has a mixed equilibrium* [Na]. In the case of zero-sum games, this fact is tantamount to linear programming duality; for the general case the proof relies on a more sophisticated fact, *Brouwer's fixpoint theorem*. For example, matching pennies has the mixed equilibrium $(\pi, \rho) = ((\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2}))$ —*flipping the pennies!*

It is easy to see that the prisoner's dilemma has *only one* equilibrium, (D, D) —this follows from the fact that, for both players, *D strictly dominates C*, in that D is better than *C independently of the other player*. It follows that only (D, D) can be an equilibrium —and it is. In other words, game theory predicts that in such a situation any two rational players must defect, even though this results in smaller payoff to each one of them than if they had both collaborated. This contradicts social experience, intuition, and actual experiments [Ax], and it presents a rather bleak view of society. In real life we do not always behave in a selfishly antisocial way, and we often give up an advantage in order to behave in a cooperative manner. Much work in game theory has been devoted to explaining this apparent paradox.

One argument could start as follows: In real life we do not play one-shot games such as these; instead, we play *repeated* games. Let $n > 1$. The *n-round prisoner's dilemma* is a game which, intuitively, entails playing the prisoner's dilemma n times in a row. The strategies are functions mapping strings over the alphabet $\{C, D\}$ of length less than n to $\{C, D\}$, and the payoffs are calculated by playing out the game and adding the payoffs at each round; the result is then divided by n. What are the equilibria of this game? Unfortunately the only equilibrium is (D^n, D^n); that is, the only rational behavior is to defect all the time! Any pair of strategies in which during a play of nonzero probability one player fails to defect in the last round has smaller payoff to that player than the perturbation in which defection occurs. Hence each player defects in the last round. The same holds for the $n-1$st round, and so on down to the first. Repetition does not seem to resolve the paradox[1]. In fact, the same holds not only in the repeated prisoner's dilemma, but in the repeated version of any game in which all equilibria coincide with the threat point (see Section 3).

The n-round prisoner's dilemma is an interesting kind of game, susceptible to complexity analysis. It has a strategy space which is *doubly exponential in n* (recall that a strategy is a function that maps strings in $\{C, D\}^{\leq n}$ to $\{C, D\}$); arguably, no realistic agent has access to so many behaviors. The undesirable equilibrium is arrived at by arguing in terms of strategies that are arbitrarily complex. Perhaps in a more realistic setting, in which the players can employ strategies that are in some sense simple, collaborative behavior is not ruled out. The question is this: *What are appropriate notions of complexity of strategies, such that, by ruling out strategies of very high complexity, collaborative behavior emerges?*

This is a special case of an influential idea due to Herbert Simon [Si], the principle of *bounded rationality.* In its most usual form, it can be paraphrased thus: *Economic theory sometimes predicts that rational agents will reason and behave in complex and devious ways in order to extract a little more payoff, destroying the system in the process. Such behaviors are ruled out if one assumes that reasoning and computation are costly, and therefore agents do not invest inordinate amounts of computational resources and reasoning power to achieve tiny payoffs.* Obviously, bounded rationality is an invitation for applying the concepts and techniques of complexity theory to economics and game theory; indeed, there has been some interesting recent research in this direction [BP, DP, FW, GKZ, GZ, KM, Pa].

A natural measure of the complexity of a strategy for the n-round prisoner's dilemma and other games is the amount of memory needed, perhaps measured in terms of the number of *states* in a finite automaton implementing the strategy. An automaton for a player P (I or II) is a 6-tuple $A = (Q, I, O, \delta, \lambda, q_0)$, where Q is the finite set of states, q_0 is the initial state, I is the set of inputs which in this context is the set of (one-shot pure) strategies of the opponent player, O is the set of outputs which here is the set of (one-shot pure) strategies of player P, $\delta : Q \times I \to Q$ is the transition function and $\lambda : Q \to O$ is the output function. The player starts at the initial state q_0. In each round, the player plays the strategy $\lambda(q)$ where q is the current state, and then moves to state $\delta(q, a)$ where a is the strategy played by the opponent in this round. As is customary, the automaton can be represented graphically by its transition diagram: a labelled graph which has one node for each state labelled by the corresponding output, and has one arc for each transition labelled by the corresponding input. The initial state is indicated by a dangling arc (an arc without a head). For example, the two-state automaton *tit-for-tat*, a reasonable strategy that actually fares well in tournaments [Ax], is shown in Figure 2. For more background on automata we refer to a textbook [HU] or the survey [Ka].

[1] Collaboration does prevail in the following three other versions of the repeated prisoner's dilemma: (a) the infinite game (see Section 3); (b) the infinite discounted game; and (c) the finite repetition with unknown, or randomly determined, number of rounds; see [AH] for more information and references.

Figure 2: Tit for tat

A variant of the n-round prisoner's dilemma now emerges: Suppose that the players' strategies are in fact restricted to be finite automata of size $s_{\mathrm{I}}(n)$ and $s_{\mathrm{II}}(n)$, where $s_{\mathrm{I}}(n), s_{\mathrm{II}}(n) \geq 2$ are given functions. The notion of an equilibrium is calculated now with respect to the restricted strategy space; that is, a pair (x, y) of automata obeying the size bounds (or mixed strategies of such automata) is an equilibrium if for any other automaton x' for I of size at most $s_{\mathrm{I}}(n)$, we have $g_{\mathrm{I}}(x', y) \leq g_{\mathrm{I}}(x, y)$ and similarly for II. It is possible that for certain bounds $s_{\mathrm{I}}(n)$ and $s_{\mathrm{II}}(n)$ there are equilibria other than the ominous (D^n, D^n). For example, when $2 \leq s_{\mathrm{I}}(n), s_{\mathrm{II}}(n) < n-1$, by the Myhill-Nerode Theorem [HU] the two automata cannot count up to n, and so the backwards induction argument above is not valid —in fact, it is not very hard to prove that *two automata playing tit-for-tat are in equilibrium* among automata with less than $n-1$ states (see Lemma 4). Bounded rationality seems to indeed foster collaboration, albeit with unreasonably restrictive complexity bounds. A. Neyman was the first to study bounds beyond n on the sizes of automata for which collaboration is enabled. An important theorem announced in 1985 [Ne1] states that if both size bounds s_{I} and s_{II} lie in a range $[n^{\frac{1}{k}}, n^k]$ for some $k > 1$, that is, between a root and a power of n, then collaboration can be approximated in equilibrium. (Section 2 contains the precise statement). A similar result for *Turing machines*, conditional on an automata-theoretic conjecture, was proved in [MW].

Summary of results

Given that bounded automata may lead to collaboration, a most interesting question now is the following: *Which bounds s_{I} and s_{II} on the number of states of the players encourage collaborative behavior?* That is, what are the rates of growth of the state space of automata such that there is an equilibrium whose payoff to the two players is close to that of collaboration (3,3). Our first result answers this question giving an almost complete characterization of these bounds (Section 2). We point out that, if $s_{\mathrm{I}}(n), s_{\mathrm{II}}(n) \geq 2^n$, then backwards induction is restored in a rather sophisticated way involving dynamic programming, and collaboration is again ruled out. In contrast, *for all subexponential complexities there are equilibria that are arbitrarily close to the collaborative behavior.*

Theorem 1: If at least one of the state bounds in the n-round prisoner's dilemma is[1] $2^{O(\epsilon n)}$, then, for large enough n, there is a (mixed) equilibrium with average payoff for each player at least $3 - \epsilon$. $\square$

Theorem 1 is quite surprising, since there are many more states than rounds, and therefore backwards induction seems to be enabled. Furthermore, note that one player could be allowed much larger automata than its opponent —more than exponential, and even unbounded. The equilibria for given bounds on the sizes are quite "fragile" in the sense that they can be defeated if one uses only slightly larger automata (and this is

[1] See Section 2 for the precise bound on the exponent. Recall that the big-O notation means that the exponent is bounded by $c\epsilon n$ where c is a constant. For general games beyond the prisoner's dilemma, c is a constant that depends on the game and the given playoff combination that we wish to approximate, but not on ϵ or n.

inherent to the problem); that is if the pair of strategies (x, y) is an equilibrium for sizes, e.g. $s_{\mathrm{I}}(n) = s_{\mathrm{II}}(n) = 2^{n/10}$, then there is an automaton with $2^{n/10} + n$ states that achieves higher payoff against y than x does. A consequence of this fact is that in order to prove that a certain pair is an equilibrium, we have to account for essentially every single state allowed by the bounds and make the most use of it. This causes a number of subtle difficulties, which we describe in more detail in Section 2 and explain how our construction overcomes them.

The payoffs of the different pairs of strategies of a game can be viewed geometrically, as shown in Figure 3 for prisoner's dilemma; the two axes correspond to the payoffs of the two players. The convex hull of the points corresponding to pure strategies contains the possible payoffs for mixed strategies, and is known as the *feasible region.* The upper boundary of this region (DC-CC-CD in the figure) consists of the *Pareto optimal* points, those that are not dominated by any other points, i.e., every other point gives smaller payoff to at least one of the two players. The point DD in the figure is known as the *threat point*; it is the point (v_1, v_2) such that each player can guarantee payoff v_i for any strategy of the opponent (we define this more precisely in Section 3). The part of the feasible region that dominates the threat point (the quadrilateral A-CC-B-DD in this case) is called the *individually rational region*, so called because obviously no player will settle for a payoff less than v_i which it can always achieve.

Figure 3: The geometry of the payoffs.

In Section 3 we generalize Theorem 1 to arbitrary games and payoff combinations. For any combination of pure strategies in the individually rational region we prove an immediate generalization of Theorem 1; the only weakening is that the constant in the exponent hidden by the big-O notation is now much smaller (and depends on the game and the payoff combination), due to the more sophisticated construction of the appropriate equilibria. We further extend this to certain families of pairs of mixed strategies: Those in the Pareto boundary, and some others that we call *independently individually rational.* We must weaken the result now by requiring that *both* bounds be subexponential. There are certain subtle technical obstacles that prevent the generalization to all of the individually rational region. We also prove a result, along more traditional and familiar lines, giving a precise characterization of games whose *infinite* repetition has a pure equilibrium consisting of (unbounded) automata.

A preliminary version of the present paper, with a proof of Theorem 1 and proof sketches of the generalizations in Section 3, appeared in the spring of 1994 [PY]. In the winter of 1995 A. Neyman kindly communicated to the authors a manuscript [Ne2] with a proof of Theorem 1, quite different from the present one, and a clever argument which generalizes our results in Section 3 to all of the individually rational region.

2. PRISONERS' DILEMMA

We consider the repeated prisoners' dilemma with n rounds, where Player I (resp. II) is restricted to automata of size $s_{\mathrm{I}}(n)$ (resp. $s_{\mathrm{II}}(n)$). A player may use a mixed strategy, i.e., a probability distribution on all automata obeying the state bounds.

Our first result deals with the case of exponentially strong players:

Proposition 1. If *both* size bounds $s_{\mathrm{I}}(n), s_{\mathrm{II}}(n)$ are at least 2^{n-1}, then the only equilibrium, is the one in which both players defect in all rounds.

Proof: Consider an equilibrium pair of mixed strategies, A_{I} and A_{II}, of the n-round prisoner's dilemma, under the given bounds. We shall describe now the optimum response (unrestricted) strategy of II against A_{I}, call it $\mathrm{opt}_{\mathrm{II}}$. II does not know which of the automata in the mixed strategy A_{I} it is playing against, but it knows the precise distribution A_{I}, and also at time i it knows the opponent's play up to time $i-1$; so it must treat the situation as a partially observable random process (or, decision-making under incomplete information). $\mathrm{opt}_{\mathrm{II}}$ is a decision tree of depth n, the number of rounds. At the ith level, $i = 1, \ldots, n-1$, it has a node for each string in $\{C, D\}^{i-1}$, corresponding to the observations of I's behavior until now. At each such node, $\mathrm{opt}_{\mathrm{II}}$ must make the optimum decision (C or D). However this is easy to do by dynamic programming bottom-up: if we know the optimum decisions for the future steps, then we can calculate (using the distribution of A_{I}) the expected payoff of each decision at the ith step, and choose the best one. To begin the bottom-up computation, at the nth stage, of course, there is no decision to make: $\mathrm{opt}_{\mathrm{II}}$ defects on positive probability histories (hence there needs to be only one node at the nth level). This concludes the description of $\mathrm{opt}_{\mathrm{II}}$.

