Discussion Papers in Economic Behaviour

DPEB
03/14

Bounded Computational Capacity Equilibrium

VNIVERSITAT DE VALÈNCIA

>Penelope Hernández
Universitat de València and ERI-CES, Spain

>Eilon Solan
Department of Statistics and Operations Research, School of Mathematical Sciences, Tel Aviv University, Israel

May, 2014

www.erices.es eri-ces

Bounded Computational Capacity Equilibrium*

Penélope Hernández† and Eilon Solan‡

May 6, 2014

Abstract

A celebrated result of Abreu and Rubinstein [1] states that in repeated games, when the players are restricted to playing strategies that can be implemented by finite automata and they have lexicographic preferences, the set of equilibrium payoffs is a strict subset of the set of feasible and individually rational payoffs. In this paper we explore the limitations of this result. We prove that if memory size is costly *and* players can use mixed automata, then a folk theorem obtains and the set of equilibrium payoff is once again the set of feasible and individually rational payoffs. Our result emphasizes the role of memory cost and of mixing when players have bounded computational power.

Keyword: Bounded rationality, automata, complexity, infinitely repeated games, equilibrium.

1 Introduction

The literature on repeated games usually assumes that players have an unlimited computational capacity, or unbounded rationality. Since in practice this assumption

*This work was conducted while the second author was visiting Universidad de Valencia. The first author thanks both the Spanish Ministry of Science and Technology and the European Feder Founds for financial support under project SEJ2007-66581 and Generalitat Valenciana (PROMETEO/2009/068). The second author thanks the Departamento de Análisis Económico at Universidad de Valencia for the hospitality during his visit. The authors thank Elchanan Ben Porath, Ehud Kalai, Ehud Lehrer, two anonymous referees, and the Associate Editor for their useful suggestions. The work of Solan was partially supported by ISF grant 212/09 and by the Google Inter-university center for Electronic Markets and Auctions.

†ERI-CES and Departamento de Análisis Económico, Universidad de Valencia. Campus de Los Naranjos s/n, 46022 Valencia, Spain. Penelope.Hernandez@uv.es.

‡Department of Statistics and Operations Research, School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel. eilons@post.tau.ac.il.

does not hold, it is important to study whether and how its absence affects the predictions of the theory.

One common way of modelling players with bounded rationality is by restricting them to strategies that can be implemented by finite state machines, also called finite automata. The game theoretic literature on repeated games played by finite automata can be roughly divided into two categories. One backed by an extensive literature (e.g., Kalai [8], Ben Porath [3], Piccione [16], Piccione and Rubinstein [17], Neyman [10], [11], [12], Neyman and Okada [13], [14], [15], Zemel [23]) that studies games where the memory size of the two players is determined exogenously, so that each player can deviate only to strategies with the given memory size. In the other, Rubinstein [18], Abreu and Rubinstein [1], and Banks and Sundaram [2] study games where the players have lexicographic preferences: each player tries to maximize her payoff, and subject to that she tries to minimize her memory size. Thus, it is assumed that memory is free, and a player would deviate to a significantly more complex strategy if that would increase her profit by one cent. Abreu and Rubinstein [1] proved that in this case, the set of equilibrium payoffs in two-player games is generally a strict subset of the set of feasible and individually rational payoffs. In fact, it is the set of feasible and individually rational payoffs that can be generated by a *coordinated play*; that is, a sequence of action pairs in which there is a one-to-one mapping between Player 1's actions and Player 2's actions. For example, in the Prisoner's Dilemma that appears in Figure 1, where each player has two actions, C and D, this set is the union of the two line segments $[3,3]-[1,1]$ and $[3,1]-[1,3]$.

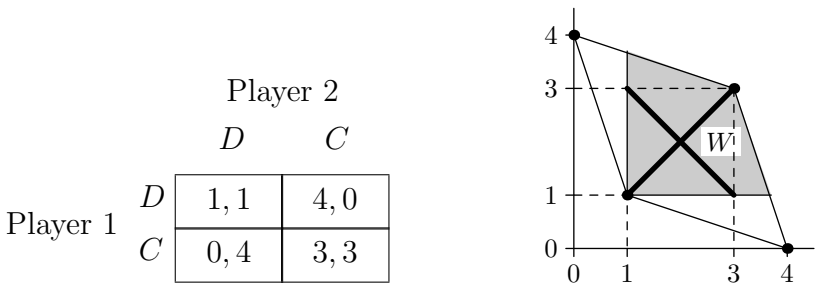


Figure 1: The Prisoner's Dilemma: the payoff matrix, the feasible and individually rational payoffs (the dark quadrilateral W), and the payoffs that correspond to coordinated play (the two thick lines).

To obtain their result, Abreu and Rubinstein [1] make two implicit assumptions: (a) memory is costless, and (b) players can use only pure automata. Removing assumption (a) while keeping assumption (b) does not change the set of equilibrium payoffs. Indeed, since the preference of the players is lexicographic, no player can

profit by deviating to a larger automaton when memory is costless, so a fortiori she has no profitable deviation when memory is costly. The construction in [1] ensures that a deviation to a smaller automaton yields the deviator a payoff which is close to her minmax value in pure strategies. Therefore as soon as memory cost is sufficiently small, there is no profitable deviation to a smaller memory as well. Removing assumption (b), on the other hand, adds new equilibrium payoffs, as exhibited by Example 6 below.

Our goal in this paper is to show that if one removes both assumptions (a) and (b) then the result of Abreu and Rubinstein [1] fails to hold. We will show that if memory is costly (yet memory cost goes to 0) and players can use mixed strategies, then a folk theorem obtains, and the set of equilibrium payoffs includes the set of feasible and individually rational payoffs (w.r.t. the minmax value in pure strategies).[1] We assume for simplicity that the players have additive utility: the utility of a player is the difference between her long-run average payoff and the cost of her computational power.

We thus present a new equilibrium concept that is relevant when memory size matters and each player's set of pure strategies is the set of finite automata. For a given positive real number c, we say that the vector $x \in \mathbb{R}^2$ is a *c-Bounded Computational Capacity equilibrium payoff* (hereafter, BCC for short) if it is an equilibrium payoff when the utility of each player is the difference between her long-run average payoff, and c times the size of its finite state machine.

A payoff vector $x \in \mathbb{R}^2$ is a *BCC equilibrium payoff* if it is the limit, as c goes to 0, of c-bounded computational capacity equilibrium payoffs, and the cost of the machines used along the sequence converges to 0.

Interestingly, the definition does not imply that the set of BCC equilibrium payoffs is a subset, nor a superset, of the set of Nash equilibrium payoffs.

Our main result is a folk theorem: in two-player games, every feasible and individually rational (w.r.t. the min-max value in pure strategies) payoff vector is a BCC equilibrium payoff.

Our proof is constructive. The equilibrium play in the BCC equilibrium that we construct is composed of three phases. The first phase, that is played only once on the equilibrium path, is a Punishment Phase; in this phase each player plays a strategy that punishes the other player, that is, an action that attains the min-max value in pure strategies of the opponent. As in [1], it is crucial to have the punishment phase on the equilibrium path; otherwise, players can use smaller machines that cannot implement punishment, thereby reducing their computation cost. However, if a machine cannot implement punishment, there is nothing that will deter the other

[1] We do not know what is the set of equilibrium payoffs in the setup of Abreu and Rubinstein [1] when one allows for mixed automata but keeps memory costless.

player from deviating. The second phase, called the Babbling Phase, is also played only once on the equilibrium path. In this phase the players play a predetermined sequence of action pairs. In the third phase, called the Regular Phase, the players repeatedly play a predetermined periodic sequence of action pairs that approximates the desired target payoff. To implement this phase, the players reuse states that were used in the babbling phase. In fact, the role of the babbling phase is to enable one to embed the regular phase within it, and its structure is designed to simplify complexity calculations. It is long enough to ensure that, with only low probability, a player can correctly guess which of the states in the other player's machine are reused.

Our construction breaks down in the setup of Abreu and Rubinstein [1] because some of the states of the automata that implement that players' strategies are reused; whereas in the construction of Abreu and Rubinstein each state had a specific role, and if the opponent deviated a long punishment phase ensued, in our construction some states accept two different actions of the opponent. When memory is costless and players can use huge automata, a player can learn which states of the opponent's automaton are reused, and play only one of these actions whenever his opponent's automaton visits a reused state, thereby increasing his overall average payoff.

The rest of the paper is organized as follows. Section 2 presents the model and the main result. The construction of a mixed equilibrium strategy for both players in the particular case of the Prisoner's Dilemma is presented in Section 3. Comments and further discussion appear in Section 4. In the appendix we indicate how the proof for the Prisoner's Dilemma should be altered to fit general two-player games.

2 The Model and the Main Result

In this section we define the model, including the concepts of automata, repeated games, and strategies implementable by an automaton, we describe our solution concept of Bounded Computational Capacity equilibrium, and we state the main result.

2.1 Repeated Games

A two-player *repeated game* is given by (1) two finite action sets $\mathcal{A}_1$ and $\mathcal{A}_2$ for the two players, and (2) two payoff functions $u_1 : \mathcal{A}_1 \times \mathcal{A}_2 \to \mathbb{R}$ and $u_2 : \mathcal{A}_1 \times \mathcal{A}_2 \to \mathbb{R}$ for the two players. We denote by $\mathcal{A} := \mathcal{A}_1 \times \mathcal{A}_2$ the set of action pairs.

The game is played as follows. At each stage $t \in \mathbb{N}$, each player $i \in \{1, 2\}$ chooses an action $a_i^t \in \mathcal{A}_i$, and receives the stage payoff $u_i(a_1^t, a_2^t)$. The goal of each player is to maximize his long-run average payoff $\lim_{t\to\infty} \frac{1}{t} \sum_{j=1}^{t} u_i(a_1^j, a_2^j)$, where $\{(a_1^j, a_2^j), j \in \mathbb{N}\}$ is the sequence of action pairs that were chosen by the players along the game.[2] A

[2] In general this limit need not exist. Our solution concept will take care of this issue.

pure strategy of player i is a function that assigns an action in $\mathcal{A}_i$ to every finite history $h \in \cup_{t=0}^{\infty}\mathcal{A}^t$. A *mixed strategy* of player i is a probability distribution over pure strategies.

2.2 Automata

A common way to model a decision maker with bounded computational capacity is as an automaton, which is a finite state machine whose output depends on its current state, and whose evolution depends on the current state and on its input (see, e.g., Neyman [10] and Rubinstein [18]). Formally, an *automaton* P is given by (1) a finite state space Q, (2) a finite set I of inputs, (3) a finite set O of outputs, (4) an output function $f : Q \to O$, (5) a transition function $g : Q \times I \to Q$, and (6) an initial state $q^* \in Q$.

Denote by q^t the automaton's state at stage t. The automaton starts in state $q^1 = q^*$, and at every stage $t \in \mathbb{N}$, as a function of the current state q^t and the current input i^t, the output of the automaton $o^t = f(q^t)$ is determined, and the automaton moves to a new state $q^{t+1} = g(q^t, i^t)$.

The *size* of an automaton P, denoted by $|P|$, is the number of states in Q. Below we will use strategies that can be implemented by automata; in this case the size of the automaton measures the complexity of the strategy.

2.3 Strategies Implemented by Automata

Fix a player $i \in \{1, 2\}$. An automaton P, whose set of inputs is the set of actions of player $3 - i$ and set of outputs is the set of actions of player i, that is, $I = \mathcal{A}_{3-i}$ and $O = \mathcal{A}_i$, can implement a pure strategy of player i. Indeed, at every stage t, the strategy plays the action $f(q^t)$, and the new state of the automaton $q^{t+1} = g(q^t, a_{3-i}^t)$ depends on its current state q^t and on the action a_{3-i}^t that the other player played at stage t. For $i = 1, 2$, we denote an automaton that implements a strategy of player i by P_i. We denote by $\mathcal{P}_i^m$ the set of all automata with m states that implement pure strategies of player i.

When the players use arbitrary strategies, the long-run average payoff needs not exist. However, when both players use strategies that can be implemented by automata, say P_1 and P_2 of sizes p_1 and p_2 respectively, the evolution of the automata follows a (deterministic) Markov chain with $p_1 \times p_2$ states, and therefore the long-run average payoff exists. We denote this average payoff by $\gamma(P_1, P_2) \in \mathbb{R}^2$.

A mixed automaton M is a probability distribution over pure automata.[3] A mixed automaton corresponds to the situation in which the automaton that is used is not

[3]To emphasize the distinction between automata and mixed automata, we call the former *pure automata*.

known, and there is a belief over which automaton is used. A mixed automaton defines a mixed strategy: at the outset of the game, a pure automaton is chosen according to the probability distribution given by the mixed automaton, and the strategy that the pure automaton defines is executed.

We will use only mixed automata whose support is pure automata of a given size m. Denote by $\mathcal{M}_i^m$ the set of all mixed automata whose support is automata in $\mathcal{P}_i^m$, and by $\mathcal{M}_i = \cup_{m\in\mathbb{N}}\mathcal{M}_i^m$ the set of all mixed automata whose support contains automata of the same size. If $M_i \in \mathcal{M}_i^m$, we say that m is the size of the automaton M_i. Thus, the size of a mixed automaton refers to the *size* of the pure automata in its support (and not, for example, to the *number* of pure automata in its support). If we interpret each pure automaton as an agent's type, and a mixed automaton as the type's distribution in the population, then the size of the mixed automaton measures the complexity of an individual agent, and not the type diversity in the population.