Notice that the optimum unrestricted response strategy can be implemented by a (single) automaton with at most 2^{n-1} states: the state transition diagram is a complete binary tree with $n-1$ levels along with an additional single node at level n. There may be more than one optimal automata, if there are ties at some nodes for the choice of the best decision, but note that any optimal automaton plays D in the last stage for every history of nonzero probability.

Since A_{I} and A_{II} were supposed to be an equilibrium pair within the $s_{\mathrm{I}}(n), s_{\mathrm{II}}(n)$ bounds, and since both bounds are at least 2^{n-1}, it follows that A_{I} and A_{II} are optimal *unrestricted* strategies against the other player's strategy. Hence they must both defect in the last round, and similarly for all rounds. $\square$

The proposition can be generalized to arbitrary games, with the bound modified to $\frac{s^{n-1}}{s-1}$, where s is the size of the strategy space. We shall next state and prove the converse of this result: As long as one of the players is restricted to subexponential automata, then approximate collaboration is possible.

Theorem 1: For every $\epsilon > 0$, in the n-round prisoner's dilemma played by automata with sizes bounded by $s_{\mathrm{I}}(n), s_{\mathrm{II}}(n)$, if at least one of the bounds is smaller than $2^{c_\epsilon n}$, where1 $c_\epsilon = \frac{\epsilon}{12(1+\epsilon)}$, then for large enough n there is a (mixed) equilibrium with average payoff for each player at least $3 - \epsilon$.

---

1 [1] Note that if $\epsilon \geq 2$, then the theorem is trivially fulfilled even if the automata are unbounded (by the continuous defection strategy for both players), and if $\epsilon < 2$, then $c_\epsilon > \epsilon/36$; thus, the theorem holds as long as one of the state bounds is smaller than $2^{\epsilon n/36}$.

As we mentioned in the introduction, our starting point was the following theorem by Abraham Neyman [Ne1].

Neyman's Theorem: For every integer k there is an integer N_0 such that if $n > N_0$ and $n^{\frac{1}{k}} \leq \min\{s_{\rm I}(n), s_{\rm II}(n)\} \leq \max\{s_{\rm I}(n), s_{\rm II}(n)\} \leq n^k$ then there is an equilibrium in the n-round prisoner's dilemma played by automata with sizes bounded by $s_{\rm I}(n)$ and $s_{\rm II}(n)$ with payoff greater than $3 - \frac{1}{k}$ for each player. $\square$

In the ten years since its announcement in *Economics Letters*, no proof of this important result has appeared, except for this elliptical paragraph in the chapter on repeated games with complete information in [AH]:

> The idea of the proof relies on the observation that the cardinality of the set of histories is an exponential function of the length of the game. It is now possible to "fill" all the memory states by requiring both players to remember "small" histories, i.e., by answering in prespecified ways after such histories (otherwise the opponent defects forever) *and then by playing collaboratively during the remaining stages.* Note that no internal state will be available to count the stages, and that collaborative play arises during most of the game.

Our proof of the stronger Theorem 1 is generally based on the idea outlined above; however, subtle and unexpected difficulties necessitate certain twists and modifications. Our equilibrium consists of mixed strategies, in which the participating automata start by exchanging short customized sequences of C's and D's (what we call "business cards" in our proof); each automaton in the support of the mixed strategy has its own individual business card. After this first stage, the two players *periodically repeat the exclusive or of their business cards throughout the n rounds*, intermittently with long periods of collaboration. (Here by "exclusive or" we mean a sequence of actions the ith of which is C if the ith action of the two business cards is the same, and D otherwise.) This ensures that they both remember their opponent's business card —and therefore have no memory left for mischief...

There are two additional difficulties, which we describe now to motivate the construction. The "business cards" exchanged by the players in the beginning are strings in $\{C, D\}^d$ for some small d. Now, exchanging them entails actually playing according to this string for several rounds. Obviously, the players that have business cards heavy in D's have a small advantage over those with business cards rich in C's. Thus, a better mixed strategy would be the pure one consisting of the straight-D automaton. But this strategy lacks the diversity that is crucial to the proof. One could imagine a variant in which all business cards have equal numbers of D's and C's; this fails even worse, since there are many states at the card-exchange phase that are not fully utilized (see the next paragraph). To overcome this problem, we use two tricks: First, the exchange of business cards is followed by a phase whose purpose is to reverse the net advantage of a D over C in business cards. This way, the effect of announcing favorable and unfavorable business cards is balanced. But another problem remains: The imbalance in the periodic repetitions. We solve this by having the two players repeat the *exclusive or* of their business cards; as a result, the average over all opponents is balanced.

Perhaps the more subtle problem (also noticed in [MW]) is the following: Most states of the automata constructed have one "expected" transition (the expected opponent behavior at this point), and one "punitive" transition to a perpetually defecting state. If a cheater plays against such a strategy, (s)he does not need the threat of the punitive transitions. But this means that the vast majority of the states (all but the initial states during which the business card exchange takes place) *are only half-used.* By this we mean that one of the two transitions is available for later reuse, thus saving potentially roughly half the states. Suppose for example that the automaton of player I has a state q in which the (honest) player I plays C and expects player II to play C, and there is another state p in which player I plays C and expects D. Then a cheater can use only one state, say q, instead of two to play the role of both q and p: set $\delta(q, D)$ equal to $\delta(p, D)$, redirect all transitions coming into p to come now into q, and remove state p. Clearly, as long as player II plays honestly (i.e. as expected), he will not observe any difference between the cheater and the original automaton. By combining such pairs of states in which player I plays the same strategy while expecting different strategies from II, a cheater can potentially save many states, which it can use instead to count and defect at the end, exposing the lack of equilibrium. In remedy, our construction guarantees that *the expected transitions of Player II always coincide with the strategy chosen by Player I, and vice-versa.* That is, after the business card exchange initial period, the two players recite the same sequence of strategies in unison, thus ruling out reuse of states.

After this motivating discussion, we are ready to proceed with the proof:

Proof of Theorem 1: We shall first describe the construction in the case in which both bounds are equal and less than $2^{c_\epsilon n}$, and in fact equal to $S = 2^d(d+\ell+2)$ for some integers d and ℓ, such that the following inequalities are satisfied:

$$3\ell + 2d < n, \quad \frac{2+d}{\ell} + \frac{4d}{n} < \epsilon. \tag{1}$$

We shall then generalize the basic construction to arbitrary state bounds of which one is at most $2^{c_\epsilon n}$, as claimed in the theorem.

The equilibrium consists of one mixed strategy for I, described next, and a symmetric one for II. The mixed strategy for I consists of 2^d automata, each with probability 2^{-d}, and each with S states. The automata are indexed by strings in $\{C, D\}^d$; automaton A_x, where $x \in \{C, D\}^d$ is the *business card* of A_x, is shown in Figure 4. The automata in the strategy of II are denoted B_y, where again y is a d-letter business card. The starting state of A_x is the root of a full binary tree of depth d, whose states at the ith level (the first level being the root) play the strategy in $\{C, D\}$ that is the ith letter of x. The 2^d leaves of the tree are indexed by the strategies in the transitions that led to this leaf: The string y corresponding to leaf y is the business card of the opponent that leads to this state, namely B_y.

Figure 4: The automaton A_x.

When the two players A_x and B_y have reached a leaf of their binary tree, we say that they have *exchanged business cards.* This accounts for $2 \cdot 2^d - 1$ states. There is also a *punitive state*, with action D and transitions back to itself. The bulk of the state space consists of disjoint sets of $d+1+\ell$ states, "hanging" from each one of the 2^d leaves. These states are arranged as a path of length $d+1$ (the *second or fixup phase*) followed by a loop of length ℓ (the *third or main phase* consisting of *cooperation and checking*). Notice that the total number of states is thus S.

From all these $2^d(d+1+\ell)$ states, one of the two transitions (called the *punitive transition*) leads to the punitive state, and the other transition leads to the next state in the path or loop called the *expected transition*, (only the expected transitions are shown in Figure 4).

The purpose of the first d steps of the fixup phase is to iron out imbalances between automata resulting from announcing more or less advantageous business cards[1]. In the ith step of the fixup phase, for $i = 1, ..., d$, both players play C iff the ith letter in both their business cards is C; otherwise (if at least one of them has a D in the ith letter) they both choose D. In the $(d+1)$st step of the fixup phase both players play D; the purpose of this step is to distinguish states of the fixup phase from the main phase where we will have both players play C in the last step.

After the fixup phase the automata enter the main phase, a loop of length ℓ. Since $\ell << n$, this loop will be repeated several times. The loop consists of two segments: a cooperation segment in which the players collaborate (this forms the bulk of the loop), and a checking segment whose purpose is to check the legitimacy of the other player (presumably B_y, since we are discussing the part of A_x that is hanging from the leaf labeled y), and to allow the other player to also check A_x's legitimacy. During the main loop the two automata collaborate (play C) all the time, except for a sequence of $d+2$ steps that form the checking segment. These $d+2$ steps are placed in the loop so that the nth round does not occur during this checking segment. Thus, assuming that $n-2d \bmod \ell < \ell-d-2$, we can place the checking segment at the end of the loop; otherwise, these steps are moved to the first $d+2$ steps of the loop (note that $d << \ell$). *During the $d+2$ steps of the checking segment A_x and B_y repeat in unison the sequence of actions $D(x \oplus y)C$,* (that is, first D, then the exclusive or of x and y, and then D), where we define $D \oplus D = C$, $C \oplus C = C$, $C \oplus D = D$, and $D \oplus C = D$. Informally, this has the following effects: (a) A_x makes sure that the opponent remembers x, A_x's business card; (b) Similarly, A_x convinces B_y that it indeed remembers y; (c) The imbalance coming from the number of D's in x vanishes when averaged over all B_y's; and (d) *After the card exchange phase, both automata recite exactly the same string;* that is, in each state in the parts of the automaton hanging from each leaf of the binary tree, the expected transition is labeled by the action of the state. This is crucial for proving state minimality.

The expected payoff of the two strategies is now easy to calculate: It is $3n - \frac{5}{2}d - 2 - (d+2) \cdot \lfloor \frac{n-2d}{\ell} \rfloor$. The $3n$ term means that the automata usually get a payoff of *three*

[1] This balancing is quite simple for the payoffs of the prisoner's dilemma as defined in Figure 1, because of lucky coincidences in the numbers. Variants of the payoffs —or totally different games, see the next section— require more elaborate constructions; this affects the constant c_ϵ of the exponent in the bound.

for most steps (with certain exceptions caused by the protocol and accounted for by the other terms). The term $\frac{5}{2}d$ means that in the first $2d$ steps, the automaton will get an average payoff of $\frac{7}{4}$ (averaged over all opponents B_y, and over all steps of the business card announcement and the fixup phases). This is because, for each $i \leq d$, if the ith letter of x is a C, then the average payoff in the business card exchange phase is $\frac{3+0}{2} = \frac{3}{2}$, whereas in the fixup phase it is $\frac{3+1}{2} = 2$; and if the ith letter is a D, then the average payoff in the business card exchange phase is $\frac{4+1}{2} = \frac{5}{2}$, while the payoff at the fixup phase is always 1. In both cases, the average is $\frac{7}{4}$. The -2 term is due to the $d+1$st step of the fixup phase. Finally, for each of the executions of the loop we lose a payoff of two at the D step that marks the beginning of the checking segment, and one on the average for each step in the exclusive or part. (If the case $n - 2d \bmod \ell \geq \ell - d - 1$ prevails, then $\lfloor \frac{n-2d}{\ell} \rfloor$ becomes $\lceil \frac{n-2d}{\ell} \rceil$.) Notice that the payoff is independent of x —this is crucial for establishing that the pair of mixed strategies being described is an equilibrium. Also, since $\frac{2+d}{\ell} + \frac{4d}{n} < \epsilon$, this payoff is within ϵn of the ideal collaborative payoff of $3n$.

It remains to prove that this pair of mixed strategies is an equilibrium. Suppose that an *impostor automaton*, call it A', plays against the present mixed strategy of II. We shall establish that A' cannot get higher payoff as long as it does not have more than S states; in fact even $n - 2d - 2\ell - 3$ extra states cannot help it.

Suppose first that A' does not cause any of the opponent automata B_y to make a punitive transition during the first $2d + 1 + 2\ell$ steps. Let x_y be the business card that A' gives to B_y (that is, the first d plays between A' and B_y).

Lemma 1. If $y \neq y'$, then $y \oplus x_y \neq y' \oplus x_{y'}$.

Proof: Let $i \geq 1$ be the first letter at which y and y' differ. Hence, x_y and $x_{y'}$ agree on the first $i-1$ letters. But since A' decides the ith letter of x_y based only on the $i-1$ first letters of y, we must have that the ith letters of x_y and $x_{y'}$ are the same. Hence, $y \oplus x_y$ and $y' \oplus x_{y'}$ differ on their ith letters. □

Let S_y be the set of states visited by A' in the rounds $d+1$ through $2d + 2\ell$ when playing against B_y.