When both players use mixed strategies that can be implemented by mixed automata, the expected long-run average payoff exists; it is the expectation of the long-run average payoff of the pure automata that the players play:

$$\gamma(M_1, M_2) := \mathbf{E}_{M_1,M_2}[\gamma(P_1, P_2)].$$

2.4 Bounded Computational Capacity Equilibrium

In the present section we study games where the utility function of each player takes into account the complexity of the strategy that she uses.

Definition 1 *Let $c > 0$. A pair of mixed automata (M_1, M_2) is a* c-BCC equilibrium, *if it is a Nash equilibrium for the utility functions $U_i^c(M_1, M_2) = \gamma_i(M_1, M_2) - c|M_i|$, for $i \in \{1, 2\}$.*

If the game has an equilibrium in pure strategies, then the pair of pure automata (P_1, P_2), both with size 1, that repeatedly play the equilibrium actions of the two players, is a c-BCC equilibrium, for every $c > 0$.

The min-max value of player i in pure strategies in the one-shot game is

$$v_i := \min_{a_{3-i}\in\mathcal{A}_{3-i}} \max_{a_i\in\mathcal{A}_i} u_i(a_i, a_{3-i}).$$

An action a_{3-i} that attains the minimum is termed a *punishing* action of player $3-i$.

Remark 2 *Abreu and Rubinstein's [1] proof implies that for every c sufficiently small, when restricted to* pure *strategies, the only c-BCC equilibrium payoffs are the feasible and individually rational payoffs that are implementable by coordinated play. Indeed,*

suppose that x is a c-BCC equilibrium payoff in pure automata that cannot be generated by a coordinated play, and let (P_1, P_2) be a c-BCC equilibrium that supports this payoff. As in Abreu and Rubinstein's [1] proof, the optimal pure automaton against an automaton of size m is an automaton of size at most m. This implies that $|P_1| = |P_2|$. In particular, no player can profit by deviating to a larger automaton than the one that he uses. Since x cannot be generated by a coordinated play, by Abreu and Rubinstein [1] the pair of pure automata (P_1, P_2) is not an equilibrium when preferences are lexicographic, and in particular one of the players, say Player 1, has an automaton smaller than P_1 that yields higher payoff against P_2 than P_1. But then (P_1, P_2) cannot be a c-BCC equilibrium, a contradiction.

To get rid of the dependency of the constant c we define the concept of a *BCC equilibrium payoff*. A payoff vector x is a *BCC equilibrium payoff* if it is the limit, as c goes to 0, of payoffs that correspond to c-BCC equilibria.

Definition 3 *A payoff vector $x = (x_1, x_2)$ is a* BCC equilibrium payoff *if for every $c > 0$ there is a c-BCC equilibrium $(M_1(c), M_2(c))$ such that $\lim_{c\to 0} \gamma(M_1(c), M_2(c)) = x$ and $\lim_{c\to 0} c|M_i(c)| = 0$ for $i = 1, 2$.*

It follows from the discussion above that every pure equilibrium payoff is a BCC equilibrium payoff. Using Abreu and Rubinstein's [1] proof, one can show that any strictly individually rational payoff (relative to the min-max value in pure strategies) that can be generated by coordinated play is a BCC equilibrium payoff. For the formal statement, assume w.l.o.g. that $|\mathcal{A}_1| \leq |\mathcal{A}_2|$.

Theorem 4 (Abreu and Rubinstein, 1988) *Let $\sigma : \mathcal{A}_1 \to \mathcal{A}_2$ be a one-to-one function. Then any payoff vector x in the convex hull of $\{u(a_1, \sigma(a_1)), a_1 \in \mathcal{A}_i\}$ that satisfies $x_i > v_i$ for $i = 1, 2$ is a BCC equilibrium payoff.*

2.5 The Main Result

The set of feasible payoff vectors is

$$F := \text{conv}\{u(a), a \in \mathcal{A}\}.$$

The set of *strictly individually rational payoff vectors* (relative to the min-max value in pure strategies) is

$$V := \left\{x = (x_1, x_2) \in \mathbb{R}^2 \colon x_1 > v_1, x_2 > v_2\right\}.$$

Our main result is the following folk theorem, that states that every feasible and strictly individually rational payoff vector is a BCC equilibrium payoff.

Theorem 5 *If the set $F \cap V$ has a non-empty interior, then every vector in $F \cap V$ is a BCC equilibrium payoff.*

Observe that Theorem 5 is not a characterization of the set of BCC equilibrium payoffs, because it does not rule out the possibility that a feasible payoff that is not individually rational (relative to the min-max value in pure strategies) is a BCC equilibrium payoff. That is, we do not know whether threats of punishments by a mixed strategy in the one-shot game can be implemented in a BCC equilibrium.

Theorem 5 stands in sharp contrast to the main message of Abreu and Rubinstein [1] where it is proved that lexicographic preferences, which are equivalent to an infinitesimal cost function c, imply that in equilibrium players follow coordinated play, so that the set of equilibrium payoffs is sometimes smaller than the set of feasible and individually rational payoffs. Our study shows that the result of Abreu and Rubinstein [1] hinges on two assumptions: (a) memory is costless, and (b) the players use only pure automata. Once we assume that memory is costly and that players may use mixed automata, the set of equilibrium payoffs changes dramatically.

2.6 A Detour to Abreu and Rubinstein [1]

Abreu and Rubinstein [1] study repeated games in which players have lexicographic preferences and can use only pure automata. They consider both the undiscounted game and the discounted game with a discount factor that is close to 1. A pair of pure automata is an *equilibrium* if (a) no player can profit by deviating to any other pure automaton, and (b) a player who deviates to a smaller automaton loses.

Abreu and Rubinstein [1] prove that the set of equilibrium payoffs is the set of feasible and individually rational payoff vectors that can be generated by a coordinated play.

In the Prisoner's Dilemma (see Figure 1) the min-max level of each player is 1, and the punishing action of each player is D. The set of feasible and (weakly) individually rational payoffs appear in Figure 1. It is equal to the quadrilateral W with extreme points $(1,1)$, $(1,3\frac{2}{3})$, $(3,3)$ and $(3\frac{2}{3},1)$. The result of Abreu and Rubinstein implies that the set of equilibrium payoffs is the union of the two line segments $(1,1)-(3,3)$ and $(1,3)-(3,1)$.

The argument leading to the result of Abreu and Rubinstein's [1] are the following.

1. When Player 1 uses an automaton with m states, Player 2's best response is an automaton with at most m states. The reason is that given the automaton of Player 1, Player 2's optimization problem reduces to a Markov decision problem with m states. In such a problem, the decision maker has a stationary pure optimal strategy, which can be implemented by an automaton with at

most m states. This property implies that when the players have lexicographic preferences, in an equilibrium both players use automata of the same size.

2. Each player's equilibrium automaton uses distinct states until it completes one cycle of its states. This follows from a result that says that if the states of one player which are used in any two periods t and t' of equilibrium play are identical, then the average payoff of the opponent between stages t to t' coincides with the average payoff from t' onwards, and therefore also the average payoff from t onwards. Therefore if the cycle starts before all the states for both player are used, players could modify their machine to skip the stages between stages t and t'. This modification does not change the long-run average payoff, yet it lowers the number of states of the automaton, contradiction with the equilibrium condition.

3. If in stage t the automaton P_i plays the same action it plays in stage t', then in stage t the automaton P_{3-i} plays the same action it plays in stage t'. Indeed, by Point 2, the automaton P_i uses different states along its cycle, and in particular the states that are used in stages t and t' are used only in those stages. Assume by way of contradiction that the automaton P_{3-i} plays differently in stages t and t'. Then player $3-i$ can lower the size of his automaton by using the same state in stages t and t', and letting the action of player $3-i$ control the transition out of this state. But this contradicts the equilibrium condition.

Abreu and Rubinstein's equilibrium construction is as follows.

- The players start by implementing a *punishment phase*: both players play the action D for a large number of stages. The states used for this phase are all distinct. Moreover those states are used only at the beginning of the equilibrium play. They could be understood as a signal of strength.

- A cycle of action pairs, which is called the *regular phase*, is repeated. The cycle starts after m_i stages using all the states once. The states used in the cycle and not in the punishment phase are used infinitely many times. Each of them may start a punishment going to the first state if a deviation is detected. The action pairs of the cycle conforms a coordinated play. This implies that there exists a one-to-one relationship between the action set of Players 1 and 2 in equilibrium.

A crucial assumption to the result of Abreu and Rubinstein is that players use pure automata. As the following example shows, the result does not hold when mixed automata are allowed.

Example 6 *Consider the two-player game that appears in Figure 2, where each player has three actions.*

		Player 2		
		A	*B*	*D*
Player 1	*A*	1, 1	3, 2	0, 0
	B	3, 2	1, 1	0, 0
	D	0, 0	0, 0	0, 0

Figure 2: The Game in Example 6.

The minmax value of each player is 0, the set of feasible and individually rational payoffs is the triangle W that appears in Figure 3, and the set of equilibrium payoffs when players are restricted to pure automata and have lexicographic preferences is composed of the two line segments $(0,0)-(1,1)$ and $(0,0)-(3,2)$, see Figure 3.

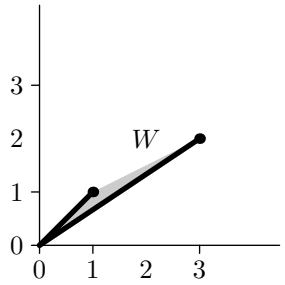


Figure 3: The feasible and individually rational payoffs in Example 6.

We claim that when the players have lexicographic preferences and they are restricted to mixed automata, there are equilibrium payoffs that are arbitrarily close to $(\frac{3}{2},1)=\frac{1}{2}(1,1)+\frac{1}{2}(3,2)$. To this end, for each action $a\in\{A,B\}$ we define the pure automaton $P_1(a)$ of Player 1, that depends on a positive integer k (see Figure 4).

- *The automaton starts by playing a punishment phase of length k^2. That is, the automaton outputs the action D and it expects the other player to play the action D; that is, it moves to the next state if the other player played the action D.*

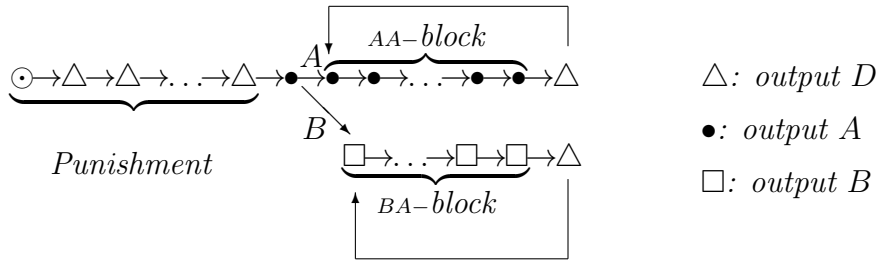
- *In stage k^2+1 it plays the action a, and observes the action played by the other player at that stage.*

- *In the following k stages it repeats the action that the other player played at stage k^2+1, and it expects the other player to play the action a in those stages.*

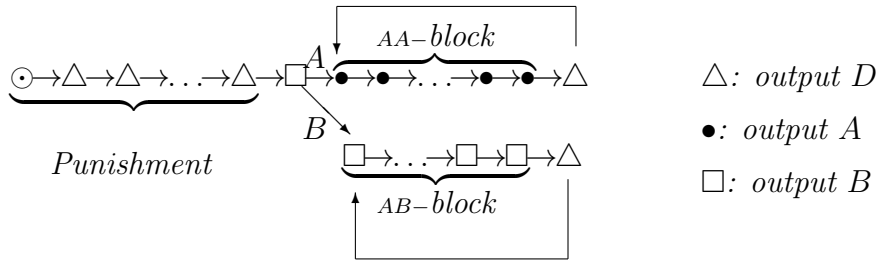
- *Finally, in stage $k^2+1+k+1$ it marks the end of the regular phase by playing D, and, if the other player plays the action D as well, it moves to the state in which it has been at stage k^2+2.*

- *Every deviation from this plan triggers a punishment phase.*

Put differently, the players jointly select one of the four non-zero entries in the payoff matrix, by having each player choose the action of the other player. The size of this automaton is $k^2+1+2(k+1)$.



The automaton $P_1(A)$



The automaton $P_1(B)$

Figure 4: The automata $P_1(A)$ and $P_1(B)$.

Let M_1 be the mixed automaton of Player 1 that plays each of $P_1(A)$ and $P_1(B)$ with equal probabilities. Let $P_2(A)$, $P_2(B)$, and M_2 be the analog automata of Player 2. We argue that (M_1, M_2) is an equilibrium when the players have lexicographic preferences. Indeed, under (M_1, M_2) the automata never restart and the long-run average payoff is $\frac{k-1}{k}\left(\frac{1}{2}(3,2)+\frac{1}{2}(1,1)\right)$.

We finally show that if, say, Player 1, deviates to a smaller automaton, he loses. Let then P_1' be a pure automaton of Player 1. To play well against M_2, the automaton P_1' should play well both against $P_2(A)$ and against $P_2(B)$. In particular, since in the first k^2 stages the play is coordinated, and since in stage k^2+1 the automaton P_1' should play an action which is not D, the automaton P_1' must devote k^2 states to pass the punishment phase.[4,5] *Since M_2 chooses between $P_2(A)$ and $P_2(B)$ with equal probabilities at stage k^2+1, Player 1 is indifferent between playing A and B at that stage, and by symmetry if he can profit by playing one of these actions, he can profit by playing the other as well. Assume then w.l.o.g. that P_1' plays the action B in stage k^2+1. In the following $k+1$ stages the play consists of a coordinated play of $k+1$ stages in which Player 1 is supposed to play A against $P_2(A)$ and B against $P_2(B)$, followed by a final stage in which it should play D, and any deviation triggers a long punishment phase. Since the transition from the last state, in which Player 1 plays D, is different when facing $P_2(A)$ and $P_2(B)$, the automaton P_1' must devote $2(k+1)$ states to succeed passing this play against both $P_2(A)$ and $P_2(B)$.*

Overall, to attain a payoff $\frac{k-1}{k}\left(\frac{1}{2}(3,2)+\frac{1}{2}(1,1)\right)$ against M_2 the automaton P_1' must be of size at least $k^2+1+2(k+1)$, as claimed.

3 BCC Equilibria in the Prisoner's Dilemma

In the present section we prove Theorem 5 for the Prisoner's Dilemma. The construction in this case contains all the ingredients of the general case, yet the simplicity of the Prisoner's Dilemma allows one to concentrate on the construction's main aspects. In Section A we generalize this basic construction to general two-player repeated games.

Consider, for example, the payoff vector $x=(\frac{7}{6},\frac{19}{6})$. This vector can be written as a convex combination of three vectors in the payoff matrix as follows:

$$(\tfrac{7}{6},\tfrac{19}{6})=\tfrac{1}{6}(1,1)+\tfrac{2}{6}(3,3)+\tfrac{3}{6}(0,4). \tag{1}$$

We now describe a c-BCC equilibrium with payoff $(\frac{7}{6},\frac{19}{6})$, for a properly chosen c. Our construction depends on a parameter k that determines the size of the automata that the players use: Player 1 mixes between pure automata of size k^3+2k^2+k+1 and Player 2 mixes between pure automata of size k^3+2k^2+k+4. We will choose k to be sufficiently large so that $\min\{x_1-1,x_2-1\}>\frac{6}{k}$. To facilitate calculations we assume that k is divisible by 4 and that it is sufficiently large so that $k^3>3k^2+2k+8$.

[4] We elaborate on this issue in Section 3.2.

[5] The punishment phase could have been significantly shorter. To make the construction here similar to subsequent constructions, we chose to have a long punishment phase here as well.

3.1 The Equilibrium Play

The equilibrium play will be independent of the pair of pure automata that the players choose and it will consist of three phases, as follows.

- A *Punishment Phase* that consists of k^3 times playing (D, D):

$$Q^* := k^3 \times (D, D).$$

- A *Babbling Phase* that consists of $2k + 1$ blocks: in odd blocks (except the last one) the players play k times (C, C); in even blocks they play k times (D, D); and in the last block the players play $k + 1$ times (C, C).

$$B^* := \sum_{n=1}^{k} \big(k \times (C, C) + k \times (D, D)\big) + (k + 1) \times (C, C).$$

- A *Regular Phase* in which the players repeatedly play actions along which the average payoff is the target payoff x.

$$R^* := 1 \times (D, D) + 2 \times (C, C) + 3 \times (C, D).$$

.

Formally, the equilibrium play path ω^* is

$$\begin{aligned} \omega^* &:= Q^* + B^* + \sum_{n=1}^{\infty} R^* \qquad (2) \\ &= \underbrace{k^3 \times (D, D)}_{\text{Punishment}} + \underbrace{\sum_{n=1}^{k} \big(k \times (C, C) + k \times (D, D)\big) + (k + 1) \times (C, C)}_{\text{Babbling}} + \underbrace{\sum_{n=1}^{\infty} R^*}_{\text{Regular}} . \end{aligned}$$

To implement other feasible and individually rational payoff vectors x as c-BCC equilibria we change the regular phase to contain a cycle of action pairs whose average payoff is close to x.

The roles of the three phases are as follows.

- As in Abreu and Rubinstein [1], the punishment phase ensures that punishment is on the equilibrium path. Because the players minimize their automaton size, subject to maximizing their payoff, if the punishment phase was off the equilibrium path, players could save states by not implementing it. But if a player cannot implement punishment, the other player may safely deviate, knowing

that she will not be punished. In our construction, detectable deviations of the other player will lead the automaton to restart and reimplement ω^*, thereby initiating a long punishment phase. The length of the punishment phase, k^3, is much longer than the babbling phase to ensure that the punishment is severe.

- The babbling phase serves two purposes. First, because it is coordinated, it is not difficult to calculate its complexity for each player i, that is, the size of the minimal pure automaton of player i that can implement player i's part in this sequence, given that the other player, player $3-i$, plays his part in the sequence. As we show below, the complexity of ω^* for Player 1 is k^3+2k^2+k+1 and its complexity for Player 2 is k^3+2k^2+k+4. This implies in particular that if a player deviates to an automaton smaller than the complexity of the sequence for that player, while the other player does not deviate, then there will be a stage in which that player's play deviates from ω^*.

  Second, the babbling phase is sufficiently long, so that to implement the regular phase one does not need new states, but can rather reuse states that implement the babbling phase. Moreover, its long length ensures that, if the states that are reused are chosen randomly, to find which states are reused with non-negligible probability, the other player must use a prohibitively large automaton: to profit by deviating the other player needs to search for the reused states, a task that requires a significantly larger automaton than the one she currently uses.

- On the equilibrium path the regular play will be played repeatedly, so that the long-run average payoff will be the average payoff along R^*, which is $(\frac{7}{6}, \frac{19}{6})$.

3.2 The Complexity of ω^*

We say that a pure automaton P_i of player i is *compatible* with the play ω^* (or that the play ω^* is compatible with the automaton P_i) if, when the other player $3-i$ plays her part in ω^*, the automaton generates the play of player i in ω^*. A mixed automaton M_i of player i is compatible with ω^* if all the pure automata in its support are compatible with ω^*.

Plainly, different automata may be compatible with ω^*. The *complexity of* ω^* *w.r.t. player* i is the size of the smallest pure automaton of player i that is compatible with ω^*. This concept was first defined and studied by Neyman [12], who also provided a simple way to calculate it.

We elaborate on the concept of the complexity of a sequence in Section 3.6. The following Lemma provides the complexity of the sequence ω^* w.r.t. the two players.

Lemma 7 *The complexity of* ω^* *w.r.t. Player 1 is* k^3+2k^2+k+1*, and its complexity w.r.t. Player 2 is* k^3+2k^2+k+4*.*

In Section 3.6 we will prove that the complexity of ω^* w.r.t. each player is at least the quantity given by Lemma 7. In Section 3.3 we provide a pure automaton for Player 1 with k^3+2k^2+k+1 states that is compatible with ω^*, and in Section 3.4 we provide a pure automaton for Player 2 with k^3+2k^2+k+4 states that is compatible with ω^*, thereby completing the proof of Lemma 7.

3.3 An Automaton P_1 for Player 1 that is Compatible with ω^*

Fix $j \in \{1,2,\ldots,k-1\}$ and $h_1, h_2 \in \{4,5,\ldots,k\}$ such that $h_1 \neq h_2$. In this section we define a pure automaton $P_1 = P_1^{j,h_1,h_2}$ for Player 1 with size k^3+2k^2+k+1 that is compatible with ω^*; that is, when Player 2 plays his part in ω^*, the generated play is ω^*.

Denote the states of P_1 by the integers $Q = \{1,2,\ldots,k^3+2k^2+k+1\}$, where $q^* = 1$ is the initial state. We will construct the automaton P_1 in four steps.

3.3.1 Step 1: Implementing the Punishment and Babbling Phases

The Punishment and Babbling Phases, whose total length is k^3+2k^2+k+1, are

$$\omega_1 = k^3 \times (D,D) + \sum_{n=1}^{k} \big(k \times (C,C) + k \times (D,D)\big) + (k+1) \times (C,C).$$

The length of these phases is similar to the size of the automaton that we construct. A naive implementation is to have one state for each action of Player 1 in ω_1: state $q \in Q$ will implement the q'th action pair in ω_1. Formally, we divide Q into three sets:

1. $Q^P = \{1,2,\ldots,k^3\}$ is the set of all states that implement the Punishment Phase.
2. $Q^C = \Big(\bigcup_{n=0}^{k-1}\{k^3+2nk+1,\ldots,k^3+2nk+k\}\Big) \cup \{k^3+2k^2+1,\ldots,k^3+2k^2+k+1\}$ is the set of states in all *C-blocks*.
3. $Q^D = \bigcup_{n=0}^{k-1}\{k^3+2nk+k+1,\ldots,k^3+2nk+2k\}$ is the set of states in all *D-blocks*.

The output function is

$$f(q) = \begin{cases} D & q \in Q^P \cup Q^D, \\ C & q \in Q^C, \end{cases}$$

and the transition function is

$$g(q, f(q)) = q + 1, \quad 1 \leq q < k^3 + 2k^2 + k + 1.$$

Because the play in ω_1 is coordinated, the transition is defined only if Player 2 complies with the desired play ω_1. Figure 5 illustrates the first step in the construction of the automaton P_1. In this figure, the initial state is the dotted circle to the left, the white squares correspond to states where the action played is D, and the black circles correspond to states where the action played is C.

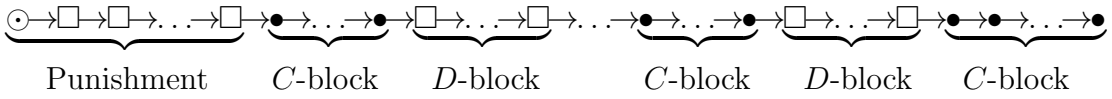


Figure 5: An implementation of ω_1.

3.3.2 Step 2: Implementing the Regular Phase.

We now add to the automaton P_1 transitions that implement the regular play, which is $(D, D) + 2 \times (C, C) + 3 \times (C, D)$. The implementation will use states in the j_1'th D-block and in the (j_1+1)'th C-block. Specifically, the last state in the j_1'th D-block is used to implement the first action pair of the regular play (which is (D, D)); the first two states in the $(j_1 + 1)$'th C-block are used to implement the next two action pairs of the regular play (both of which are (C, C)); the third state in the $(j_1 + 1)$'th C-block, as well as the h_1'th and h_2'th states, are used to implement the subsequent action pair of the regular play (which are (C, D)).

Formally, we add the following transitions (see Figure 6):

$$\begin{aligned} g(k^3 + 2k^2 + k + 1, C) &= k^3 + 2j_1k, && (3)\\ g(k^3 + 2j_1k + 3, D) &= k^3 + 2j_1k + h_1, && (4)\\ g(k^3 + 2j_1k + h_1, D) &= k^3 + 2j_1k + h_2, && (5)\\ g(k^3 + 2j_1k + h_2, D) &= k^3 + 2j_1k. && (6) \end{aligned}$$

We call the three states $k^3 + 2j_1k + 3$, $k^3 + 2j_1k + h_1$, and $k^3 + 2j_1k + h_2$ the *accept-all states* of the automaton P_1. In Figure 6, the three accept-all states are denoted by triangles. When the automaton P_1 is at such a state it plays the action C; if Player 2 plays the action C, the transition is to the subsequent (black circle) state, whereas if Player 2 plays D, the transition is to the next triangle state.

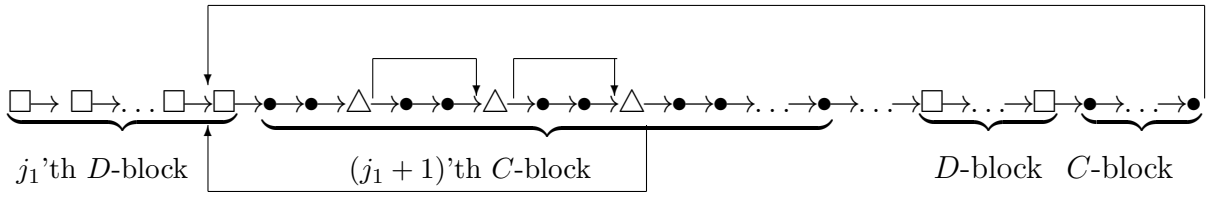


Figure 6: The j_1'th D-block and the (j_1+1)'th C-block in P_1.

3.3.3 Step 3: Deviations.

By construction, the automaton P_1 is compatible with ω^*. We now add transitions to detect deviations of Player 2 as follows: all transitions that were not defined in Steps 1-3 lead to state 1.

Only the three accept-all states accept both actions of Player 2; the other states accept only the action that they output. Because a punishment phase of length k^3 begins in state 1, any deviation in a non-accept-all state is followed by a severe punishment.

3.3.4 A Mixed Automaton $M_1 = M_1(k)$

The pure automaton $P_1 = P_1(j, h_1, h_2)$ that was constructed in Section 3.3.1–3.3.3 depends on three parameters: j, h_1, and h_2. We will now define a mixed automaton $M_1 = M_1(k)$ that chooses these parameters randomly.