Lemma 2. There are at least $d + 1 + \ell$ different states in each S_y. Furthermore, if $y \neq y'$, then S_y and $S_{y'}$ are disjoint.

Proof: Consider any state $s \in S_y$, and let i be a step at which s was visited. Define the *normal path* from s at step i to be the path of length $2d + 2\ell - i$ that starts from s and always follows the transition that corresponds to the action taken at the state. We will usually use the term "normal path" to refer to the string of actions along the path. It follows from the definitions that, if the same state is visited more than once, then its various normal paths are prefixes of one another.

Since we assume that A' is not punished by B_y during the first $2d + 1 + 2\ell$ steps, the normal path starting from the state visited at step $d+1$ by A' must be exactly the same as the path of expected transitions starting from the x_y leaf of B_y. This latter path consists of the $d+1$ transitions of the fixup phase, followed by $[C^{\ell-d-2} Dy \oplus x_y C]^2$. (We assume for concreteness that the $Dy \oplus x_y C$ section occurs at the end of the main loop; the same argument applies if it occurs at the beginning.) Thus S_y must contain enough states to realize this normal path. It is easy to check that the first $d+1+\ell$ states of this path must be distinct because their normal paths are not a prefix of one another, by simply considering for each of these normal paths the first occurrence of a block of at least $\ell-d-2$ C's followed by a D. The states of the fixup and the main phase are distinguished from each other by the length and the position of this block of $C's$. Note, this is true even in the case in which $y \oplus x_y = C^d$, because of the starting D before $y \oplus x_y$. Finally, the normal paths of the $d+1+\ell$ first states in S_y and $S_{y'}$ are not prefixes of one another (because of Lemma 1), and so they must be all different. □

We have thus "accounted for" all states of A_x except for the top 2^d-1 ones, and the punitive state. This latter cannot be accounted for, because an impostor may choose not to punish its opponent, and still fare well against B_y. But the binary tree states must all be distinct, and also distinct from the S_y states:

Lemma 3. The states to which A' can get during the first d steps are 2^d-1 states that are disjoint from the S_y's.

Proof: Consider the 2^d-1 strings of length at most $d-1$, and the states to which they drive A'. We claim that these states are distinct and disjoint from the S_y's. Consider a string z of length $i<d$ and let $s(z)$ be the state reached by A' in response to z. Let w be the normal path (string) starting at $s(z)$ of length $d-i$, let $y=zw$ and let $s(y)$ be the final state of this path. By definition, $s(y)$ is the state visited by A' at step $d+1$ when playing with B_y, i.e., the first state of S_y. It follows then from Lemma 2 that the states $s(z)$ for distinct strings z of length less than d are different from each other. It is easy to see also that these states are different than all the states of all the sets S_y, by similar arguments. □

Note that A' need not be identical to any one of the A_x because it may still give different business cards x_y to different B_y. However, for each $i=1,...,d$, the ith symbol that A' gives is independent of the ith symbol of y. By the construction of the fixup phase, this implies that the payoff gained by A' in the first $2d+1$ steps is the same as the payoff of any A_x, i.e., an average payoff of $\frac{7}{4}$ for each one of the first $2d$ steps and 1 for the $(2d+1)$th. By Lemma 1, the strings $y \oplus x_y$, for all y, span all possible strings of length d, hence A' gets the same payoff as any A_x in the next 2ℓ steps. The only way that A' can gain any extra payoff afterwards, is if it deviates from the normal play for some B_y. Suppose this is the case. After the deviation, A' will be punished for the remainder of the game, therefore a deviation is profitable only if it occurs in the last step, i.e., A' plays D instead of C. Thus, the automaton A' playing with B_y follows from step $2d+1$ up to n a normal path of the form $[C^{\ell-d-2}D(y \oplus x_y)C]^k C^j D$, with $j<\ell-d-2$. It is easy to check that all the suffixes of length more than ℓ are not prefix-related, hence all but possibly the last ℓ states are distinct. By similar arguments, they are also distinct from states of other $S_{y'}$ and the states visited by A' in the first d steps. We already counted (in S_y) the first ℓ states of this path. Therefore, A' needs an additional set of $n-2d-2\ell-2$ states in order to be able to deviate in the last step and gain an advantage. Hence, if A' is not punished by any B_y in the first $2d+1+2\ell$ rounds, then it cannot improve on the payoff of A_x.

Suppose that A' is punished by certain B_y's in the first $2d+1+2\ell$ rounds, and let P be the set of y's for which this happens. For each $y \in P$, the payoff of A' with B_y is at least

$(n-2d-1-2\ell)$ below the payoff of any A_x, (since A_x and B_y play CC in more than half of the steps in every loop), for a total loss of $|P|(n-2d-1-2\ell)$ in payoff. Lemma 2 still applies to the S_y's corresponding to $y \notin P$. As for the proof of Lemma 3, it applies to all strings of length less than d that are not prefixes of any $y \in P$. That is, for each $y \in P$ we may fail to completely account for at most $(2d+1+\ell)$ states, for a total of $|P|(2d+1+\ell)$ states. The only way for A' to profit from this deviation is by gaining a payoff of 1 at the last round for certain y's. To do so, A' must unravel the ℓ cycle corresponding to each such y, and "count up to n." As we explained above, this will take at least $n-2d-1-2\ell$ additional states for each such y. The total gain in payoff is $\frac{|P|(2d+1+\ell)}{n-2d-2\ell}$, less than the loss. We must therefore assume that $P = \emptyset$.

This completes the proof for the case of the restricted bound $S = 2^d(d+\ell+2)$. The pair of strategies that we constructed cannot be improved for either player (even using automata with up to $S+(n-2d-2\ell-3)$ states). We shall next modify slightly the basic construction so as to relax the assumptions about the bounds. First, let us take care of an easy case: When one of the bounds is less than n.

Lemma 4. If $S_{\mathrm{I}}(n) < n-1$ for some n, then there is an equilibrium with payoff at least $3n-3$.

Proof: If $S_{\mathrm{II}}(n) < n$, then two automata playing tit-for-tat are at equilibrium. Otherwise, the equilibrium consists of tit-for-tat, against an automaton with n states that collaborates at all rounds except the last. □

Therefore, we can assume that the bounds are both at least $n-1$. We shall first relax the assumption that S, the common state bound, is of the form $S = 2^d(d+\ell+2)$. Let $S \leq 2^{c_\epsilon n}$ be the given bound, let d be the largest integer such that $2^d d(2+\frac{2}{\epsilon}) \leq S$, and let ℓ be the largest integer such that $2^d(d+\ell+2) \leq S$. From our choice of d we have $d + \log d + \log(2+\frac{2}{\epsilon}) \leq c_\epsilon n$, hence $d+1 \leq c_\epsilon n = \epsilon n/12(1+\epsilon)$. On the other hand, $2^{d+1}(d+1)(2+\frac{2}{\epsilon}) \geq S \geq n-1$. From the choice of ℓ and d, we have $d+\ell+2 \leq S/2^d < 2(d+1)(2+\frac{2}{\epsilon})$; therefore $3\ell+2d < (d+1)12(1+\epsilon)/\epsilon = (d+1)/c_\epsilon < n$, fulfilling the first inequality of (1). Also, $d+\ell+3 > S/2^d \geq d(2+\frac{2}{\epsilon})$; if n is large enough so that $d \geq 6$ (e.g $n \geq 1+2^6 6(2+\frac{2}{\epsilon})$)), then this inequality implies that $2\ell\epsilon > 4d+2\epsilon(d-3) \geq 3d+6$, hence, $\frac{2+d}{\ell} < \frac{2\epsilon}{3}$. Since $\frac{4d}{n} < \frac{\epsilon}{3}$, it follows that the second inequality of (1), $\frac{2+d}{\ell} + \frac{4d}{n} < \epsilon$ also holds.

The construction for this extension is a small modification of the basic one, accounting for $R < 2^d$ additional states, where R is the remainder of the division of S by 2^d. We achieve this as follows: Let T be any set of R strings of length d. For each pair of automata A_x and B_y such that $x \oplus y \in T$, consider the fixup phases in these automata that correspond to the interaction between the two. Increase the lengths of these phases from $d+1$ to $d+2$ by inserting a state that plays C right before the last state of the phase (which plays D). Obviously, this modification does not affect the payoff, and has the effect of increasing the number of states in all automata to S.

Finally, we extend the construction to the general case, where the bounds may be unequal, and in fact only one player may be bounded. By Lemma 4 we can assume that both bounds are at least $n-1$. If they are both at most $n+1$, we can use the original construction with $S = n-1$; recall that our pair of strategies cannot be improved by any impostor automaton even if it has $n-2d-2\ell-3$ extra states. So assume one of the bounds, say $S_{\mathrm{II}}(n) \geq n+2$, and suppose that Player I is bounded by $S_{\mathrm{I}}(n) \leq 2^{c_\epsilon n}$.

In our equilibrium pair, I uses a strategy of only two (equiprobable) automata corresponding to the business cards C^d and $C^{d-1}D$, and II uses a mixed strategy consisting of 2^d automata, one for each $y \in \{C,D\}^d$, with all of them equiprobable. The expected transitions are as before: announcement of business card y (d steps), fixup phase, and repeated executions of the ℓ-long main loop. However, the fixup and main phases are now different, because the averaging arguments based on the diversity of the x's are no longer valid. The general structure of each automaton of I is similar to the basic construction: a full binary tree on top followed by chains and loops hanging from the leaves. Every automaton of II has $n+2$ states that form a chain of height n with one state at each level except for levels $d+1$ and $d+2$ (corresponding to the first and second step of the fixup phase) that have two states. The last state (at the nth level) is the punitive state.

The announcement phase, consisting of the first d steps, is simple: I plays his card C^d or $C^{d-1}D$ and II plays $y \in \{C,D\}^d$. Player I records y, i.e., the first d levels of each automaton of I form a full binary tree, while each automaton of II starts as a path of length d (with all transitions on symbol C). In the rest of the game the players play deterministically in unison, CC or DD.

We shall describe the main (the third) phase first before describing the fixup phase. Each automaton of I has again a loop for each $y \in \{C,D\}^d$ of length ℓ. Let $\hat{d}$ be an even number which is at least $d+\lceil \log d \rceil$, and let f be any one-to-one function from $\{C,D\}^d$ to the set of strings of length $\hat{d}$ that contain an equal number of C's and D's; note that $\hat{d}$ is large enough so that there are at least 2^d strings of length $\hat{d}$ with an equal number of C's and D's, and thus there exists such a function f. The main phase does not contain now the $\oplus$ of the two business cards, but consists of the loop $C^{\ell-\hat{d}-2}Df(y)C$. The loop is entered again at an appropriate state so that the game finishes in the block of C's. This construction again guarantees balanced payoffs for this phase. In each automaton of player II the loop is unrolled until the last stage n at which point II defects.

The fixup phase has $d+1$ steps except for a set T of R business cards y of II, for which we add one more step to the fixup phase as before so that the space bound for Player I is completely used (we give the precise value of R below). The players behave as follows. In the following we refer to the last step of the first phase (the step where I plays C or D) as the *random step*. In the first step of the fixup phase both players play C if they both played C in the random step, otherwise they both play D; this step serves to balance the payoff of the random step for both players. The rest of the fixup phase has the purpose of balancing the payoff to Player II of the first $d-1$ steps from the different business cards. Suppose that y, the business card of II, contains in the first $d-1$ letters an even number of D's, say $2a$. Then in the remainder of the fixup phase both I and II recite the string $D^aC^{d-a-1}D$ if $y \notin T$ and $D^aC^{d-a}D$ if $y \in T$ (i.e., if $y \in T$, we add an additional C step before the final D step). Suppose that y contains an odd number of D's, say $2a+1$, in the first $d-1$ letters. Then, if the two players agreed in the random step (both played C or both D), then both players recite the string $CD^aC^{d-a-2}D$ if $y \notin T$ and $CD^aC^{d-a-1}D$ if $y \in T$; otherwise (i.e., if they disagreed in the random step) they both recite the string $D^{a+1}C^{d-a-2}D$ if $y \notin T$ and $D^{a+1}C^{d-a-1}D$ if $y \in T$.

Figure 5: The fixup portion of the automaton of Player II.

The fixup portion of an automaton of Player I hanging from a leaf corresponding to a business card y is simply a path. The fixup portion of an automaton of Player II has two nodes at the first two levels and one node in the remaining ones. We depict in Figure 5 an example of the fixup phase for Player II. In the figure we assume there is an odd number of D's in the first $d-1$ letters of y; parts (a) and (b) show the cases that Player II plays C and D respectively in the random step. We have included for clarity also the random step (the last step of the first phase) corresponding to the top nodes in the figure. Only the expected transitions are shown, and the rest go to the punitive state. Note that for any business card y of II, the play of I in the random step affects only the first two steps of the fixup phase if the number of D's in the first $d-1$ letters of y is odd (in the remaining steps the players recite the string $D^a C^{d-a-2} D$ or $D^a C^{d-a-1} D$ depending on whether $y \in T$), and only the first step if it is even.

It is easy to check that Player I has no incentive to choose one symbol over the other in the random step when playing against this strategy of II: Suppose that the business card of II has an even number of D's in the first $d-1$ letters. Then the choice of Player I in the random step affects only this step and the next step. If he chooses C, then his expected payoff is $1/2(3+3)+1/2(0+1)=7/2$. If he chooses D then his expected payoff is $1/2(4+1)+1/2(1+1)=7/2$. Suppose that the business card of II has an odd number of D's in the first $d-1$ letters. Then the expected payoff for I from the random step and the first step of the fixup phase is again $7/2$ regardless of whether he plays C or D. The expected payoff for the remaining steps of the fixup phase does not depend on his choice in the random step because for either choice there is the same probability $1/2$ that it will agree or disagree with the choice of Player II in the random step.

Consider the expected payoff for Player II from the first two phases for a business card $y \notin T$. The random step and the first step of the fixup phase contribute as above $7/2$ (regardless of the choice of II in the random step). If y has an even number, $2a$, of D's in its first $d-1$ letters, then the remaining steps of the first two phases contribute $(2a)4+(d-1-2a)3$ from the first phase and $(a+1)1+(d-a-1)3$ from the second phase which sum to $6d-5$. If y has an odd number, $2a+1$, of D's in its first $d-1$ letters, then the remaining steps of the first two phases contribute $(2a+1)4+(d-1-2a-1)3$ from the first phase and $1/2[(a+1)1+(d-a-1)3+(a+2)1+(d-a-2)3]$ from the second phase, which sum again to $6d-5$. Thus, if $y \notin T$, the expected payoff to Player II from the first two phases is $6d-3/2$, independent of y. For a card $y \in T$, the expected payoff during the first two phases is $6d+3/2$ (independent again of y), i.e., has an additional payoff of 3 units because of the extra (C,C) step in the fixup phase, which is however compensated by the fact that there will be one less (C,C) step during the final main phase. Thus, the total expected payoff for II is independent of the card y.

Note that each automaton of Player I has $2^d(d+\ell+2)+R$ states. The parameters d, ℓ and R are chosen again as follows. Let d be the largest integer such that $2^d d(2+\frac{2}{\epsilon}) \leq S_{\mathrm{I}}(n)$. Let ℓ be the largest integer such that $2^d(d+\ell+2) \leq S_{\mathrm{I}}(n)$. As we showed before, for large enough n, constraint (1) holds. We let $R = S_{\mathrm{I}}(n) - 2^d(d+\ell+2)$ (i.e., R is the remainder of the division of $S_{\mathrm{I}}(n)$ by 2^d), and let T be any set of R strings of length d. This concludes the description of the two strategies.

It is easy to see that every automaton of II plays optimally (among unrestricted strategies) against the strategy of I: The string of length d played by II in the first d steps fixes the rest of the game because if II deviates before the end, he will be punished; furthermore, II defects in the last step as he should. By the construction, all initial strings of length d give the same payoff for II. The proof that I cannot improve upon his strategy for the given state bound is the same as before. This completes the proof of Theorem 1. $\square$

3. GENERAL GAMES

Let G be a (finite, two-person) game, X, Y the sets of pure strategies, and $\bar{X}$, $\bar{Y}$ the sets of mixed strategies of the two players. We may assume that X and Y have cardinality at least 2, since otherwise the equilibria are trivial. The pure (resp. mixed) *minmax* of player I is $v_1 = \min_{y \in Y} \max_{x \in X} g_{\mathrm{I}}(x, y)$ (resp. $\bar{v}_1 = \min_{y \in \bar{Y}} \max_{x \in \bar{X}} g_{\mathrm{I}}(x, y)$. Player I can always guarantee this much payoff, assuming that II uses a pure (resp. mixed strategy) known to I. The minmax v_2 (resp. $\bar{v}_2$) for II is defined similarly. The pair $v = (v_1, v_2)$ (resp. $\bar{v} = (\bar{v}_1, \bar{v}_2)$) is called the pure (resp. mixed) *threat point* of the game —for the prisoner's dilemma the two points are equal to $(1, 1)$. In general, $v \geq \bar{v}$ (comparison is component-wise), and the two points may differ. The pure or mixed *individually rational region* of a game is the part of the feasible region (the convex hull of the payoff combinations) that dominates the corresponding threat point. It can be further categorized to *strict* individually rational region (the points that strictly dominate the threat point) and *nonstrict.*

In the *infinitely repeated game* there are infinitely many rounds, the strategies are functions from sequences of opponent strategies to strategies, and the payoff is the limsup of the time-averaged payoff. The following is a well-known result [AH]:

The Folk Theorem: In the infinitely repeated game all points in the mixed individually rational region are equilibria. $\square$

Suppose that the infinitely repeated game is played by finite-state automata. That is, each finite state automaton is a strategy. The question arises, which payoff combinations are achieved by a pure equilibrium? A mixed equilibrium with finite support? We have the following variant of the Folk Theorem. (The statement below assumes that the given game payoffs are integers; for real-valued payoffs, the requirement in condition (3) for (a, b) to be a rational point should be replaced by '(a, b) is a rational convex combination of payoffs of the one-shot game'.)

Theorem 2 (The Folk Theorem for Automata). Let (a, b) be a payoff combination in the infinitely repeated game with automata. Then the following are equivalent:

(1) (a, b) is a pure equilibrium payoff.

(2) (a, b) is a mixed equilibrium payoff with finite support and rational coefficients.

(3) (a, b) is a rational point (i.e., with coordinates ratios of integers) in the pure nonstrict individually rational region.

Proof: It is trivial that (1) implies (2). To show that (2) implies (3), consider any mixed equilibrium with rational coordinates consisting of a finite set of automata for each player. First, since any play by automata is ultimately periodic, it is immediate that the payoff is a rational point (recall our assumption that the payoffs are integers). Suppose that the payoff is strictly below the pure threat point for one of the players, say player I. We can construct an automaton for player I that does the following: Initially it plays an appropriate sequence until it determines (up to equivalence in the future behavior) the particular automaton among those in the support of II it is playing against. If the support of II has m automata with at most n states each, then nm steps suffice for I to identify the automaton of its opponent (up to equivalence in the future behavior). From then on, I maximizes the payoff, given II's move. This strategy of I can be rendered as a finite automaton (2^{nm} states suffice) that achieves a payoff for I that is at least as good as the pure threat point. Hence the equilibrium cannot yield a payoff below the threat point for either opponent.

Finally, to show that (3) implies (1), take any rational point in the pure nonstrict individually rational region. It can be expressed as the rational convex combination of three payoffs. We can define a pure equilibrium as follows. Let M be the least common multiple of all the denominators in the convexity coefficients. The pure equilibrium consists of $M+1$-state automata that have expected transitions forming a cycle of length M that goes the right number of times through each strategy combination, and with a punitive state in which each player plays the strategy responsible for their opponent's minmax. □

In what follows, we restrict our attention to the pure threat point and the pure individually rational region. Any point in the individually rational region can be approximated by a rational convex combination of at most three strategy pairs. Call two pure strategy pairs A, B and A', B' *dependent* if $A = A'$ or $B = B'$, and *independent* otherwise. We say the pairs are *aligned* if $g_{\mathrm{I}}(A, B) = g_{\mathrm{I}}(A', B')$ or $g_{\mathrm{II}}(A, B) = g_{\mathrm{II}}(A', B')$, and *nonaligned* otherwise. Note that every point on the Pareto boundary corresponds to either a pure strategy or the convex combination of two nonaligned pure strategies.

We are now ready to state our main result for this section (this holds for games with real payoffs as well):

Theorem 3. Let G be an arbitrary game and let $p = (p_1, p_2)$ be a point in the (strict, pure) individually rational region. For every $\epsilon > 0$, there is a $a_\epsilon > 0$, $c_\epsilon > 0$, $n_\epsilon > 0$ such that for $n \geq n_\epsilon > 0$, in the n-round repeated game G played by automata with sizes bounded from above by $s_{\mathrm{I}}(n), s_{\mathrm{II}}(n) \geq a_\epsilon$, there is a (mixed) equilibrium with average payoff for each player within ϵ of p_i if either

(a) p can be realized by pure strategies and at least one of the bounds $s_{\mathrm{I}}(n), s_{\mathrm{II}}(n)$ is smaller than $2^{c_\epsilon n}$, or

(b) p can be realized as the convex combination of two nonaligned (or two independent) pure strategy pairs, and *both* bounds are smaller than $2^{c_\epsilon n}$.

The factor c_ϵ in the exponent is linear in ϵ, i.e., it is of the form $c\epsilon$, where c is a constant that depends on the game G and the point p (but not ϵ or n).

Notice that in (b) we require that both bounds be subexponential. In the case of the prisoner's dilemma, the region covered by part (b) consists of parts of the two diagonals, and the Pareto boundary (Figure 3). We do not know whether our techniques can be extended to the full individually rational region.

The structure of the proof is as follows. We will first take care of the case when one of the upper bounds is subquadratic. In this case the result holds for all points in the individually rational region. We break the subquadratic case into two subcases: when a bound is sublinear (Lemma 5 below) and when both bounds are at least linear (Lemma 6 below). In the sublinear case the construction is quite simple, and there is even a pure equilibrium. In the other subcase (i.e., when the smaller of the two bounds is between linear and quadratic) the construction is somewhat more complicated and uses the techniques of the previous section. The final case when both bounds are at least quadratic is the more complex one and requires some new ideas. The construction in this case is similar to the last construction of Section 2, but it requires now more work to take care of the fixup so that all business cards are equally profitable. We proceed now with the proof. In the following we use the notation of Theorem 3, i.e., G is a game. $p = (p_1, p_2)$ is a point in the individually rational region, and $s_{\mathrm{I}}(n), s_{\mathrm{II}}(n)$ the upper bounds for the automata of the two players. We start with the case that one of the upper bounds is sublinear.

Lemma 5. Let G be an arbitrary game and $p = (p_1, p_2)$ a point in the (pure, strict) individually rational region. For every $\epsilon > 0$, there are $a_\epsilon > 0$, $b_\epsilon > 0$, $n_\epsilon > 0$ such that for all $n \geq n_\epsilon > 0$, if $a_\epsilon \leq \min(s_{\mathrm{I}}(n), s_{\mathrm{II}}(n)) \leq b_\epsilon n$ then the n-round repeated game G has a (mixed) equilibrium with average payoff for each player within ϵ of p_i.

Proof: The point p can be written as a convex combination of the payoffs of (at most three) pure strategy pairs (A_i, B_i). We can assume without loss of generality that the coefficients in the convex combination are rationals, because otherwise, we can just take such a point p' that is within $\epsilon/2$ of p and prove the lemma for p' and $\epsilon/2$. So, let $p = \sum \frac{c_i}{M} q_i$, where q_i is the payoff of the pair (A_i, B_i) and the c_i's are integers that sum to M. Assume without loss of generality that $s_{\mathrm{I}}(n) \leq s_{\mathrm{II}}(n)$. Let $(A*, B*)$ be a pair of pure strategies of G with the highest payoff for II, say w. The equilibrium consists of a pair of pure strategies for the two players that are cycles of the same length, $K = kM + 1$ for an appropriate constant k, and an additional punitive state. Assume without loss of generality that the pairs (A_i, B_i) are indexed in increasing order of their payoffs to player II. Then the cycle for player I has a state that plays $A*$ followed by kc_1 states that play A_1, then kc_2 states that play A_2, etc. The expected transition out of states $A*$, A_1, A_2, etc. are respectively $B*$, B_1, B_2, etc.; the other transitions go to the punitive state that plays the strategy that achieves the minmax for the opponent. The automaton for player II is defined symmetrically. The initial states are defined so that the two automata play $(A*, B*)$ in round n. The constant k is chosen so that the average payoff for each player is within ϵ of p. We assume wlog that ϵ is small enough, namely $\epsilon < p_1 - v_1$ (recall that we assumed that p is in the *strict* individually rational region; here we only need $p_1 > v_1$). We let $a_\epsilon = K + 1$ and $b_\epsilon = 1/2K$.