Let $\mathcal{H} = \{(j^d, h_1^d, h_2^d) \colon 1 \le d \le \frac{k}{4}\}$ be a collection of $\frac{k}{4}$ triplets that satisfy the following conditions:

A1 $(j^d)_{d=1}^{k/4}$ are distinct elements from $\{1, 2, \ldots, k-1\}$, $(h_1^d)_{d=1}^{k/4}$ are distinct elements from $\{4, 5, \ldots, k\}$, and $(h_2^d)_{d=1}^{k/4}$ are distinct elements from $\{4, 5, \ldots, k\}$.

A2 $\{h_1^{d_1}, h_2^{d_1}\} \cap \{h_1^{d_2}, h_2^{d_2}\} = \emptyset$ for every distinct $d_1, d_2 \in \{1, 2, \ldots, \frac{k}{4}\}$.

A3 $h_2^{d_1} - h_1^{d_1} \neq h_2^{d_2} - h_1^{d_2}$ for every distinct $d_1, d_2 \in \{1, 2, \ldots, \frac{k}{4}\}$.

One can define, e.g., $j^d = d$, $h_1^d = 3 + d$ and $h_2^d = h_1^d + \frac{k}{4} + d$ for every $d \in \{1, 2, \ldots, \frac{k}{4}\}$.

The mixed automaton $M_1 = M_1(k)$ chooses uniformly one of the pure automata $\{P_1^{j,h_1,h_2}, (j, h_1, h_2) \in \mathcal{H}\}$. In particular, all pure automata in the support of M_1 are compatible with ω^* for Player 1, so that M_1 is compatible with ω^* for Player 1 as well.

The choice of the parameters j, h_1, and h_2 that define these pure automata was done to ensure that, when Player 2 deviates from ω^*, at most one of the automata in $\{P_1^{j,h_1,h_2}, (j, h_1, h_2) \in \mathcal{H}\}$ will not restart. This property is stated in the following Lemma.

Lemma 8 *Let $P_1 = P_1^{j,h_1,h_2}$ and $P_1' = P_1^{j',h_1',h_2'}$ be two different pure automata in the support of M_1 and let P_2 be any pure automaton of Player 2. Let t be the first stage in which the play under (P_1, P_2) differs from ω^*. Then at least one of the automata P_1 and P_1' restarts before stage $t + 2k^2 + k + 1$.*

Note that since both P_1 and P_1' are compatible with ω^*, the first stage in which the play under (P_1', P_2) differs from ω^* is also t. The Lemma is valid for any strategy of Player 2, and not necessarily only for those implementable by pure automata. Conditions (A1)–(A3) required from the parameters of the automata in the support of M_1 are the key ingredient in the proof of Lemma 8.

Proof. Denote by $q(t)$ (resp. $q'(t)$) the state of the automaton P_1 (resp. P_1') when facing P_2. Denote by $\omega^*(t)$ the action pair at stage t according to ω^*. Then $\omega_2^*(t)$ is the action that Player 2 is supposed to play at stage t according to ω^*.

Since P_1 and P_1' are compatible with ω^*, and since in stage t the play under (P_1, P_2) differs from ω^*, it follows that in stage t the pure automaton P_2 does not play the action $\omega_2^*(t)$. If $q(t)$ (resp. $q'(t)$) is not an accept-all state, then the automaton P_1 (resp. P_1') restarts at stage t, and the lemma follows. Thus, we assume from now on that both $q(t)$ and $q'(t)$ are accept-all states.

In which stages do both P_1 and P_1' visit an accept-all state? During the punishment phase none of these automata visits an accept-all state, and since $j_1 \neq j_1'$, during the implementation of the babbling phase they do not visit an accept-all states at the same stage. Thus, only in the regular phase both automata visit accept-all states simultaneously, when implementing the action pairs (C, D). We will show that if P_2 deviates when P_1 implements either one of these action pairs, a punishment phase will ensue in at most $2k^2 + k + 1$ stages. This will follow because the accept-all states of P_1 lie in a different C-block than the accept-all states of P_1'.

Suppose first that state $q(t)$ is the h_1'th state of the j_1'th C-block. Then $q'(t)$ is the h_1''th state of the j_1''th C-block. Since P_2 deviates in stage t, it plays C instead of D, so that $q(t+1) = q(t) + 1$ and $q'(t+1) = q'(t) + 1$. The automaton P_1 expects now the sequence $(k - h_1) \times (C, C) + 1 \times (D, D)$ and is going to visit an accept-all state in $h_2 - h_1$ stages. Similarly, the automaton P_1' expects now the sequence $(k - h_1') \times (C, C) + 1 \times (D, D)$ and is going to visit an accept-all state in $h_2' - h_1'$ stages. By (A2) and (A3) we have $h_1 \neq h_1'$ and $h_2 - h_1 \neq h_2' - h_1'$, and therefore no sequence of actions that P_2 can generate is compatible with both automata, hence at least one of them will restart within at most k stages.

The argument is similar if state $q(t)$ is the h_2'th stage of the j_1'th C-block.

It is left to handle the case in which state $q(t)$ is the third stage of the j_1'th C-block, in which case state $q'(t)$ is the third stage of the j_1''th C-block. The automaton

P_1 expects now the sequence

$$\omega := (k-3)\times(C,C)+k\times(D,D)+\sum_{j=j_1+2}^{k}(k\times(C,C)+k\times(D,D))+(k+1)\times(C,C),$$

and visits two accept-all states in h_1-3 and h_2-3 stages. Similarly, the automaton P_1' expects the sequence

$$\omega' := (k-3)\times(C,C)+k\times(D,D)+\sum_{j=j_1'+2}^{k}(k\times(C,C)+k\times(D,D))+(k+1)\times(C,C),$$

and visits two accept-all states in $h_1'-3$ and $h_2'-3$ stages. By (A2) and (A3) we have $h_1 \neq h_1'$, $h_2 \neq h_2'$, and $j_1 \neq j_1'$, and therefore no sequence of actions that P_2 can generate is compatible with both automata, hence at least one of them will restart within at most $2k^2+k+1$ stages. The proof is thus completed. ■

Remark 9 *Lemma 8 assumes that both automata start at state 1. However, the reader can verify that the proof is valid as soon as the two automata start at the same state; that is, it holds whenever $q(1) = q'(1)$.*

3.4 An Automaton P_2 for Player 2 that is Compatible with ω^*

As in Section 3.3 we define a family of pure automaton for Player 2, which are compatible with ω^* and have size k^3+2k^2+k+4. As for player 1, the automata in the family depend on two parameters, an integer $j \in \{1,2,\ldots,k-1\}$ and a set H of integers.

3.4.1 Step 1: Implementing the Punishment and Babbling Phases.

We start by implementing the Punishment and Babbling Phases whose length is k^3+2k^2+k+1 states.

$$\omega_2 = k^3\times(D,D)+\sum_{n=0}^{k-1}(k\times(C,C)+k\times(D,D))+(k+1)\times(C,C),$$

As for player 1, we define an automaton that implements each action pair in one state. Let $Q=\{1,2,\ldots,k^3+2k^2+k+4\}$ be the set of states of the automaton with $q^*=1$ the initial state. The sets Q^P, Q^C, and Q^D of the states that implement the

Punishment Phase, the C-blocks, and the D-blocks, respectively, are defined as in Step 1 in Section 3.3.

The output function is:

$$f(q) = \begin{cases} D & q \in Q^P \cup Q^D, \\ C & q \in Q^C, \end{cases}$$

and the transition function is

$$g(q, f(q)) = q + 1, \quad 1 \leq q < k^3 + 2k^2 + k + 1.$$

3.4.2 Step 2: Implementing the play $(D, D) + 2 \times (C, C)$ of the Regular Phase.

We now add the transitions that implement the next three actions pairs in ω^*, which are $\omega_5 = (D, D) + 2 \times (C, C)$. To this end we use the last three states of Q.

The action function for these states is given by

$$f(k^3 + 2k^2 + k + 2) = D; f(k^3 + 2k^2 + k + 3) = C; f(k^3 + 2k^2 + k + 4) = C$$

and the transition is given by

$$g(q, f(q)) = q + 1, \quad k^3 + 2k^2 + k + 1 \leq q < k^3 + 2k^2 + k + 4.$$

3.4.3 Step 3: Implementing the play $3 \times (C, D)$ of the Regular Phase.

The last part of the Regular Phase is implemented by reusing states of the Babbling Phase. Fix $j \in \{1, \ldots, k\}$ and $h_1, h_2, h_3 \in \{1, 2, \ldots, k\}$ such that $h_1 < h_2 < h_3$. We will reuse the h_1'th, h_2'th, and h_3'th states of the j'th D-block. Formally,

$$\begin{aligned} g(k^3 + 2k^2 + k + 4, C) &= k^3 + 2kj + h_1 && (7) \\ g(k^3 + 2kj + h_1, C) &= k^3 + 2kj + h_2, && (8) \\ g(k^3 + 2kj + h_2, C) &= k^3 + 2kj + h_3. && (9) \end{aligned}$$

Finally, from the last state $k^3 + 2kj + h_3$ the Regular Phase should be repeated, so that we define

$$g(k^3 + 2kj + h_3, C) = k^3 + 2k^2 + k + 2.$$

The three reused states $k^3 + 2kj + h_1$, $k^3 + 2kj + h_2$, and $k^3 + 2kj + h_3$ are called the *accept-all* states of P_2^{j,h_1,h_2,h_3}.

3.4.4 Step 4: Deviations.

As for Player 1 we add transitions to handle deviations in states that are not accept-all states. All transitions that are not defined in Steps 1–3 lead to state 1, so that such deviations initiate a long punishment phase.

3.4.5 Mixed strategy of player 2.

The definition of the mixed strategy M_2 is analog to that of M_1. The pure automaton $P_2 = P_2(j, h_1, h_2, h_3)$ that was constructed in Sections 3.4.1–3.4.4 depends on four parameters j, h_1, h_2 and h_3. We will now define a mixed automaton $M_2 = M_2(k)$ that chooses these parameters randomly.

Let $\mathcal{H} = \{(j^d, h_1^d, h_2^d, h_3^d) \colon 1 \leq d < \frac{k}{4}\}$ be a collection of $\frac{k}{4}$ triplets that satisfy the following conditions:

B1 $(j^d)_{d=1}^{k/4}$ are distinct elements from $\{1, 2, \ldots, k-1\}$, and for each $l \in \{1, 2, 3\}$, $(h_l^d)_{d=1}^{k/4}$ are distinct elements from $\{1, 2, \ldots, k\}$.

B2 $\{h_1^{d_1}, h_2^{d_1}, h_3^{d_1}\} \cap \{h_1^{d_2}, h_2^{d_2}, h_3^{d_2}\} = \emptyset$ for every distinct $d_1, d_2 \in \{1, 2, \ldots, \frac{k}{4}\}$.

B3 For every distinct $d_1, d_2 \in \{1, 2, \ldots, \frac{k}{4}\}$ the six numbers $h_2^{d_1} - h_1^{d_1}$, $h_2^{d_2} - h_1^{d_2}$, $h_3^{d_1} - h_2^{d_1}$, $h_3^{d_2} - h_2^{d_1}$, $h_3^{d_1} - h_1^{d_1}$, and $h_3^{d_2} - h_1^{d_1}$ are distinct.

One can define, e.g., $j_1^d = d$, $h_1^d = d$, $h_2^d = 2d + \frac{k}{4}$, and $h_3^d = 3d + 3\frac{k}{4}$, for every $d \in \{1, 2, \ldots, \frac{k}{4}\}$.

The mixed automaton $M_2 = M_2(k)$ chooses uniformly one of the pure automata $\{P_2^{j,h_1,h_2,h_3}, (j, h_1, h_2, h_3) \in \mathcal{H}\}$. As for Player 1, all pure automata in the support of M_2 are compatible with ω^* for Player 2, so that M_1 is compatible with ω^* for Player 1 as well. The analog of Lemma 8 is the following.

Lemma 10 *Let $P_2 = P_2^{j,h_1,h_2,h^3}$ and $P_2' = P_2^{j',h_1',h_2',h_3'}$ be two different pure automata in the support of M_2 and let P_1 be any pure automaton of Player 1. Let t be the first stage in which the play under (P_1, P_2) differs from ω^*. Then at least one of the automata P_2 and P_2' restarts before stage $t + 2k^2 + k + 1$.*

3.5 (M_1, M_2) is a c-BCC Equilibrium

In this section we prove that (M_1, M_2) is a c-BCC equilibrium, provided the cost of memory C is neither too high nor too low. Note that if c is very low, switching to a significantly larger automaton may not be too costly, while if c is very high, the cost of the automaton M_i, which is $c|M_i|$, is high, so that the players will profit by deviating to a small automaton, thereby saving the cost of the automaton. Here

we will prove that if $\frac{12}{k^4} < c < \frac{\eta}{4k^3}$ then (M_1, M_2) is a c-BCC equilibrium, where $\eta < \min\{x_1^* - 1, x_2^* - 1\}$.

By construction, the average payoff under (M_1, M_2) is

$$\gamma(M_1, M_2) = \tfrac{1}{6}u(D, D) + \tfrac{2}{6}u(C, C) + \tfrac{3}{6}u(C, D) = (\tfrac{7}{6}, \tfrac{19}{6}).$$

How can a player increase his payoff? To this end he needs to learn which states are the accept-all states of the other player's realized pure automaton. In our construction each state is reused by at most one pure automaton, and therefore learning the reused states essentially means enumerating over all possible reused states. There are $O(k)$ different possibilities for reused states, and each failed attempt requires k^3 new states to pass the ensuing punishment phase. It therefore follows that to successfully learn the reused states the deviator needs to use a memory size of the order of k^4. The relation between c and k will ensure that such a deviation is not profitable. We now turn this intuition into a formal argument.

3.5.1 A Lower Bound on the Size of Player 2's Automaton that can Gain Against M_1

In the present section we provide a lower bound on the size of an automaton P_2 of Player 2 that profits when facing M_1. As we will see, the size of such an automaton P_2 will be larger than $O(k^3)$, the complexity of ω^*.

Denote by $(P_1^d)_{d=1}^{k/4}$ the pure automata in the support of M_1. Suppose that the players use the automata (P_1^d, P_2). Denote by $q_2(t; P_1^d)$ the state of the automaton P_2 at stage t when it faces the automaton P_1^d.

If P_2 is not compatible with ω^* for Player 2, then P_1^d restarts whenever a deviation from ω^* is detected, and a punishment phase starts. Denote by t_n^d the stage at the n'th time in which P_1^d visits state 1 when facing P_2, that is, the stage in which the n'th punishment phase starts:

$$\begin{aligned} t_1^d &:= 1, \\ t_{n+1}^d &:= \min\left\{t > t_n^d \colon q_1(t) = 1\right\}, \qquad n \geq 1. \end{aligned}$$

By convention, the minimum of an empty set is ∞.

There are two scenarios in which Player 2 may improve her long-run average payoff. One possibility is if there exists n such that $t_n^d < \infty = t_{n+1}^d$. Then t_n^d is the last stage in which the automaton P_1^d restarts. If the play after stage t_n^d is different from ω^*, it means that Player 2 plays as if she knows (some of) the parameters that determine P_1^d, and she might use this information to improve her payoff. Another

possibility is that $(t_n^d)_{n\in\mathbb{N}}$ are finite and between two of these stages the average payoff of Player 2 is higher[6] than x_2^*.

This leads us to the following definition. For every $d \in \{1, 2, \ldots, \frac{k}{4}\}$ and every $n \in \mathbb{N}$ let ω_n^d be the play generated from stage n and on under (P_1^d, P_2).

Definition 11 *The automaton* P_2 fools *the automaton* P_1^d *if either one of the following conditions hold when the automata* (P_1^d, P_2) *face each other.*

C1) There is $n_0 \in \mathbb{N}$ *such that* $t_{n_0}^d < \infty = t_{n_0+1}^d$ *and* $\omega_{n_0}^d \neq \omega^*$.

C2) $t_n^d < \infty$ *for every* $n \in \mathbb{N}$*, and there is* $n_0 \in \mathbb{N}$ *such that the average payoff for Player 2 between stages* $t_{n_0}^d$ *and* $t_{n_0+1}^d - 1$ *is strictly higher*[7] *than* x_2^*.

Since the punishment phase lowers the average payoff, provided that k is sufficiently large, if Condition C2 holds, then the play between stages $t_{n_0}^d$ and $t_{n_0+1}^d - 1$ is not a prefix of ω^*.

Neither C1 nor C2 imply that the long-run average payoff under (P_1^d, P_2) is higher than x_2^*. Yet, as the next lemma shows, the converse is true: if the long-run average payoff of Player 2 under (P_1^d, P_2) exceeds x_2^*, then P_2 must have fooled P_1^d.

Lemma 12 *If* P_2 *does not fool* P_1^d *then* $\gamma_2(P_1^d, P_2) \leq x_2^*$.

Proof. Since both P_1^d and P_2 are automata, $\gamma_2(P_1^d, P_2)$, which is the long-run average payoff of Player 2 under (P_1^d, P_2), exists. Suppose first that $t_n^d < \infty$ for every $n \in \mathbb{N}$. Since P_2 does not fool P_1^d, for every $n \in \mathbb{N}$ the average payoff of Player 2 between stages t_n^d and $t_{n+1}^d - 1$ is at most x_2^*, and therefore $\gamma_2(P_1^d, P_2) \leq x_2^*$.

Suppose now that there is $n_0 \in \mathbb{N}$ such that $t_{n_0}^d < \infty = t_{n_0+1}^d$. Since P_2 does not fool P_1^d, we have $\omega_{n_0}^d = \omega^*$, so that $\gamma_2(P_1^d, P_2) = x_2^*$, and the result follows. ■

If condition C1 holds, we say that P_2 fools P_1^d in stages $\{t_{n_0}^d, t_{n_0}^d + 1, \ldots\}$. If condition C2 holds, we say that P_2 fools P_1^d in stages $\{t_{n_0}^d, t_{n_0}^d + 1, \ldots, t_{n_0+1}^d - 1\}$. In both cases[8] we set $t_*^d = t_{n_0}^d$, and we say that at stage t_*^d Player 2 starts to fool P_1^d. Denote by $R_d = \{q_2(t_*^d; P_1^d), q_2(t_*^d + 1; P_1^d), \cdots, q_2(t_*^d + k^3 - 1; P_1^d)\}$ the k^3 states that P_2 visits at the beginning of the period in which it fools P_1^d. During these stages the automaton P_1^d executes the punishment phase, and the payoff of Player 2 is low. We now prove that the sets $(R_d)_{l=1}^{k/4}$ are disjoint, thereby bounding from below the size of any automaton of Player 2 that obtains high payoff when facing M_1.

[6] In fact, if $(t_n^d)_{n\in\mathbb{N}}$ are finite then, so that Player 2 improves her payoff, the average payoff between t_n^d and $t_{n+1}^d - 1$ should be higher than x^* infinitely often.

[7] Observe that in this case $t_{n_0+1}^d \geq t_{n_0}^d + k^3$. In fact, a stronger bound can be obtained.

[8] If condition C2 holds, there may be several stages n_0 at which P_2 starts to fool P_1^d. In such a case we choose one of them arbitrarily.

Lemma 13 *Let $1 \leq d_1 < d_2 \leq \frac{k}{4}$. If P_2 fools both $P_1^{d_1}$ and $P_1^{d_2}$, then $R_{d_1} \cap R_{d_2} = \emptyset$.*

An immediate corollary of Lemma 13 is the following, which is the main result of this section.

Theorem 14 *Denote by L_0 the number of pure automata P_1^d, $1 \leq d \leq \frac{k}{4}$, that P_2 fools. Then $|P_2| \geq L_0 k^3$.*

Proof of Lemma 13.

The proof relies on Lemma 18 that we will prove in Section 3.6, when we discuss complexity of sequences of action pairs.

The first $k^3 + 1$ action pairs of ω^* are coordinated, and the $(k^3 + 1)$'th action of Player 2 differs from her actions in the first k^3 stages. Lemma 18(3) implies the following:

Fact 1: If P_2 fools P_1^d then the states in R_d are distinct: $|R_d| = k^3$.

Moreover, Lemma 18(3) also implies the following.

Fact 2: If R_{d_1} and R_{d_2} are not disjoint, then the last state in R_{d_1} coincides with the last state in R_{d_2}, that is, $q_2(t_*^{d_1} + k^3 - 1; P_1^{d_1}) = q_2(t_*^{d_2} + k^3 - 1; P_1^{d_2})$.

Indeed, suppose that R_{d_1} and R_{d_2} are not disjoint, and assume that $q_2(t_*^{d_1} + n_1; P_1^{d_1}) = q_2(t_*^{d_2} + n_2; P_1^{d_2})$. We argue that necessarily $n_1 = n_2$. This will imply that the last state in R_{d_1} coincides with the last state in R_{d_2}. Assume then to the contrary that, w.l.o.g., $n_1 < n_2$. Lemma 18(3) implies that $q_2(t_*^{d_1} + n_1 + s; P_1^{d_1}) = q_2(t_*^{d_2} + n_2 + s; P_1^{d_2})$ for every s that satisfies $1 \leq s \leq k^3 - n_2 + 1$. Since P_2 fools $P_1^{d_1}$, the action that P_2 plays in state $q_2(t_*^{d_1} + n_1 + k^3 - n_2 + 1; P_1^{d_1})$ is D. Since P_2 fools $P_1^{d_2}$, the action that P_2 plays in state $q_2(t_*^{d_2} + n_2 + k^3 - n_2 + 1; P_1^{d_2})$ is C. But $q_2(t_*^{d_1} + n_1 + k^3 - n_2 + 1; P_1^{d_1}) = q_2(t_*^{d_2} + n_2 + k^3 - n_2 + 1; P_1^{d_2})$, a contradiction.

We are now ready to prove that $R_{d_1} \cap R_{d_2} = \emptyset$.

Assume to the contrary that R_{d_1} and R_{d_2} are not disjoint. By Fact 2, the last state in R_{d_1} coincides with the last state in R_{d_2}, that is, $q_2(t_*^{d_1} + k^3 - 1; P_1^{d_1}) = q_2(t_*^{d_2} + k^3 - 1; P_1^{d_2})$. Because, for $i = 1, 2$, at stage $t_*^{d_i}$ the automaton P_2 starts fooling $P_1^{d_i}$, it follows that the play under $(P_1^{d_i}, P_2)$ after this stage is different from ω^*. Denote by $t_*^{d_i} + t$ the first stage in which the play under $(P_1^{d_i}, P_2)$ differs from ω^*. Lemma 8 implies that at least one of the automata $(P_1^{d_i})_{i=1,2}$, say the automaton $P_1^{d_1}$, restarts before stage $t_*^{d_i} + t + 2k^2 + k + 1$ (see Comment 9). We argue that P_2 does not fool $P_1^{d_1}$, a contradiction.

Indeed, the play from stage $t_*^{d_1}$ until the automaton $P_1^{d_1}$ restarts consists of

- k^3 stages of the punishment phase, in which Player 2's payoff is 1 per stage;

- $2k^2+k+1$ stages of the babbling phase in which his payoff is at most 4 in each stage;[9]
- several rounds, say r, of the regular phase, in which his average payoff is x_2 per round;
- if deviation occurs in a regular phase, at most 6 stages in a portion of the regular phase, in which the per-period payoff is at most 4;
- and at most k^2+k+1 stages between the deviation and the stage in which $P_1^{d_1}$ restarts, in which his payoff is at most 4 in each stage.

Thus, Player 2's average payoff between stage $t_*^{d_1}$ and the stage in which $P_1^{d_1}$ restarts is at most

$$\frac{k^3\times 1+(3k^2+2k+8)\times 4+6r\times x_2}{k^3+3k^2+2k+8+6r}, \tag{10}$$

which is strictly lower than x_2 provided

$$x_2 > \frac{k^3+12k^2+8k+32}{k^3+3k^2+2k+8} > 1+\frac{6}{k},$$

as we claimed. ■

The analog of Theorem 14 is the following.

Theorem 15 *Let P_1 be a pure automaton of Player 1, and denote by L_0 the number of pure automata P_2^d that P_1 fools. Then $|P_1|\geq L_0k^3$.*

3.5.2 A BCC-Equilibrium

Let $\eta<\min\{x_1^*-1,x_2^*-1\}$. In this section we argue that the pair of automata (M_1,M_2), which was constructed in Sections 3.3 and 3.4, is a c-BCC-equilibrium, provided k is sufficiently large and $\frac{12}{k^4}<c<\frac{\eta}{4k^3}$. We only prove the claims for Player 2. The claims for Player 1 can be proven analogously. Below we denote the state of an automaton of player i at stage t by $q_i(t)$.

Denote the pure automata in the support of M_1 by $P_1^1,P_1^2,\ldots,P_1^{k/4}$. Let j^l and H^l be respectively the parameters j and H of P_1^d, for $d=1,2,\ldots,k_2$. Let P_2 be an arbitrary pure automaton that implements a strategy of Player 2. We denote by ω^l the play that is generated under (P_1^d,P_2).

[9] Or at most $2k^2+k+1$ stages, if deviation occurs during the babbling phase.

Recall that the min-max value in pure strategies of both players is 1. Therefore, $\min\{x_1 - 1, x_2 - 1\} > 0$ is the minimal difference between the target payoff x and the min-max value. We now prove that Player 2 cannot profit by deviating to an automaton smaller than M_2.