Let F_1, F_2 be the two automata. We claim that F_2 is an optimal unrestricted strategy for player II. If II deviates at some point from the expected play, it can only get at that point at most the payoff of $(A*, B*)$, and will then get v_2 for each remaining round. The order of the states on the cycles ensures that this is less than the payoff in the expected play.

Let F_1' be an automaton for player I with at most $n/2K$ states. If F_1' deviates at some point in the first $n/2$ rounds, then it may gain a one-time constant advantage at that particular point, but then will lose payoff proportional to n in the remaining $n/2$ rounds. We can choose n_ϵ so that it is not profitable for I to deviate in the first $n/2$ rounds. Let σ be the string of the inputs around the cycle of F_1 starting from the initial state (i.e., the sequence of the first K expected plays of Player II). Since the length of σ is K and F_1' has no more than $n/2K$ states, after $n/2K$ applications of σ, i.e. by round $n/2$, the automaton F_1' has entered into a loop. That is, if F_1' has not deviated by round $n/2$, it will never deviate. □

We show next that the same result holds if one of the bounds $s_{\mathrm{I}}(n), s_{\mathrm{II}}(n)$ is $o(n^2)$.

Lemma 6. Let G be an arbitrary game and $p = (p_1, p_2)$ a point in the (pure, strict) individually rational region. For every $\epsilon > 0$, there are $a_\epsilon > 0$, $d_\epsilon > 0$, $n_\epsilon > 0$ such that for all $n \geq n_\epsilon > 0$, if $a_\epsilon \leq \min(s_{\mathrm{I}}(n), s_{\mathrm{II}}(n)) \leq d_\epsilon n^2$ then the n-round repeated game G has a (mixed) equilibrium with average payoff for each player within ϵ of p_i.

Proof: We can assume again without loss of generality that p is a rational convex combination of the payoffs of (at most three) pure strategy pairs, and let M be the least common multiple of the denominators in the convexity coefficients. Fix an $\epsilon > 0$. By Lemma 5 we may assume that both $s_{\mathrm{I}}(n), s_{\mathrm{II}}(n) > b_\epsilon n$. We will describe first the strategies of the two players, and then describe the corresponding automata and explain how to choose the parameters in the strategies. Choose two arbitrary pure strategies of G for each player and call them C, D. Assume without loss of generality that $s_{\mathrm{I}}(n) \leq s_{\mathrm{II}}(n)$. We call Player II the *master*. Player II has a set S_2 of "business cards" that are strings of length $d = 2\lfloor \log n \rfloor$ over C, D, We let S_2 consist of the lexicographically smallest $|S_2|$ strings of length d, where say $C < D$. Player II has one automaton B_y (pure strategy) for each card y, and all of them are equiprobable. Player I has a pure strategy, one automaton A_{C^d} corresponding to the "business card" $x = C^d$. The automaton of Player I is depicted in Figure 6. The "expected" play for I and card y of II is as follows. A player goes to the permanently punitive state if the opponent deviates.

Phase 1: In the first $2d$ steps the two players exchange cards and fix up the payoff; we perform the fixup now after each symbol. Formally, for $j = 1, ..., d$, in step $2(j-1)+1$ player I plays C and II plays the jth symbol y_j of his card y, and in the next step $2(j-1)+2$, I plays C and II plays $\bar{y}_j$, the complement of y_j (where $\bar{C} = D$ and $\bar{D} = C$).

Phase 2: Players I and II play CC for ℓ steps. (The purpose of this phase is to use up most of the states allowed for Player I.)

Phase 3: In the following $2d$ steps, players I and II play in unison $y_d \bar{y}_d ... y_1 \bar{y}_1$. (The purpose of this phase is for Player I to show that he remembers the card y of II.)

Phase 4: In the last phase I and II play repeatedly in a coordinated fashion a loop of $K = kM + 1$ pure strategy pairs which is defined as in Lemma 5: it contains the best pair $(A*, B*)$ for Player II along with a constant number k of repetitions of the strategy pairs that achieve p, ordered again according to their payoff for II, and such that the play at

Figure 6: The automaton A_{C^d} of Player I.

step n is $(A*, B*)$. (The purpose of this phase, which will consume most of the game, is to approximate the desired payoff for the two players.)

Each automaton B_y for II is a path of length $4d + \ell$ followed by the loop, and has also the punitive state. We describe now the automaton A_{C^d} that realizes the strategy of player I (see Figure 6). Consider the *trie* T_2 of the strings in S_2; it is a binary tree (not necessarily full) with every edge labelled by C or D, it has exactly $|S_2|$ leaves all at depth d, and the labels on the path from the root to a leaf $y \in S_2$ spell the string y. Since S_2 consists of the lexicographically smallest strings, T_2 has less than $2|S_2| + d$ nodes and edges (and more than $2|S_2| - 2$). The portion of the automaton A_{C^d} for Phase 1 consists of the trie T_2 with one state inserted in the middle of each edge. Phase 2 corresponds to a path of length ℓ hanging from every leaf of Phase 1. Phase 3 consists of an inverted copy of the trie T_2 again with subdivided edges, and which has the last nodes of the paths of Phase 2 as its leaves. Phase 4 is a cycle of length $K = kM + 1$. Finally there is the additional punitive state. The number of states of A_{C^d} is less than $K + (\ell + 8)|S_2| + 4d$.

We choose the parameters as follows. Pick k a constant large enough so that the average payoff for each player in the loop is within $\epsilon/2$ of p_i. Let $\ell = t \cdot n$, where t is the largest number that satisfies the following inequalities: (1) $t \leq b_\epsilon/2$ (so that we can define below S_2 of cardinality at least 1); (2) $t\delta \leq \min\{\epsilon/2, (p_i - v_i)/2\}$, where δ is the maximum absolute difference between the payoff of either player for one of the strategies CC, CD, DC, DD and p_i (so that Phases 1-3 play a small role in the total payoff); and (3) $t \leq \min\{1/8K, \sqrt{(p_1 - v_1)/(w_1 - p_1)}/K\}$, where w_1 is the maximum payoff that Player I receives in G from any pair (so that it will be unprofitable for either player to deviate). Let $d_\epsilon = t^2/10$. Choose S_2 as large as possible so that $K + (\ell + 8)|S_2| + 4d$ does not exceed the corresponding upper bound on the number of states. Then the difference between the upper bound and the number of states $\ell|S_2|$ in Phase 2 of A_{C^d} is no more than $K + 1 + 8|S_2| + 4d + \ell + 8$, which is less than 2ℓ (for large enough n).

Consider an impostor automaton A' for Player I with at most $s_I(n)$ states. Let P be the set of the cards y of II such that A' playing with B_y deviates in the first $4d + \ell$ steps. The extra benefit that A' derives in the first $4d + \ell$ steps is at most $w_1 + (4d + \ell)\delta$, while it loses in the remaining steps $[n - (4d + \ell)](p_1 - v_1)$, so the net loss is at least $n(p_1 - v_1)/2$. Consider the y's that are not in P; by looking at the normal paths of the states visited in the first $4d + \ell$ steps, we can argue as before that A' contains disjoint sets of ℓ distinct nodes for these y's. This leaves at most a slack of $(|P| + 2)\ell$ states from the upper bound $s_{\mathrm{I}}(n)$. Automaton A' can use these states to try to gain some extra payoff for some y's not in P by deviating at some point. If it deviates too early, say before step $3n/4 + (4d + \ell)$, then the net difference will be clearly negative (for large enough n). So suppose that A' deviates after step $3n/4 + (4d + \ell)$ for some y not in P. During these $3n/4$ steps of Phase 4, A' and B_y play repeatedly the same K-length sequence of pairs before deviating, therefore A' must visit at least $3n/4K \geq n/2K + 2\ell$ distinct states, and thus $|P| \geq 1/2tK$. The total loss due to P is at least $|P|n(p_1 - v_1)/2 \geq n(p_1 - v_1)/4tK$. The maximum possible gain for the y's that are not in P is $|S_2|(w_1 - p_1)K < tn(w_1 - p_1)K/10$ which is less than the loss according to (3).

As for Player II, it is easy to check that the payoff is independent of y, and that II will only lose by deviating at any point. Thus, each automaton B_y achieves the optimal unrestricted payoff against the strategy of I. $\square$

We assume from now on that both bounds are at least quadratic[1] As in the last section, the basic construction consists again of business card exchange and fixup, followed by the main loop of cooperation and checking. Each player has 2^d automata in his mixed strategy, where each automaton corresponds to a "business card" that is a string of length d using two of the player's strategies, which we shall still call C and D. The different automata (cards) may not be equiprobable now: in each position of his card, Player I (respectively Player II) uses C with probability q_1 (respectively, q_2 for II) and D otherwise. The probabilities q_1, q_2 may be different from each other and from $1/2$, but each player uses the same probability for each position and independent of the letters in the other positions. We perform here (as in Lemma 6) the card exchange and the fixup together in the same phase (Phase 1): each step in which the players exhange a letter of their card (we call these the "random" steps) is followed by a coordinated sequence of steps which balances the payoff of the random step to each player, then the players proceed to the next random step (exhange of the next letter), its fixup, and so on.

One of the players is designated as the *master* and the other is called the *server*; the server has a bounded number of states while the master may be unbounded. The master does not need to remember anything about the business card of the server after the first phase; the only reason that the server uses randomization in his mixed strategy is to keep the master honest. The server does need to remember some information about the master's card. Note that since q_1 and q_2 may be different than $1/2$, different business cards of the master can have (potentially drastically) unequal probabilities. If we required the server to remember the whole card of the master, a cheater could ignore the low probability cards, saving states in the process, which it could use to count and gain an advantage over the more probable cards. For this reason, we group together the master's business cards into groups, called *companies* (we will explain later how to do this), so that all the companies have approximately the same probability, say, within a factor 3 of each other. The server does not need to remember the whole business card of the master but only his company. In the second and main phase of the game, the players execute repeatedly a loop, which consists for the most part of the two players playing in the right proportion the strategy pairs that give the desired average payoff p, and in addition has a portion that encodes the master's company. In the final step the master deviates.

We now proceed to the details of the construction. The novel part is in the first phase (announce and fixup). We describe first the fixup segment for a random step, in which

[1] For the following construction it suffices actually that the bounds be at least $2n$. Lemma 6 served the purpose of ensuring this ($d_\epsilon n^2 > 2n$ for sufficiently large n); besides that, the lemma has the additional features that it covers all of the (pure, strict) individually rational region, and it requires only one of the players to have a bounded size automaton (with a quadratic bound) while the other can be arbitrary.

the players play C with probability q_1 (Player I) and q_2 (Player II), and D otherwise. The fixup segment consists of L steps using only two strategy pairs, which we will call (A, B), (A', B'); some of these strategies A, B, A', B' may be equal to C and D or to each other. The only required assumption for the following analysis is that the quantities $\Delta = g_{\mathrm{II}}(A, B) - g_{\mathrm{II}}(A', B')$ and $\Delta' = g_{\mathrm{I}}(A, B) - g_{\mathrm{I}}(A', B')$ are both nonzero. The number of AB steps they play, is determined by the outcome of the random step: If Player I played C and Player II also C, then it is x_1; if the combination was C, D then x_2; if D, C, then x_3; and if D, D, then x_4. The question is, *how can we choose the parameters q_1, q_2, and the x_i's so that each player is indifferent to the outcome of its own random step?*

Let us take Player II (the situation for Player I is symmetric). If the outcome for Player II is C, then the payoff is

$$q_1 x_1 \Delta + q_1 g_{\mathrm{II}}(C, C) + (1 - q_1) x_3 \Delta + (1 - q_1) g_{\mathrm{II}}(D, C),$$

and if it is D it is

$$q_1 x_2 \Delta + q_1 g_{\mathrm{II}}(C, D) + (1 - q_1) x_4 \Delta + (1 - q_1) g_{\mathrm{II}}(D, D),$$

where we have omitted from both expressions the common term $L g_{\mathrm{II}}(A', B')$. Player II is indifferent to the outcome of its own random step if and only if these two quantities are the same, and this allows us to solve for q_1. A straightforward calculation gives $q_1 = \alpha/(\alpha+\beta)$, where

$$\alpha = (x_4 - x_3)\Delta + g_{\mathrm{II}}(D, D) - g_{\mathrm{II}}(D, C) \text{ and } \beta = (x_1 - x_2)\Delta + g_{\mathrm{II}}(C, C) - g_{\mathrm{II}}(C, D).$$

Under which conditions is q_1 between 0 and 1? The answer is, whenever the two expressions α and β have the same sign. Since we can choose appropriately the parameters x_i, we conclude that this can always be done *as long as* $\Delta \neq 0$, by taking for example $(x_4 - x_3)$ and $(x_1 - x_2)$ to be sufficiently large and of the same sign. Similar calculations for Player I give $q = \gamma/(\gamma + \delta)$, where

$$\gamma = (x_4 - x_2)\Delta' + g_{\mathrm{I}}(D, D) - g_{\mathrm{I}}(C, D) \text{ and } \delta = (x_1 - x_3)\Delta' + g_{\mathrm{I}}(C, C) - g_{\mathrm{I}}(D, C).$$

We can ensure that Player I is indifferent if $\Delta' \neq 0$ by picking appropriate x_i so that γ and δ have the same sign.