Lemma 16 *Assume that k and c satisfy $\frac{24}{k} < \frac{\eta}{2}$ and $c < \frac{\eta}{2k^3}$. Let P_2' be an automaton for Player 2 with size smaller than $k^3 + 2k^2 + k + 4$. Then $\gamma_2(M_1, P_2') - c|P_2'| \leq \gamma_2(M_1, M_2) - c|M_2|$.*

Proof. Because the complexity of ω^* w.r.t. Player 2 is $k^3 + 2k^2 + k + 4$, the play under (P_1^l, P_2') is not ω^*. By Lemma 13, and because the size of P_2 is smaller than $2k^3$, the automaton P_2' can fool at most one of the automata $(P_1^d)_{d=1}^{k/4}$. Because it cannot generate ω^*, any automaton that P_2 does not fool restarts after at most $k^3 + 2k^2 + k$ stages, so that the average payoff is at most $\frac{k^3}{k^3+2k^2+k} + 4\frac{2k^2+k}{k^3+2k^2+k}$. It follows that the expected payoff $\gamma_2(M_1, P_2')$ is at most

$$4\frac{1}{k/4} + \frac{\frac{k}{4} - 1}{\frac{k}{4}}\left(\frac{k^3}{k^3 + 2k^2 + k} + 4\frac{2k^2 + k}{k^3 + 2k^2 + k}\right) \leq 1 + \frac{24}{k} < 1 + \frac{\eta}{2}.$$

Because the size of the automaton M_2 is $k^3 + 2k^2 + k + 1$, the gain of reducing the size of automaton from $|M_2|$ to $|P_2'|$ is at most $c(k^3 + 2k^2 + k)$. So that Player 2 does profit by this deviation, we need to require that

$$x_2^* \geq 1 + \frac{24}{k} + c(k^3 + 2k^2 + k),$$

and therefore it is enough to require that

$$x_2^* - 1 > \eta > \frac{24}{k} + c(k^3 + 2k^2 + k).$$

The right-hand side inequality holds provided

$$c < \frac{\eta - \frac{24}{k}}{k^3 + 2k^2 + k},$$

so it is enough to require that $c < \frac{\eta}{4k^3}$. ■

We finally prove that Player 2 cannot profit by deviating to an automaton larger than M_2.

Lemma 17 *Let P_2' be a pure automaton such that $\gamma_2(M_1, P_2') > x_2$. Then $\gamma_2(M_1, P_2') - c|P_2'| \leq \gamma_2(M_1, M_2) - c|M_2|$, provided $c > \frac{12}{k^4}$.*

Proof. Let L_0 be the number of pure automata $(P_1^d)_{d=1}^{k/4}$ that P_2 fools. Because $\gamma_2(M_1, P_2') > x_2^*$ we have $L_0 \geq 1$. If P_2 fools P_1^d, then Player 2's long-run average payoff is at most 4, the maximal payoff in the game. If P_2 does not fool P_1^d, then Player 1's long-run average payoff is at most x_2^*. The expected long-run average payoff of Player 2 then satisfies

$$\gamma_2(M_1, P_2') \leq 4\frac{L_0}{\frac{k}{4}} + x_2^*\frac{\frac{k}{4} - L_0}{\frac{k}{4}} < x_2^* + 12\frac{L_0}{k}.$$

By Theorem 15 we have $|P_2'| \geq L_0 k^3$, and therefore

$$\gamma_2(M_1, P_2') < x_2^* + 12\frac{L_0}{k} = x_2^* + 12\frac{L_0 k^3}{k^4} \leq x_2^* + |P_2'| \times \frac{12}{k^4}.$$

Therefore, as soon as $c > \frac{12}{k^4}$ Player 2 does not profit by this deviation. ■

To summarize, given the feasible and an individually rational payoff vector x^*, we choose $\eta \in (0, \min\{x_1^* - 1, x_2^* - 1\})$. For every $c > 0$ we define $k = k(c)$ by the equality $c = \frac{\eta}{k^{3.5}}$. Then $\frac{12}{k^4} < c < \frac{\eta}{3k^3}$, provided c is small enough (so that $k(c)$ is large enough). The pair of automata $(M_1(k(c)), M_2(k(c)))$ are then c-BCC equilibrium with payoff x^*.

Let $c > 0$ be sufficiently small, and let $k = k_c$ satisfy $\frac{3}{k^3 k_2} < c < \frac{\eta}{3k^3}$. Then the automata $(M_1(k), M_2(k))$ form a c-BCC equilibrium. Since the size of the automata $M_1(k)$ and $M_2(k)$ are $k^3 + 2k^2 + 1$ and $k^3 + 2k^2 + k + 1$, if for each $k \geq 1$ we set $\widehat{c}_k = \frac{4}{k^3 k_2}$, then $\frac{3}{k^3 k_2} < \widehat{c}_k < \frac{\eta}{2k^3}$ and $\widehat{c}_k M_1(k)$ and $\widehat{c}_k M_2(k)$ are both smaller than $\frac{10}{k_2}$, which goes to 0 as k goes to infinity (and $\widehat{c}_k$ goes to 0). It follows that x^* is a BCC equilibrium payoff.

3.6 The Complexity of a Sequence of Action Pairs

Let ω be a (finite or infinite) sequence of action pairs and let P_i be an automaton of player i that is compatible with ω. Denote by $\text{comp}_i(\omega)$ the complexity of ω w.r.t. player i. In the rest of this subsection we prove that the complexity of ω^*, which is defined in (2), w.r.t. each of the players is at least the quantities given in Lemma 7.

When $\omega = (\omega(t))_t$ is a (finite or infinite) sequence of action pairs, we denote by $\omega_i(t)$ player i's action in time t at ω. Recall that $q_i(t)$ is the state of the automaton P_i at time t.

The following lemma lists several simple observations that we will use in the sequel. The first property says that if the action that P_i plays in stage t_1 differs from the action it plays in stage t_2, then in those stages it is in different states. The second

property says that if P_i is in different states in stages $t_1 + 1$ and $t_2 + 1$, and if the action pair played in stage t_1 equals the action pair that is played in stage t_2, then P_i must have been in different states already in stages t_1 and t_2. The third property is a generalization of the second property: if P_i is in different states in stages $t_1 + m$ and $t_2 + m$, and if the action pair played in stage $t_1 + l$ equals the action pair that is played in stage $t_2 + l$ for $l \in \{0, 1, \ldots, m-1\}$, then P_i must have been in different states already in stages t_1 and t_2. The fourth property says that the complexity of a finite sequence of action pair w.r.t. Player 1 is independent of the action that Player 2 plays in the last stage.

Lemma 18 *Let P_i be a pure automaton of player i that is compatible with ω.*

1. *If $\omega_i(t_1) \neq \omega_i(t_2)$ then $q_i(t_1) \neq q_i(t_2)$.*
2. *If $q_i(t_1 + 1) \neq q_i(t_2 + 1)$ and $\omega(t_1) = \omega(t_2)$, then $q_i(t_1) \neq q_i(t_2)$.*
3. *More generally, if $q_i(t_1 + m) \neq q_i(t_2 + m)$ and $\omega(t_1 + l) = \omega(t_2 + l)$ for every $l \in \{0, 1, \ldots, m-1\}$, then $q_i(t_1) \neq q_i(t_2)$.*
4. *If $\omega = (\omega(t))_{t=1}^{T}$ and $\omega' = (\omega'(t))_{t=1}^{T}$ are two finite sequences that differ only in the action of Player 2 at stage T, that is, $\omega_i(t) = \omega'_i(t)$ for every $t \in \{1, 2, \ldots, T\}$ and every $i \in \{1, 2\}$, except of for $t = T$ and $i = 2$, then the complexity of ω w.r.t. Player 1 is equal to the complexity of ω' w.r.t. Player 1.*

Proof. The first claim holds since the automaton's output is a function of the automaton's state. The second claim follows since the new state of the automaton is a function of the current state and of both players' actions. The third claim follows from the second claim by induction. The fourth claim follows since for a finite sequence, the action of Player 2 in the last stage T does not affect the evolution of the automaton of Player 1 in the first T stages. ■

A (finite or infinite) sequence of action pairs $\omega = (\omega(t))_t$ is *coordinated* if $\omega_1(t) = \omega_1(t')$ if and only if $\omega_2(t) = \omega_2(t')$, for every $t \neq t'$. The following result follows from Neyman [12].

Lemma 19 *Let $\omega = (\omega(t))_{t=1}^{T}$ be a coordinated sequence of action pairs and let $T_0 \leq T$. If $(\omega(t))_{t=t_2}^{T}$ is not a prefix of $(\omega(t))_{t=t_1}^{T}$ for every $t_1 < t_2 \leq T_0$, then $\mathrm{comp}_i(\omega) \geq T_0$ for each player i.*

Proof. Assume to the contrary that the condition of the lemma holds but there is a pure automaton for player i with size less than T_0 that is compatible with ω. By the pigeon hole principle, there are $t_1 < t_2 \leq T_0$ such that $q_i(t_1) = q_i(t_2)$. By

Lemma 18(1) $\omega_i(t_1) = \omega_i(t_2)$, and since ω is coordinated we have $\omega_{3-i}(t_1) = \omega_{3-i}(t_2)$. It follows by Lemma 18(2) that $q_i(t_1 + 1) = q_i(t_2 + 1)$. Continuing inductively we deduce that $q_i(t_1 + l) = q_i(t_2 + l)$ for every l for which $t_2 + l \leq T$. This implies that $(\omega(t))_{t=t_2}^{T}$ is a prefix of $(\omega(t))_{t=t_1}^{T}$, a contradiction. ■

Corollary 20 $\mathrm{comp}_1(\omega^*) \geq k^3 + 2k^2 + k + 1$.

Proof. The definition of the complexity of a sequence implies that the complexity of a sequence cannot be lower than the complexity of any of its subsequences. Consider then the prefix ω' of length $k^3 + 2k^2 + k + 4$ of ω^*, which involves only a coordinated play. For this sequence the condition in Lemma 19 is satisfied for ω^* with $T_0 = k^3 + 2k^2 + k + 1$, and therefore $\mathrm{comp}_1(\omega') \geq k^3 + 2k^2 + k + 1$, as desired. ■

Lemma 21 $\mathrm{comp}_2(\omega^*) \geq k^3 + 2k^2 + k + 4$.

Proof. Consider the prefix ω' of ω^* of length $k^3 + 2k^2 + k + 4$. Let ω'' be the sequence ω' after adding the action pair (D, D) at the end, and let ω''' be the sequence ω' after adding the action pair (C, D) at the end. Note that ω''' is a prefix of ω^*, hence $\mathrm{comp}_2(\omega^*) \geq \mathrm{comp}_2(\omega''')$. By Lemma 18(4), $\mathrm{comp}_2(\omega''') = \mathrm{comp}_2(\omega'')$. Apply Lemma 19 to the sequence ω'' with $T_0 = k^3 + 2k^2 + k + 4$ to deduce that $\mathrm{comp}_2(\omega'') \geq k^3 + 2k^2 + k + 4$. The result follows. ■

4 Comments and Discussion

4.1 On the definition of BCC equilibria

The definition of the concept of BCC equilibrium is analogous to the definition of the concept of Nash equilibrium; in both we ask whether a specific behavior (that is, a pair of strategies) is stable. Thus, in a c-BCC equilibrium we assume that each player already has an automaton with which she is going to play the game, and we ask whether playing this automaton is the best response given the automaton that the other player is going to use. As in the definition of Nash equilibrium, we do not ask how the players arrive at these automata, and we do not restrict the sizes of these automata (though the memory cost does bound the maximum size of automata that the players will use). In principle, it may well be that some BCC equilibrium payoff can be supported only with prohibitively large automata, which we would like to rule out. That is, we may want to add the size of the automata that the players use to the definition itself. In our construction (see the proof of Theorem 5), to support a c-BCC equilibrium payoff that is close to some target payoff x we use two automata of similar sizes; the size of each automaton is related to both c and to the level of

approximation to the target payoff: as c gets closer to 0, and as the c-BCC equilibrium payoff gets closer to x, we use larger automata.

Even though the definition of a BCC equilibrium is theoretically appealing, to prove the folk theorem we use outrageously large automata. For example, the automata that we construct to approximate a target payoff vector by 0.01 is about $(100)^3$.

4.2 BCC equilibria and Nash equilibria

As mentioned before, Theorem 5 does not rule out the possibility that there is a payoff vector that is *not* strictly individually rational and yet is a BCC equilibrium; that is, a BCC equilibrium payoff need not be a Nash equilibrium payoff. The theorem also does not rule out the possibility that some payoff vector that is individually rational w.r.t. the min-max value in *mixed* strategies, but not individually rational w.r.t. the min-max value in pure strategies, would not be a BCC equilibrium payoff, so that a Nash equilibrium payoff need not be a BCC equilibrium payoff.

Moreover, in zero-sum games it is not clear whether there is a unique BCC equilibrium payoff. If in zero-sum games there is always a unique BCC equilibrium payoff, then this quantity can be called the *BCC value* of the game. However, it is possible that in zero-sum games there will be more than one BCC equilibrium payoff, in which case even in this class of games, the outcome will crucially depend on the relative computational power of the players.

4.3 The discounted game

When two pure automata play against each other, the play enters a cycle, and therefore the sequence of stage payoffs is eventually periodic. For such sequences, the limit of the discounted sum is equal to the long-run average payoff. The definition of a c-BCC equilibrium is changed to take into account the discount factor.

Given $c > 0$ and a discount factor $\lambda \in (0,1)$, a pair of mixed automata (M_1, M_2) is a (c,λ)-*BCC equilibrium payoff* if it is a Nash equilibrium for the utility functions $U_i^{c,\lambda}(M_1,M_2) = \gamma_i^{\lambda}(M_1,M_2) - c|M_i|$ for $i = 1,2$, where $\gamma_i^{\lambda}(M_1,M_2)$ is the λ-discounted payoff of player i when the players use the automata (M_1,M_2). A vector $x \in \mathbb{R}^2$ is a BCC equilibrium payoff if it is the limit, as c goes to 0 and λ goes to 1, of payoffs that correspond to (c,λ)-BCC equilibria. That is, there is a sequence $(c_n)_{n\in\mathbb{N}}$ and $(\lambda_n)_{n\in\mathbb{N}}$ that converge to 0 and 1, respectively, and for each n there is a (c_n,λ_n)-BCC equilibrium $(M_1^{c_n,\lambda_n}, M_2^{c_n,\lambda_n})$, such that $\lim_{n\to\infty} \gamma^{\lambda}(M_1^{c_n,\lambda_n}, M_2^{c_n,\lambda_n}) = x$ and $\lim_{n\to\infty} c_n|M_i^{c_n,\lambda_n}| = 0$ for $i = 1,2$.

Our folk theorem holds for this concept, with the same construction.

4.4 A more general definition of a BCC equilibrium

The definition of the concept of c-BCC equilibrium assumes that the utility of each player is additive, and that the memory cost is linear in the memory size. There are applications where the utility function U_i has a different form.

- Players may disregard the memory cost, but be bounded by the size of memory that they use.

$$U_i(M_1, M_2) = \begin{cases} \gamma_i(M_1, M_2) & |M_i| \leq k_i, \\ -\infty & |M_i| > k_i. \end{cases}$$

  This situation occurs, e.g., when players are willing to invest huge amounts of money even if the profit is low, but the available technology does not allow them to increase their memory size beyond some limit. Such a situation may occur, e.g., in the area of code breaking, where countries invest large sums of money to be able to increase the number of other countries' codes that they break, yet they are bounded by technological advances.

- Memory is costly, yet players do not save money by reducing their memory size. That is, a pair of mixed automata (M_1, M_2) is a c-BCC equilibrium if for each $i \in \{1, 2\}$ and for every pure automaton $P_i \in M_i$ one has $\gamma_i(M_i, M_{3-i}) \geq \gamma_i(P_i, M_{3-i})$, and, if $P_i > M_i$, one has $\gamma_i(M_i, M_{3-i}) \geq \gamma_i(P_i, M_{3-i}) - c(|P_i| - |M_i|)$. This situation occurs, e.g., when the players are organizations whose size cannot be reduced.

It may be of interest to study the set of equilibrium payoffs for various utility functions U_i, and to see whether and how this set depends on the shape of this function.

4.5 More than two players

The concept of BCC equilibrium payoff is valid for games with any number of players. However, Theorem 5 holds only for two-player games. One crucial point in our construction is that if a deviation is detected, a player is punished for a long (yet finite) period of time by a punishing action. When there are more than two players, the punishing action of, say, Player 1 against Player 2 may be different that the punishing action of Player 1 against player 3. It is not clear how to construct an automaton that can punish each of the other players, if necessary, and such that all these memory cells will be used on the equilibrium path.

4.6 BCC equilibria in one-shot games

The concept of BCC equilibrium that we presented here applies to repeated games. The concept can be naturally adapted to one-shot games as well.[10] For example, consider the following game, that appears in Halpern and Pass [7]. Player 1 chooses an integer n and tells it to Player 2; Player 2 has to decide whether n is a prime number or not, winning 1 if she is correct, losing 1 if she is incorrect. Plainly the value of this game for Player 2 is 1: Player 2 can check whether the choice of Player 1 is a prime number. However, as there is no efficient algorithm to check whether an integer is a prime number, it is not clear whether in practice risk-neutral people would be willing to participate in this game as Player 2.

The concept of BCC equilibrium can be adapted to such situations, and one can study the set of BCC equilibrium payoffs, and how this set depends on the relative memory cost of the two players.

In the context of the Computer Science literature one could conceive of an analog solution concept, where automata are replaced by Turing machines, and the memory size is replaced by the length of the machine's tape.

References

[1] D. Abreu and A. Rubinstein. (1988) The structure of Nash equilibrium in repeated games with finite automata, Econometrica 56, 1259-1281.

[2] J. Banks, and R. Sundaram (1990) Repeated games, finite automata and complexity, Games and Economic Behaviour, 2, 97-117.

[3] E. Ben Porath, (1993), Repeated games with finite automata, Journal of Economic Theory, 59, 17-32.

[4] K. Chatterjee, and H. Sabourian (2000), Multiperson bargaining and strategic complexity, Econometrica, 68, 1491-1509.

[5] K. Chatterjee, and H. Sabourian (2008), Game Theory and Strategic Complexity, in Encyclopedia of Complexity and System Science, Editor-in-Chief Robert A. Meyers, Springer.

[6] D. Gale, and H. Sabourian, (2005) Complexity and competition, Econometrica, 73, 739-770.

[10] We thank Ehud Kalai for drawing our attention to this issue.

[7] Halpern J.Y. and Pass R. (2008) Game Theory with Costly Computation, preprint.

[8] E. Kalai (1990) Bounded Rationality and Strategic Complexity in Repeated Games, in Game Theory and Applications, eds. Ichiishi, Neyman and Tauman, San Diego: Academic Press, 1990, 131-157.

[9] E. Maenner (2008) Adaptation and complexity in repeated games, Games and Economic Behavior, 63, 166-187.

[10] A. Neyman, (1985) Bounded complexity justifies cooperation in the finitely-repeated Prisoners' Dilemma, Economics Letters, 19, 227-229.

[11] A. Neyman, (1997) Cooperation, repetition and automata, in Cooperation: Game-Theoretic Approaches, NATO ASI Series F, Vol. 155, S. Hart and A. Mas-Colell (eds.), Springer-Verlag. 233 255.

[12] A. Neyman (1998) Finitely repeated games with finite automata, Mathematics of Operations Research, 23, 513-552

[13] A. Neyman, and D. Okada (1999) Strategic entropy and complexity in repeated games. Games and Economic Behavior, 29, 191-223.

[14] A. Neyman, and D. Okada (2000) Repeated games with bounded entropy. Games and Economic Behavior, 30, 228-247.

[15] A. Neyman, and D. Okada, (2000) Two-person repeated games with finite automata. International Journal of Game Theory, 29, 309-325.

[16] M. Piccione (1992) Finite automata equilibria with discounting, Journal of Economic Theory, 56, 180-193.

[17] M. Piccione, and A. Rubinstein (1993) Finite automata play a repeated extensive game, Journal of Economic Theory, 61, 160-168.

[18] A. Rubinstein (1986) Finite automata play the repeated prisoner's dilemma, Journal of Economic Theory, 39, 83-96.

[19] A. Rubinstein, (1998) Modeling bounded rationality, MIT Press, Cambridge, Mass.

[20] H. Sabourian, (2003) Bargaining and markets: complexity and the competitive outcome, Journal of Economic Theory, 116 , 189-228.

[21] H.A.Simon, (1972) Theories of bounded rationality, in "Decision and Organization" (C.B. McGuire and R. Radner, Eds), North- Holland, Amsterdam.

[22] H.A. Simon,(1978) On how to decide what to do, Bell Journal of Economics, 9, 494-507.

[23] E. Zemel (1989) Small talk and cooperation: A note on bounded rationality, Journal of Economic Theory, 49, 1-9.

A Proof of Theorem 5

This section is devoted to the proof of Theorem 5. To keep consistency with Section 3 we denote the min-max action of each player by D. Assume w.l.o.g. that payoffs are bounded by 1, and let $x \in F \cap V$ satisfy $x_i > v_i$ for each $i = 1, 2$. Fix $\eta < \min\{x_1 - v_1, x_2 - v_2\}$. To rule out trivial cases, assume that each player has at least two actions.

By Carathéodory's Theorem, the vector x is a convex combination of three entries in the payoff matrix, say $\{(a_1^1, a_2^1), (a_1^2, a_2^2), (a_1^3, a_2^3)\}$. Moreover, one of the three vectors can be arbitrarily chosen. We will assume that $a_1^1 = D$ and $a_2^1 = D$ are the punishment actions of the two players. Write

$$x = \sum_{i=1}^{3} \alpha_i u(a_1^i, a_2^i),$$

where $(\alpha_i)_{i\in\{1,2,3\}}$ are non-negative numbers summing to 1. Fix $\varepsilon > 0$, a natural number $k_0 > \frac{1}{3\varepsilon}$, and a natural number $k > (k_0 + 1)^3$. Let k_1, k_2, k_3 be three positive integers such that (a) $\sum_{i=1}^{3} k_i = k_0$, and (b) $|k_i - \alpha_i k_0| \leq 1$ for $i \in \{1, 2, 3\}$.

The action pairs (a_1^2, a_2^2) and (a_1^3, a_2^3) can have various configurations. The Babbling Phase differs among these configurations, and therefore the equilibrium play differs as well. However, no new ideas are necessary for the construction. Below we will list all possible configurations, write down the equilibrium path that corresponds to each such configuration, and describe the pure and mixed automata that the players use. We could have provided one construction that takes care of all configurations simultaneously, but the proof would be much more complicated and would requires new ideas, so we prefer to handle each configuration separately.

Denoting the (two or three) actions of each player that take part in the convex combination by D (which is the punishment action) and C (and if necessary, B), the possible configurations of the three entries in the matrix that take part in the convex combination of x are (up to symmetries):

C1) Coordinated play: (D, D), (C, C), and (B, B).

C2) (D,D), (C,C), and (C,D).

C3) (D,D), (D,C), and (C,D).

C4) (D,D), (D,C), and (D,B).

C5) (D,D), (D,C), and (C,B).

C6) (D,D), (C,C), and (C,B).

A.1 The Three Phases of the Equilibrium Play

To save repetitions, we provide here some definitions that will be used in all configurations. As in Section 3, the equilibrium play will be divided into three phases. The Punishment Phase will be

$$P^* = k^3 \times (D,D). \tag{11}$$

In Configuration (C1) the play is coordinated, and there is no need to use a Babbling Phase. In Configurations (C2) and (C3) the Babbling Phase will be similar to that in Section 3:

$$B^* := \sum_{n=1}^{k} \big(k \times (C,C) + k \times (D,D)\big) + (k+1) \times (C,C). \tag{12}$$

In Configurations (C4)–(C6) we will append to the Babbling Phase an additional block, to incorporate the third action of Player 2 on the equilibrium path. As in Section 3, the base of the Regular Phase will be

$$R^* = \sum_{i=1}^{3} k_i \times (a_1^i, a_2^i).$$

Since payoffs are bounded by 1, the average payoff along R^* is within ε of x.

The set of states of the automaton will always be denoted by $Q = \{1, 2, \ldots, |Q|\}$. The sets Q^P, Q^C, and Q^D of the states that implement the Punishment Phase, the C-blocks, and the D-blocks, respectively, are defined as in Step 1 in Section 3.3.

The initial state will always be

$$q^* = 1.$$

As in Section 3, the constructions that we provide below define only some of the transitions. All transitions that are not define lead to state 1, thereby initiate a long Punishment Phase.

A.2 Configuration C1: (D,D), (C,C), and (B,B).

Here the equilibrium path is coordinated, and one can use the construction of Abreu and Rubinstein [1], which does not use a Babbling Phase. That is, consider the following play path:

$$\begin{aligned}\omega^* &= P^* + \sum_{n=1}^{\infty} R^* \\ &= k^3 \times (D,D) + \sum_{n=1}^{\infty} \left(k_1 \times (D,D) + k_2 \times (C,C) + k_3 \times (B,B)\right).\end{aligned}$$

The play path ω^* is coordinated, and by Lemma 19 its complexity w.r.t. each player is at least $k^3 + k_0$. Define the following pure automaton P_1 with $k^3 + k_0$ states, which implements the play path ω^* in a naive way.

$$\begin{aligned} f(q) &= D, \quad q \in \{1,2,\ldots,k^3+k_1\}, \\ f(q) &= C, \quad q \in \{k^3+k_1+1, k^3+k_1+2, \cdots, k^3+k_1+k_2\}, \\ f(q) &= B, \quad q \in \{k^3+k_1+k_2+1, k^3+k_1+k_2+2, \cdots, k^3+k_1+k_2+k_3\}, \\ g(q,f(q)) &= q+1, \quad q \in \{1,2,\ldots,k^3+k_0-1\}, \\ g(k^3+k_0, f(k^3+k_0)) &= k^3+1. \end{aligned}$$

Let $P_2 = P_1$; that is, the automaton P_2 is identical to the automaton P_1.

The automata P_1 and P_2 are compatible with ω^* for Players 1 and 2 respectively, and the pair (P_1, P_2) forms a c-BCC equilibrium for every $c > 0$ sufficiently small. This implies that the automata (P_1, P_2) also form a Nash equilibrium, that is, a c-BCC equilibrium for $c = 0$.

A.3 Configuration C2: (D,D), (C,C), and (C,D).

This configuration generalizes the one we handled in Section 3. The Punishment Phase and Babbling Phase are given in Eqs. (11) and (12), and the base of the Regular Phase is

$$R^* := k_1 \times (D,D) + k_2 \times (C,C) + k_3 \times (C,D).$$

By Lemma 19 the complexity of ω^* w.r.t. Player 1 satisfies $\mathrm{comp}_1(\omega^*) \geq k^3+2k^2+k+1$, and the complexity of ω^* w.r.t. Player 2 satisfies $\mathrm{comp}_2(\omega^*) \geq k^3+2k^2+k+1+k_1$.