We may not be able to achieve probability 1/2 for q_1 and q_2 because of the divisibility properties of the payoffs, but we can pick x_1 and x_4 to be large enough constants so that q_1 and q_2 are as close to 1/2 as we wish. For concreteness, let us pick $x_2 = x_3 = 4$ and $x_1 = x_4 = 20\lceil w/\delta \rceil + 4$, where w is the largest absolute value of an entry in the payoff matrices, and $\delta = min(|\Delta|, |\Delta'|)$. With these parameters, $|q_1 - (1 - q_1)| = |(\alpha - \beta)/(\alpha + \beta)|$ is bounded by $4w/36w = 1/9$, hence q_1 and $1 - q_1$ are between 4/9 and 5/9, and similarly for q_2 and $1 - q_2$. We let $L = x_1 + 2 = 20\lceil w/\delta \rceil + 6$. The order of the L steps of the fixup segment after a random step is as follows. Depending on whether the two players agreed or disagreed in the random step, they first play $20\lceil w/\delta \rceil$ steps of the pair (A, B) or (A', B') respectively. The final six steps in either case are: (A', B') once, then (A, B) four times, and finally (A', B') once again. The final 6 steps have the purpose of identifying the ends of the fixup segments: the same pattern does not occur anywhere else in the play.

We assume without loss of generality for concreteness in the following that Player II is the master. Figure 7 shows the section of the master's automaton for a fixup segment: it consists of two paths that merge for the common suffix of the final 6 steps. The first phase (announce and fixup) of the master's automaton consists of d such segments, chained together one after the other. This is followed by a path for the main phase, up to the nth step, when the master deviates (and there is in addition a punitive state). Thus, the master's automaton has less than $2n$ states.

Figure 7: The fixup segment for the master.

The portion of the server's automaton corresponding to the first phase is a rooted "tree-like" graph, whose structure reflects the way in which the master's business cards are grouped into companies. To define formally the grouping of the business cards we use a rooted tree T_d of depth d with parallel edges, where every internal node has two outgoing edges labelled C and D which may lead to the same node or to two different nodes at the next level. The paths from the root to the leaves correspond to the different business cards of the master (i.e., the 2^d strings in $\{C, D\}^d$), and the leaves correspond to the different companies of the master. Every node u has an associated probability $p(u)$ which is the sum of the probabilities of the strings that lead from the root to that node, where C has probability q_2, D has probability $1-q_2$, and the probability of a string is the product of the probabilities of its letters. Assume without loss of generality that $q_2 \geq 1 - q_2$ (otherwise switch q_2 and $1 - q_2$ in the following); thus $5/9 \geq q_2 \geq 1/2$. We construct T_d inductively level by level. To begin with, we have just the root at level 1 with associated probability 1. To construct level $i + 1$ from level i consider a node u at level i. If the probability $p(u)$ of u satisfies $(1 - q_2)p(u) \geq q_2^i/3$, then the two edges out of u go to two distinct nodes at the next level $i + 1$ (with respective probabilities $q_2 p(u)$ and $(1 - q_2)p(u)$). If $(1 - q_2)p(u) < q_2^i/3$, then we only have one node for u at level $i + 1$ and both outgoing edges of u go to that node (which has probability $p(u)$).

A straightforward induction shows that (i) the maximum probability at level i is q_2^{i-1}, and (ii) every node at level i has probability within a factor 3 of the maximum, i.e. at least $q_2^{i-1}/3$. The claims hold trivially for $i = 0$. For the induction step, assume the claims hold for level i. A level-i node with probability q_2^{i-1} will clearly generate a child with probability q_2^i. The nodes at level $i + 1$ have probabilities of the form $q_2 p(u)$, or $(1 - q_2)p(u)$ or $p(u)$ for some node u at level i. Note that $(1 - q_2)p(u) < q_2^i/3$ implies that $p(u) < q_2^i$ since $1 - q_2 \geq 1/3$. Consequently, claim (i) holds for level $i + 1$ because $q_2 \geq 1 - q_2$. Claim (ii) for level $i + 1$ follows also since it is explicitly enforced when we create the children of level-i nodes.

Let r_i be the number of nodes of T_d at level i; thus, r_{d+1} is the number of leaves, i.e., the number of companies. Note that if a node u at some level $i < d$ generates only one child at the next level, then its child will generate definitely two children at the following level, because $1 - q_2 > q_2^2$, hence $p(u) \geq q_2^{i-1}/3$ implies that $(1 - q_2)p(u) \geq q_2^{i+1}/3$. Thus, the number of nodes doubles every two levels, i.e., $r_i \geq 2r_{i-2}$, and since $r_1 = 1$, $r_2 = 2$, we have $r_i \geq 2^{\lfloor i/2 \rfloor}$ for all i.

For each card $x \in \{C, D\}^d$ of the server we have an automaton A_x whose portion corresponding to the first phase is obtained from the tree T_d by replacing the edges by paths of length $L+1$ as follows. Consider a node u of T_d at level $i \leq d$. The corresponding node, say u', in A_x plays the ith letter of x. If the two edges out of u are not parallel, then we have two disjoint paths from u' to the nodes corresponding to the two children of u; the two paths have L intermediate nodes which play according to the steps of the fixup segment we defined above and expect the master to do likewise. If the two edges out of u are parallel, then the two corresponding paths out of u' merge for the last 6 steps that are common, leading to the node corresponding to the child of u; that is, the structure is similar to that shown in Figure 7 (switching A with B and A' with B'). The nodes of A_x corresponding to the leaves of T_d are the entry points of the main loops of Phase 2. Thus, the number m_d of states of Phase 1 is less than $2d(L+1)r_{d+1}$, where r_{d+1} is the number of companies.

The second and main phase is similar to our previous constructions. It involves the repeated execution of a loop of length ℓ, which consists of a cooperation section and a checking section. The purpose of the checking section is to have the server show that he remembers the master's company. We fix an encoding of the different companies into strings of the same length, for instance we can pick a representative business card from each company. We let the checking section contain $2d$ steps where in steps $2i-1$ and $2i$ the players play the pairs $(A, B), (A', B')$ if the ith letter of the company's code (eg. the representative business card) is C, and they play the pairs in the reverse order $(A', B'), (A, B)$ if the ith letter is D. (Note, this assumes that $A \neq A'$, though B and B' may be equal.) In the cooperation section, which is the bulk of the loop (ℓ is chosen much larger than d), the players play repeatedly in the right proportion the strategy pairs in the convex combination of the point p. The loop is entered at an appropriate point so that the game finishes with a long enough sequence of cooperation steps and the strategy pairs in the combination that realizes p are arranged so that the master does not have any incentive to deviate from the expected play before the last step. Two other details that we need also to make sure (and which are easy to take care) are that we do not form inadvertently the endpattern of the fixup stage and that the strategy pair of the last step before entering the loop differs from the last node of the loop (i.e., the other predecessor of the first node of the loop) so that the two nodes are distinguished. In addition, in order to exhaust the state bound of the server we may need to add some more steps before the main loop. The master unrolls the loop and in the last step of the game plays his most profitable strategy against the server's strategy in that step; thus, the Phase 2 portion of the master's automaton is just a path.

We will describe in each case how to choose the strategies C, D, A, A', etc. and the parameters d, ℓ, and argue that the proposed strategies form an equilibrium with payoff within ϵ of p. For simplicity, assume without loss of generality that all payoffs are at least 1 (otherwise, increase all payoffs by the same amount), and that $\epsilon < \frac{1}{4}\min(p_1 - v_1, p_2 - v_2, 1)$. Let w be the maximum payoff to either player.

For part (a) of Theorem 3, consider a pair of pure strategies (call them again C, C)

that realizes the desired payoff p, which strictly dominates the pure threat point. Take D for each player to be the strategy that achieves the other player's minmax. We use the pairs (C, C) and (D, D) in place of (A, B) and (A', B') in the construction. The payoff of D, D is dominated by the pure threat point, which, by hypothesis, is *strictly* dominated by C, C. It follows that both Δ and Δ' are nonzero (in fact, positive).

Suppose without loss of generality that $s_{\mathrm{I}}(n) \leq s_{\mathrm{II}}(n)$ and that $s_{\mathrm{I}}(n) \leq 2^{c_\epsilon n}$, where $c_\epsilon = \epsilon/8(L+4)w$. We designate Player I as the server and II as the master. Choose d to be the largest integer such that $m_d + 1 + r_{d+1}(d\frac{5w}{\epsilon} + 1) \leq s_{\mathrm{I}}(n)$, and then choose ℓ to be the largest integer such that $m_d + 1 + r_{d+1}\ell \leq s_{\mathrm{I}}(n)$. The term m_d accounts for the states in Phase 1 of the server's automaton, 1 for the punitive state and $r_{d+1}\ell$ for Phase 2. Let $R = s_{\mathrm{I}}(n) - m_d - 1 - r_{d+1}\ell < r_{d+1}$ be the remainder of the division of $s_{\mathrm{I}}(n) - m_d - 1$ by r_{d+1}. As in the last section, we choose a subset of R companies and insert for each one of them one CC step right before the last DD step of Phase 1. We enter the main loop at a suitable point so that (i) the entry point lies inside the cooperation section and more than 6 steps before its end (so that the endpattern of the fixup segment is avoided when we enter the loop, and the last node of Phase 1 which plays D is distinguished from the node of the loop that precedes the entry point), and (ii) the game ends with a sequence of more than $(2d)\frac{w}{\epsilon}$ CC steps (to ensure that the master has no incentive to deviate before the last step). The loop is long enough so that these are easy to achieve: Position first the entry point so that the game finishes at the last node of the cooperation section. If (i) is violated, then move the entry point at most $2d+6$ steps earlier in the loop to satisfy (i); condition (ii) is satisfied because the choice of ℓ implies $\ell \geq 5d\frac{w}{\epsilon}$, hence the number of CC steps at the end of the game is at least $\ell - (4d+6) > 2d\frac{w}{\epsilon}$.

The choices of d and ℓ imply the following bounds. Since $r_{d+1} \geq 2^{d/2}$, we have $2 \cdot 2^{d/2} \leq s_{\mathrm{I}}(n) \leq 2^{c_\epsilon n}$, therefore $d + 1 \leq n\epsilon/4(L+4)w$. This implies in particular that $d(L+1)$ (the length of Phase 1) is less than $n/10$. On the other hand, since $d+1$ is too large for the state bound and since $s_{\mathrm{I}}(n) \geq n$, $d = \Omega(\log n)$. For ℓ, we noted already that $\ell \geq d\frac{5w}{\epsilon}$. On the other hand, we know that $m_{d+1} + 1 + r_{d+2}((d+1)\frac{5w}{\epsilon} + 1) > s_{\mathrm{I}}(n)$. Using the facts $m_{d+1} \leq m_d + 2r_{d+1}(L+1)$ and $r_{d+2} \leq 2r_{d+1}$, we have $m_d + 1 + r_{d+1}(2(d+1)\frac{5w}{\epsilon} + 2L+4) > s_{\mathrm{I}}(n)$. From our choice of ℓ it follows that $\ell < 2(d+1)\frac{5w}{\epsilon} + 2L + 4 \leq \frac{10n}{4(L+4)} + 2L + 4$ which is less than $n/10$ for large anough n (recall, $L \geq 26$ is a constant).

Consider now a player that does not deviate when playing against the mixed strategy of the opponent that we specified above. That is, the player may choose arbitrarily his strategy at the d random steps (possibly depending on the history up to that point), but plays as expected in the other steps. The choice in a random step affects only the following fixup segment, and in the case of the master it affects also the order of two steps in the main loop (but not the steps themselves). As we already argued, both choices in the random step lead to the same payoff in the fixup segment, and clearly the same is true for the two steps in the main loop. Also, in the case of the master an additional implication of the random choices is on whether the card belongs to one of the R companies that have an additional CC step before the main loop, but as before, this again cancels with a later CC step of Phase 2 and does not affect the payoff. Thus, the payoff of a nondeviating player is independent of his random choices. The difference of each player's payoff from the desired payoff $p_i n$ is due to Phase 1 and the executions of the checking section in the main loop. Within a complete execution of the loop, the difference due to the checking section is at most $2dw < (\epsilon/2)\ell$. The difference due to Phase 1 and the at most one checking section that does not belong to a complete execution of the main loop is bounded by $d(L+1)w + 2dw = d(L+3)w < (\epsilon/4)n$. So the total difference is less than ϵn, i.e., the average payoff per step to each player is within ϵ of p_i.

We claim that the master's strategy is an optimal unrestricted response to the server's strategy. Suppose that the master deviates with k remaining steps. If $k \geq n/2$, then the honest master receives payoff at least $k(p_i - \epsilon)$: by the above accounting the discrepancy from kp_i due to complete executions of the loop is less than $(\epsilon/2)k$, and that due to Phase 1 and one checking section of an incomplete loop execution is less than $(\epsilon/4)n \leq \epsilon/2k$. The deviating master receives payoff $w + kv_i < w + k(p_i - 4\epsilon) < k(p_i - \epsilon) - (n\epsilon - w)$ since $k \geq n/2$; i.e. the deviation results in a loss, since $n\epsilon - w > 0$ for large enough n. If $k < n/2$ and hence we are in Phase 2, clearly, the master has no incentive to deviate except possibly in the checking section of the last execution of the loop. In that case his payoff in the remaining steps is at most $2dw + hv_2$, where h is the length of the final CC segment at the end of the game. Since $h \geq 2d\frac{w}{\epsilon}$, this payoff is at most $h\epsilon + hv_2 < hp_2$, so the master is better off playing the prescribed strategy.

We will show now that the server cannot improve on his strategy using $s_{\mathrm{I}}(n)$ states. Consider an impostor automaton A' for the server. Let Y_1 be the set of the master's business cards y such that A' playing with B_y does not deviate up to round $n/2$, and let Y_1' be the complementary set. Note that $d(L+1)+4\ell \leq n/2$, i.e., by round $n/2$ the players have finished Phase 1 and at least 4 executions of the main loop. The server follows the normal path after Phase 1, and the normal paths of length ℓ of the nodes in the main loops define different strings. Hence, we deduce as in the last section that if y is in Y_1, then A' contains a set of ℓ distinct nodes $S_{c(y)}$ corresponding to the company $c(y)$ of y, and these sets are disjoint for different companies.

Recall that the probability of a string over $\{C, D\}$ is the product of the probabilities of its letters where C has probability q_2 and D has $1-q_2$. The probability of a set H of strings, $Pr(H)$, is the sum of the probabilities of the strings in H. The probability of a company c is $Pr(c) = Pr(H_c)$, where H_c is the set of business cards in c. Let us say that a string σ over $\{C, D\}$ of length d or less is *weak* if $Pr(\{\tau | \sigma\tau \in Y_1'\}) > 1/4$, where by convention a string of length 0 has probability 1, that is, a string of length d is weak iff it is in Y_1'. In other words, a string σ is weak if, given that a business card has prefix σ, it belongs to Y_1' with probability greater than $1/4$. Let Y_2 be the set of cards in Y_1 that have no weak prefix, and let Y_2' be the complementary set of cards. For every weak string σ, we have $Pr(\{y | y \in Y_1', \sigma \text{ is a prefix of } y\}) > \frac{1}{4}Pr(\sigma) = 4Pr(\{y | y \in \{C, D\}^d, \sigma \text{ is a prefix of } y\})$. Summing this inequality over all weak strings σ that do not have a weak prefix, we have $Pr(Y_1') > \frac{1}{4}Pr(Y_2')$, since every card has at most one minimal weak prefix, if any. Let us say that a company c is *weak* if $Pr(H_c \cap Y_2') \geq (4/9)Pr(H_c)$, and *strong* otherwise. Let Φ be the set of strong companies and Φ' the complementary set of weak companies. The total probability of weak companies is $Pr(\Phi') = \Sigma_{c \in \Phi'} Pr(H_c) \leq (9/4)\Sigma_{c \in \Phi'} Pr(H_c \cap Y_2') \leq (9/4)Pr(Y_2') < 9Pr(Y_1')$.

We will argue now that the impostor automaton A' needs all the states of the honest server's automaton that are used by cards of the strong companies. We already noted this for the states of the main loops (in fact, for all companies that contain a card in Y_1). Consider the set of states traversed by A' during Phase 1 when playing with the master's automaton B_y for a card $y \in Y_2$. Suppose that one of these states belongs to the set S_c for a company c of some card in Y_1, say after $a(L+1)-b$ steps the automaton A' is in state $u \in S_c$, where a is the number of random steps played so far (i.e., $0 \leq b \leq L$). Let σ be the prefix of y consisting of the random choices played by the master up to step $a(L+1)-b$. If σ is a proper prefix of y (i.e., $a < d$), then consider the continuation of the game until step $(a+1)(L+1)$, where in the next random step the master plays the same letter as the server (this happens with probability at least 4/9). Note that during these steps, the server is supposed to play according to the normal path (i.e., always the same strategy as the master) and in the last 6 steps the distinguishing end pattern of a fixup segment must occur. However, the server follows the normal path starting at node $u \in S_c$, i.e. a node whose normal path agrees with a node of the main loop of some company c, and this path does not contain the distinguished end pattern. Thus, after the prefix the impostor automaton will deviate with probability at least 4/9, hence σ is a weak prefix of y, contradicting $y \in Y_2$. If $\sigma = y$, then the server has a forced play from that point on up to round $n/2$ (since $y \in Y_2$), which is the normal path starting at the corresponding node for that step in Phase 1 of the honest server automaton. Each node in the last segment of Phase 1 has a different normal path than the nodes of the loops, so, again A' will deviate in the next L steps. We conclude that all the states traversed by A' during Phase 1 for cards $y \in Y_2$ are distinct from all the states in the sets S_c for companies c that contain a card in Y_1, and in particular for strong companies c.

Suppose that $y, y' \in Y_2$ are two cards such that the two corresponding paths of A' during Phase 1 intersect at some point: A' is at node u after $a(L+1)-b$ steps against B_y and also after $a'(L+1)-b'$ steps against $B_{y'}$, where $0 \leq b, b' \leq L$. Let σ, σ' be the respective prefixes of y, y' executed up to that point. We shall show that $a = a'$, $b = b'$, and that if we follow in the tree T_d the paths corresponding to the strings σ, σ' we'll end up at the same node.

Suppose that $b \neq b'$ and consider the extension of the two partial executions from node u for L more steps, where the random step of the master encountered in each case (if any - note there is at most one in each case) is resolved according to the normal path, i.e., the master plays the same strategy as the server (probability at least 4/9). If $b \neq b'$, then the server is supposed to play differently in the two cases because of the distinguishing endpattern of the fixup segment that has to occur at different times (note that this holds even if the server is already within the end pattern in either or both cases, i.e. if $b \leq 7$ and/or $b' \leq 7$). However, since the continuations start from the same node u and both follow the normal path, they will agree and hence one of them will be rejected by the master. That is, at least one of σ, σ' is a weak prefix, contradicting $y, y' \in Y_2$. Therefore, we must have $b = b'$.

Suppose that $a \neq a'$, say $a > a'$. Consider a string τ of length $d-a$ such that $\sigma\tau \in Y_1$. This means that the path of A' starting at node u which takes the normal transition at each step except for the random steps where it moves according to τ, ends up in the set $S_{c(\sigma\tau)}$. Consider the behaviour of A' if the master's card has prefix $\sigma'\tau$; if the master plays in the next random step as in the normal path (probability at least 4/9), a discrepancy will be observed at the following distinguishing endpattern of the fixup segment. Thus, $Pr(\{\tau'|\sigma'\tau' \in Y_1'\} \geq \frac{4}{9}Pr(\{\tau|\tau\sigma \in Y_1\}$. Since σ is not a weak prefix, $Pr(\{\tau|\tau\sigma \in Y_1\} \geq \frac{3}{4}$, and therefore $Pr(\{\tau'|\sigma'\tau' \in Y_1'\} \geq \frac{1}{3}$, thus, σ' is a weak prefix, contradicting $y' \in Y_2$.

Thus, assume $a = a'$, $b = b'$. Suppose that σ and σ' lead the tree T_d to different nodes. Then, for every string τ of length $d - a$, the cards $\sigma\tau$ and $\sigma'\tau$ belong to different companies, which require different expected play from the server. Since A' is in the same state u at step $a(L+1) - b$ after σ and σ', it follows that $\{\tau|\sigma\tau \in Y_1\}$ and $\{\tau|\sigma'\tau \in Y_1\}$ are disjoint, and therefore at least one of σ, σ' is weak, contradicting again $y, y' \in Y_2$.

Consider the set of states of an honest server's automaton traversed for the master cards that belong to strong companies. We claim that A' contains a corresponding set of at least as many states that are traversed for cards $y \in Y_2$ of the strong companies. We already accounted for the Phase 2 states, and argued that the states traversed in Phase 1 are distinct from them and are partitioned into layers according to the step of the game. Consider a strong company $c \in \Phi$ and the corresponding section of Phase 1 nodes between two consecutive random steps $a, a+1$; in the honest automaton it consists of one path or two paralell paths. We already accounted for at least one path in A'. Suppose the honest automaton has two paths. We claim that c contains two cards in Y_2 that agree in the first $a - 1$ letters and disagree in the ath letter. If that is not the case, then for every card y of c that is in Y_2, the card obtained by flipping the ath letter of y (which is also a card of c) is not in Y_2. Hence $Pr(H_c \cap Y_2) \leq (5/9)Pr(H_c)$, contradicting the fact that c is a strong company. So, let $y, y' \in Y_2$ be two cards of c that agree in the first $a-1$ letters and disagree in the ath letter. The impostor automaton A' will follow the same path for y and y' up to the ath random step, at which point it will agree with one of y, y' and disagree with the other. After this, A' must traverse two distinct parallel paths as in the honest automaton, one playing C and the other D in all the steps until the common endpattern of the fixup segment.

We conclude that the impostor automaton A' contains distinct states corresponding to all the states of the honest server's automaton that are used by the cards of strong companies. Thus, the only states that A' can save are those for weak companies. Note that there are less than $2d(L+1)+\ell < (3/10)n$ states used by each company. Let $\rho = q_2^d$ be the maximum probability of a company, and recall that every company has probability at least $\rho/3$. Thus, A' saves at most $3(3/10)nPr(\Phi')/\rho < nPr(\Phi')/\rho < 9nPr(Y_1')/\rho$ states, which it can use to count and gain an advantage in some cases. It is not profitable for A' to deviate too early - in Phase 1 or an execution of the loop before the last one (because the one-shot gain of at most w will be cancelled in the next ℓ steps since $\ell(p_i - \epsilon) - \ell v_i > \ell\epsilon > w$), and even within the last loop execution, the server will not gain by deviating in the checking section, for the same reasons that we gave earlier for the master. In order for A' to deviate in the last $\ell < n/10$ steps for a card y of the master, he needs a set of at least $n/2$ additional states (besides the ones we counted), and these sets of states must be disjoint for cards that belong to different companies; the arguments are the same as in the last section based on the normal paths of the nodes that are traversed. The gain that A' obtains by deviating against a single card y is at most w, and for this it needs $n/2$ extra states that may be also used for the other cards in the same company. Hence $n/2$ extra states are needed to achieve at most expected payoff gain of $w\rho$. Thus, the total gain in the expected payoff that A'

can achieve by using the available extra states is less than $\frac{9nPr(Y_1')}{\rho}\frac{w\rho}{n/2} = 18wPr(Y_1')$. On the other hand, for each $y \in Y_1'$, the impostor A' loses at least $n\epsilon - w$: if there are k steps remaining after A' deviates, then A will get payoff $w + kv_1$, instead of at least $k(p_i - \epsilon)$ that the honest player gets, i.e. A' looses $k(p_i - v_i - \epsilon) - w > \frac{n}{2}2\epsilon - w = n\epsilon - w$, since $k \geq \frac{n}{2}$ and $p_i - v_i > 3\epsilon$. Thus, the loss in expected payoff due to the cards in Y_1' is at least $(n\epsilon - w)Pr(Y_1')$, which exceeds the maximum possible expected gain of $18wPr(Y_1')$ since $n\epsilon > 19w$ for large enough n. This concludes the proof for part (a) of Theorem 3.

For part (b) of Theorem 3, suppose that the payoff p is the convex combination of two pure strategy pairs. We describe how to pick the strategies A, B, A', B' etc. There are two basic properties we want to enforce: One is that the quantities Δ, Δ' that we defined for the pairs $(A, B), (A', B')$ are nonzero. The other property we want is that we can associate with every strategy of the server a unique strategy $f(\cdot)$ of the master so that in every nonrandom step of the game the master plays the strategy associated with the server's strategy at the same step (i.e., if the server plays E then the master plays $f(E)$), and for the random steps we have $f(C), f(D) \in \{C, D\}$ (this last constraint can be always trivially satisfied because the strategies C, D of the random steps can be arbitrary and don't need to be related to the strategies used for the fixup or in the convex combination that achieves the payoff p). Note, different strategies of the server can be mapped to the same strategy of the master. The second property allows us to define normal paths for the automata of the server, by picking for each state u with label $\lambda(u)$ the transition labelled by the master's strategy $f(\lambda(u))$ that is associated with the label of the state.

Suppose first that the strategy pairs of the convex combination are independent, and let us call them again (C, C) and (D, D). If the pairs are not aligned, then the result holds even if only one of the bounds is subexponential (and the other arbitrary). In this case we simply let $(A, B) = (C, C)$, $(A', B') = (D, D)$, and let the server be the player with the smaller bound.

Suppose that (C, C) and (D, D) are aligned, say $g_{\mathrm{II}}(C, C) = g_{\mathrm{II}}(D, D)$. We use II as the master and use the fact that $s_{\mathrm{I}}(n) \leq 2^{c_\epsilon n}$. (Recall that the hypothesis in (b) is that both bounds are subexponential. If $g_{\mathrm{I}}(C, C) = g_{\mathrm{I}}(D, D)$, then we let I be the master and use the fact that $s_{\mathrm{II}}(n) \leq 2^{c_\epsilon n}$.) Let E be the strategy of I that achieves the minmax for II. Since $p_2 = g_{\mathrm{II}}(C, C) = g_{\mathrm{II}}(D, D) > v_2$, E is distinct from both strategies C and D of I, and (E, C) has smaller payoff for II than both (C, C) and (D, D). Since the latter two pairs have unequal payoffs for I, it follows that at least one of (C, C) and (D, D), say (C, C), has different payoff than (E, C) for both players. We use these two pairs (C, C) and (E, C) in place of (A, B) and (A', B'). Note that we can associate for every server strategy a unique master strategy as desired: $f(C) = C, f(D) = D, f(E) = C$.

Finally, suppose that the payoff p is the convex combination of the payoffs of two dependent, but nonaligned strategy pairs (C, C) and (D, C). We use II as the master and assume that $s_{\mathrm{I}}(n) \leq 2^{c_\epsilon n}$. Since (C, C) and (D, C) are not aligned, we can use these two pairs in place of (A, B), (A', B') and let $f(C) = C, f(D) = C$. We pick for the master an arbitrary second strategy D for his random steps.

The parameters are picked similarly as in part (a). We can assume without loss of generality again that the coefficients in the convex combination for p are rationals c_i/M where $M = \Sigma c_i$ (otherwise, pick a point p' within $\epsilon/2$ of p that is a rational convex combination of the same strategy pairs and follow the construction for p' and $\epsilon/2$). We can assume also that all c_i are greater than $7w/\min(p_1 - v_1, p_2 - v_2)$ (multiply if necessary all the c_i and M by the same number). Assume again that the server's bound is at most $2^{c_\epsilon n}$ where $c_\epsilon = \epsilon/8(L+4)w$. Let us include an addititional step at the end of Phase 1; we choose the strategy pair (A, B) or (A', B') for this step to distinguish it from the last node of the loop. So thus far, we have m_d states for the old Phase 1, r_{d+1} for this extra step for all companies, 1 for the punitive state and $r_{d+1}\ell$ for the loops. In this case each loop consists of the $2d$-long checking section and k blocks, each of length M consisting of c_i nodes for each strategy pair in the convex combination; the pairs are in increasing order within the block according to their payoff to the master. Let d be the maximum integer such that $m_d + 1 + r_{d+1}(d\frac{5w}{\epsilon} + 3M + 1) \leq s_{\mathrm{I}}(n)$. Let k be the maximum integer such that $m_d + 1 + r_{d+1}(2d + kM + 1) \leq s_{\mathrm{I}}(n)$, and let $\ell = 2d + kM$. Let t be the integer part of the division of $(s_{\mathrm{I}}(n) - (m_d + 1 + r_{d+1}(\ell + 1))$ by r_{d+1} and R the remainder; $0 \leq t < M$ and $0 \leq R < r_{d+1}$. Add t nodes in Phase 1 for all companies right before the last extra step; the play in these steps is flexible, for example alternate the pairs (A, B) and (A', B'). Choose R companies and add one more step for them before the last extra step of Phase 1 playing the most profitable pair for the master in the convex combination. Now we have exhausted all the states allowed for the server. The choice of the parameters imply the inequalities: $d < n\epsilon/4(L+4)w < n/12(L+4)$, and $d5w/\epsilon + 2M < \ell < n/10$ (for large enough n). Position the entry point to the main loop so that (i) the game finishes in all cases close to the end of a block, say within 3 steps, (ii) the last $2dw/\epsilon$ steps of the game are in the cooperation section, (iii) the loop is entered within the cooperation section and stays within it say for the first M steps, and (iv) the end of Phase 1 and the beginning of the loop do not form the distinguishing endpattern of the fixup segment. The loop is large enough so that these properties are again easy to achieve. First, position the entry to the loop so that the game finishes at the last step of the block before the checking section (or the step before that for the R companies whose Phase 1 has an additional step); this satisfies (i) and (ii). If the entry point occurs inside or at distance M before the checking section, then move the entry point an integral number of at most $\lceil 2d/M \rceil + 1$ blocks earlier so that it is now in the cooperation section, satisfying also (iii); (i) and (ii) are still satisfied. If (iv) is violated, then move the entry point one step. Note that the loop is long enough ($\ell > 5dw/\epsilon + 2M > 2dw/\epsilon + 4d + 2M + 2$), so that even after moving the entry point the game finishes with more than $2dw/\epsilon$ cooperation steps.

The proof goes along the same lines as for part (a). Similar calculations show that the player's prescribed strategies have payoff per step within ϵ of p_i. The difference of the total payoff from $p_i n$ is due to Phase 1 and the checking sections within the loops on the one hand, which is bounded exactly as before, and on the other hand now in addition there may be a difference due to the at most M steps added at the end of Phase 1 and to a possible incomplete execution of a block of M steps from the loop; the difference of these latter $2M$ steps is less than $2Mw$, i.e. a constant, and hence is insignificant compared to $n\epsilon$ for large enough n. Thus, the difference of the payoff from $p_i n$ is again less than ϵn.

Similarly, it is easy to show again that the master does not gain anything by deviating before the last step: It is suboptimal for the master to deviate before the last execution of the main loop. Within this last execution, deviation in the checking section causes a loss again, because if there are h more cooperating steps, the honest master receives payoff at least $hp_i - 3w$ (the $-3w$ term is because the last 3 steps of the last block may be missing), whereas the deviating master receives no more than $2dw + hv_i$, which is smaller since $h > 2dw/\epsilon$ and $p_i - v_i > 4\epsilon$. In the final cooperation section, if the master deviates in the last segment containing his most profitable pair it is clearly best to do it in the very last step; if he deviates earlier with h' steps remaining (note, $h' \geq c_j - 3 \geq 7w/(p_i - v_i) - 3$), then he receives payoff $w + h'v_i$ whereas the honest master receives at least $h'p_i - 3w$ (because the pairs are ordered within a block in increasing payoff to the master). Since $h' \geq c_j - 3 \geq 7w/(p_i - v_i) - 3$, it follows that $h'p_i - 3w > w + h'v_i$, so the deviation of the master from the prescribed play results again in a loss.

The proof that the server cannot improve on his strategy is essentially the same as in part (a). We define again for an impostor automaton A' the sets of cards Y_1, Y_2 and the set Φ of strong companies exactly the same way. The proof in (a) was based on using the normal paths to distinguish between the nodes. With the above choices for the strategies, we can define normal paths for the server, and use the same arguments as in part (a) to show that A' can only save nodes used by weak companies, thus at most $(n/3)Pr(\Phi')/(\rho/3) < 9nPr(Y_1')/\rho$ states (every company uses at most $2d(L+1)+M+1+\ell \leq n/3$ states and ρ is the maximum probability of a company). For the same reasons as before (and as for the master), it is not profitable for A' to deviate before the cooperation section of the last loop execution, and it needs $n/2$ extra states to do this, which must be distinct for cards of different companies. The best that A' can do is deviate in the last block of the game and the gain is certainly at most Mw. Thus, the total gain in the expected payoff that A' can achieve by using the available extra states is less than $(9nPr(Y_1')/\rho)(Mw\rho)/(n/2) = 18MwPr(Y_1')$. On the other hand, for each $y \in Y_1'$, the impostor A' loses at least $n\epsilon - w$ as in part (a). Thus, the loss in expected payoff due to the cards in Y_1' is at least $(n\epsilon - w)Pr(Y_1')$, which exceeds the maximum possible expected gain of $18MwPr(Y_1')$ since $n\epsilon > 18Mw + w$ for large enough n. This concludes the proof of Theorem 3. □

We can extend the result along the same lines to all the *independent individually rational* points: the points of the individually rational region that can be written as rational convex combinations of the payoffs of independent strategy pairs. Recently, A. Neyman [Ne2] used a more powerful technique to generalize it to the whole of the individually rational region.

REFERENCES

[AH] R. J. Aumann, S. Hart (editors) *Handbook of Game Theory with Economic Applications, vol. I*, North-Holland, 1992.

[Ax] R. Axelrod *The Evolution of Cooperation*, Basic Books, 1984.

[BP] E. Ben-Porath "The complexity of computing a best response automaton in repeated games with mixed strategies," *Games and Economic Behavior* 2, pp. 1-12, 1990.

[DP] X. Deng, C. H. Papadimitriou "The complexity of solution concepts," *Math. of O.R.*, to appear.

[FW] L. Fortnow, D. Whang "Optimality and domination in repeated games with bounded players," *Proc. 26th ACM Symposium on the Theory of Computing,* pp. 741-749, Las Vegas, May 1994.

[GKZ] I. Gilboa, E. Kalai, E. Zemel "The complexity of eliminating dominated strategies," *Math. of O.R.* 18, pp. 553-565, 1993.

[GZ] I. Gilboa, E. Zemel "Nash and correlated equilibria: some complexity considerations," *Games and Economic Behavior* 1, pp. 80-93, 1989.

[HU] J. E. Hopcroft and J. D. Ullman *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

[Ka] E. Kalai "Bounded rationality and strategic complexity in repeated games," *Game Theory and Applications*, T. Ichiishi, A. Neyman, Y. Tauman eds., pp. 131-157, 1990.

[KM] D. Koller, N. Megiddo "The complexity of two-person zero-sum games in extensive form," *Games and Economic Behavior* 4, pp. 528-552, 1992.

[MW] N. Megiddo, A. Wigderson "On play by means of computing machines," *Proc. 1st Conf. on Theor. Aspects of Reasoning About Knowledge*, pp. 259-274, 1986.

[Na] J. F. Nash "Non-cooperative games," *Ann. Math.* 54, pp. 286-295, 1951.

[Ne1] A. Neyman "Bounded complexity justifies cooperation in the finitely repeated prisoners' dilemma," *Economics Letters* 19, pp. 227-229, 1985.

[Ne2] A. Neyman "Finitely repeated games with finite automata," to appear in *M.OR.*

[Pa1] C. H. Papadimitriou "On players with a bounded number of states," *Games and Economic Behavior* 4, pp. 122-131, 1992.

[PY] C. H. Papadimitriou, M. Yannakakis "On Complexity as Bounded Rationality," *Proc. 26th ACM Symposium on the Theory of Computing,* pp. 726-733, Las Vegas, May 1994.

[Ru] A. Rubinstein "Finite automata play the repeated prisoner's dilemma," *J. Econ. Theory* 39, pp. 83-96, 1986.

[Sa] S. Sahni "Computationally related problems," *SIAM J. Computing* 3, pp. 262-279, 1974.

[Si] H. Simon *The Sciences of the Artificial*, 1969.

[So] S. Sorin "Repeated games with complete information," Chapter 4 in [AH], 1992.