We now explain the construction of a pure automata for Player 1 that is compatible with ω^*. Let $j_1 \in \{1,2,\ldots,k-1\}$ and let $H_1 = \{h_1, \cdots, h_{k_3}\}$ be a set of distinct elements from the set $\{1,2,\ldots,k\}$ satisfying $h_1 = k_2 + 1$. The index j_1 determines

the C-block that will be reused to implement the part $k_3 \times (C, D)$ of the Regular Phase, and the set H_1 will indicate the states in this block that are reused.

Define the following pure automaton $P_1^{j_1,H_1}$ with $k^3 + 2k^2 + k + 1$ states.

- The output function is given by

$$\begin{aligned} f(q) &= D, \quad q \in Q^P \cup Q^D, \\ f(q) &= C, \quad q \in Q^C. \end{aligned}$$

- The Punishment and Babbling Phases are implemented naively:

$$g(q, f(q)) = q + 1, \quad q \in \{1, 2, \ldots, k^3 + 2k^2 + k\}.$$

- To implement the part $k_1 \times (D, D)$ of the Regular Phase we reuse states from the j_1'th D-block:

$$g(k^3 + 2k^2 + k + 1, C) = k^3 + 2j_1k - k_1 + 1.$$

- To implement the part $k_2 \times (C, C) + k_3 \times (C, D)$ from the Regular Phase we reuse states from the $(j_1 + 1)$'th C-block:

$$g(k^3 + 2j_1k + h_l, D) = k^3 + 2j_1k + h_{l+1}, \quad l \in \{1, 2, \ldots, k_3 - 1\}.$$

- After implementing R^* the automaton moves back to the reused states that implements the first pair of $k_1 \times (D, D)$:

$$g(k^3 + 2j_1k + h_{k_3}, D) = k^3 + 2j_1k - k_1 + 1.$$

One can verify that this automaton is compatible with ω^* for Player 1.

We now describe how to define the mixed automaton M_1 for Player 1. The number L of pure automata in the support of M_1 is denoted by L and given by $L = \lceil k^{1/3} \rceil$. Let $(j_1^d)_{d=1}^L$ be a set of distinct integers from $\{1, 2, \ldots, k - 1\}$, and let $(H_1^d)_{d=1}^L$ be a collection of sets, where $H_1^d = \{h_1^d, h_2^d, \cdots, h_{k_3}^d\}$, which satisfies the following conditions:

(E1) For each $d \in \{1, 2, \ldots, L\}$ we have $H_1^d \subset \{1, 2, \ldots, k\}$ with $h_1^d = k_2 + 1$.

(E2) For each two distinct indices $d, d' \in \{1, 2, \ldots, L\}$ we have $(H_1^d \setminus \{h_1^d\}) \cap (H_1^{d'} \setminus \{h_1^{d'}\}) = \emptyset$.

(E3) For each two distinct indices $d, d' \in \{1, 2, \ldots, L\}$ and each $1 \leq l < l' \leq k_3$ we have $h_{l'}^d - h_l^d \neq h_{l'}^{d'} - h_l^{d'}$.

We now exhibit one way to define $(j_1^d)_{d=1}^L$ and $(H_1^d)_{d=1}^L$. Let $p_1, p_2, \cdots, p_L$ be the first L prime numbers that are larger than L. By the Prime Number Theorem, and since $L = \lceil k^{1/3} \rceil$, we have $p_L < \frac{k}{L}$, provided k is sufficiently large. For every $d \in \{1, 2, \ldots, L\}$ set $j_1^d = d$ and $h_l^d = k_2 + 1 + (l-1)p_d$ for each $l \in \{1, 2, \ldots, k_3\}$. Since $k > (k_0 + 1)^3$ it follows that $L > k_3$, and therefore $(j_1^d)_{d=1}^L$ and $(H_1^d)_{d=1}^L$ satisfy (E1)–(E3). Note that the largest element of $(H_1^d)_{d=1}^L$ is $h_{k_3}^L$, which is $k_2 + 1 + (k_3 - 1)p_L < k$. We let M_1 be the mixed automaton for Player 1 that chooses one of the automata $(P_1^{j_1^d, H_1^d})_{d=1}^L$ according to the uniform distribution.

We now define a pure automaton for player 2 with $k^3 + 2k^2 + k + 1 + k_1$ states that is compatible with ω^*; the last k_1 states of the automaton will implement the part $k_1 \times (D, D)$. Let $j_2 \in \{1, 2, \ldots, k-1\}$ and let $H_2 = \{h_1, \cdots, h_{k_3}\}$ be a set of distinct elements from the set $\{1, 2, \ldots, k\}$ satisfying $h_1 = 1$. The index j_2 determines the D-block that will be reused to implement the part $k_3 \times (C, D)$ of the Regular Phase, and the set H_2 will indicate the states in this block that are reused.

Define the following pure automaton $P_2^{j_2, H_2}$.

- The first $k^3 + 2k^2 + k + 1$ implement naively the Punishment and Babbling Phases, and the last k_1 states implement the part $k_1 \times (D, D)$ of the Regular Phase:

$$\begin{aligned} f(q) &= D, \quad q \in Q^P \cup Q^D, \\ f(q) &= C, \quad q \in Q^C, \\ f(q) &= D, \quad k^3 + 2k^2 + k + 2 \le q \le k^2 + 2k^2 + k + 1 + k_1, \\ g(q, f(q)) &= q + 1, \quad k \in \{1, 2, \ldots, k^3 + 2k^2 + k + k_1\}. \end{aligned}$$

- To implement the part $k_2 \times (C, C)$ the automaton moves to the j_2'th C-block.

$$g(k^3 + 2k^2 + k + 1 + k_1, D) = k^3 + 2(j_2 - 1)k + k - k_2 + 1.$$

- The implementation of the part $k_3 \times (C, D)$ is done in the j_2'th D-block,

$$g(k^3 + 2(j_2 - 1)k + k + h_l, C) = k^3 + 2(j_2 - 1)k + k + h_{l+1}, \quad l \in \{1, 2, \ldots, k_3 - 1\}.$$

- After completing one period of the Regular Phase the automaton moves to the state that implemented the beginning of the Regular Phase:

$$g(k^3 + 2(j_2 - 1)k + k + h_{k_3}, C) = k^3 + 2k^2 + k + 2.$$

One can verify that this automaton is compatible with ω^* for Player 2. We define a mixed automaton M_2 for Player 2 analogously to the definition of M_1.

Lemma 16 holds, and with the same proof, provided that $\frac{\eta}{2} > \frac{8}{k} + \frac{4}{L}$ and $c < \frac{\eta}{4k^3}$. Lemma 17 holds, and with the same proof, provided that $c > \frac{3}{Lk^3}$. In particular, it follows that x^* is indeed a BCC-equilibrium payoff.

A.4 Configuration C3: (D,D), (D,C), and (C,D).

In this case the Punishment Phase and Babbling Phase are given in Eqs. (11) and (12), and the base of the Regular Phase is

$$R^* = k_1 \times (D,D) + k_2 \times (D,C) + k_3 \times (C,D).$$

By Lemma 19 the complexity of ω^* w.r.t. Player 1 satisfies $\mathrm{comp}_1(\omega^*) \geq k^3 + 2k^2 + 1$ and its complexity w.r.t. Player 2 satisfies $\mathrm{comp}_2(\omega^*) \geq k^3 + 2k^2 + k + 1$.

We now construct a pure automaton for Player 1 with $k^3 + 2k^2 + 1$ states that is compatible with ω^*. The automaton depends on an integer $j_1 \in \{1, 2, \ldots, k-1\}$, which determines the blocks of the reused states, and two sets of distinct indices, $H_1 = \{h_1, \cdots, h_{k_2}\}$ and $\widehat{H}_1 = \{\widehat{h}_1, \cdots, \widehat{h}_{k_3}\}$, all in the range $\{1, 2, \ldots, k\}$. The set H_1 indicates the states in the j_1'th D-block that are reused to implement the part $k_2 \times (D,C)$; we require that $h_1 = k_1 + 1$. The set $\widehat{H}_1$ indicates the states in the (j_1+1)'th C-block that are reused to implement the part $k_3 \times (C,D)$.

The pure automaton $P_1^{j_1, H_1, \widehat{H}_1}$ for Player 1 contains $k^3 + 2k^2 + 1$ states and is constructed as follows.

- The $k^3 + 2k^2 + 1$ states of the automaton implement naively the Punishment Phase and the Babbling Phase, except of its last k stages.

- The automaton then moves to the beginning of the j_1'th C-block, which is reused to implement the last k stages of the Babbling Phase:

$$g(k^3 + 2k^2 + 1, C) = k^3 + 2(j_1 - 1)k + 1.$$

- The j_1'th D-block is reused to implement the part $k_1 \times (D,D) + k_2 \times (D,C)$ of the Regular Phase:

$$g(k^3 + 2(j_1-1)k + k + h_l, C) \;=\; k^3 + 2(j_1-1)k + k + h_{l+1}, \quad \forall l \in \{1, 2, \ldots, k_2 - 1\}.$$

- States $\widehat{H}_1$ of the (j_1+1)'th C-block are reused to implement the part $k_3 \times (C,D)$ of the Regular Phase:

$$\begin{aligned} g(k^3 + 2(j_1-1)k + k + h_{k_2}, C) &= k^3 + 2j_1 k + \widehat{h}_1, \\ g(k^3 + 2j_1 k + \widehat{h}_l, D) &= k^3 + 2j_1 k + \widehat{h}_{l+1}, \ \forall l \in \{1, \ldots, k_3 - 1\}. \end{aligned}$$

- After implementing $k_3 \times (C,D)$ the automaton moves to the beginning of the j_1'th D-block, where we implemented the part $k_1 \times (D,D)$:

$$g(k^3 + 2j_1 k + \widehat{h}_{k_3}, D) \;=\; k^3 + 2(j_1 - 1)k + k + 1.$$

To define the mixed automaton M_1 we let $(j_1^d, H_1^d, \widehat{H}_1^d)_{d=1}^L$ be a collection of triplets, where (a) $L = \lceil k^{1/3} \rceil$, (b) $(j_1^d)_{d=1}^L$ are distinct integers from $\{1, 2, \ldots, k-1\}$, and (c) each of the collections $(H_1^d)_{d=1}^L$ and $(\widehat{H}_1^d)_{d=1}^L$ satisfies (E2)–(E3). The mixed automaton M_1 chooses one of the pure automata $(P_1^{j_1^d, H_1^d, \widehat{H}_1^d})_{d=1}^L$ according to the uniform distribution.

We now turn to the definition of the pure automaton $P_2^{j_2, H_2, \widehat{H}_2}$ for Player 2 that is compatible with ω^*, where $j_2 \in \{1, 2, \ldots, k-1\}$, $H_2 = \{h_1, \cdots, h_{k_2}\} \subseteq \{1, 2, \ldots, k\}$, $h_1 = 1$, and $\widehat{H}_2 = \{\widehat{h}_1, \cdots, \widehat{h}_{k_3}\} \subseteq \{1, 2, \ldots, k\}$. The automaton $P_2^{j_2, H_2, \widehat{H}_2}$ has $k^3 + 2k^2 + k + 1$ states and is defined as follows.

- The $k^3 + 2k^2 + k + 1$ states of the automaton implement naively the Punishment Phase and the Babbling Phase.

- At the end of this implementation, the automaton moves to the j_2'th D-block, to implement the part $k_1 \times (D, D)$ of the Regular Phase:

$$g(k^3 + 2k^2 + k + 1, C) = k^3 + 2j_2k - k_1 + 1.$$

- The $(j_2 + 1)$'th C-block is reused to implement the part $k_2 \times (D, C)$ of the Regular Phase:

$$g(k^3 + 2j_2k + h_l, D) = k^3 + 2j_2k + h_{l+1} \quad \forall l \in \{1, 2, \ldots, k_2 - 1\}.$$

- The $(j_2 + 1)$'th D-block is reused to implement the part $k_3 \times (C, D)$ of the Regular Phase:

$$\begin{aligned} g(k^3 + 2j_2k + h_{k_2}, D) &= k^3 + 2j_2k + k + \widehat{h}_1, \\ g(k^3 + 2j_2k + k + \widehat{h}_l, C) &= k^3 + 2j_2k + k + \widehat{h}_{l+1}, \quad \forall l \in \{1, 2, \ldots, k_3 - 1\}. \end{aligned}$$

- After one period of the Regular Phase ends the automaton moves to the state in the j_2'th D-block where the part $k_1 \times (D, D)$ starts:

$$g(k^3 + 2j_2k + k + \widehat{h}_{k_3}, C) = k^3 + 2j_2k - k_1 + 1.$$

The definition of the mixed automaton M_2 is analogous to that of M_1 in the current configuration. The rest of the proof remains as for Configuration (C2).

A.5 Configuration C4: (D, D), (D, C), and (D, B).

This configuration is different from the previous three configurations in an important aspect: whereas so far on the equilibrium path both players used the same number of actions, now the number of actions each player plays is different. If Player 1's action set contains at least three actions, say D, C, and B, then as in Configurations (C2) and (C3) we could still implement a Babbling Phase in which the players repeatedly play three types of blocks, namely, D-blocks, C-blocks, and B-blocks. This cannot be done if $|\mathcal{A}_1| = 2$. Here we show how to adapt the construction in this case.

We augment the Babbling Phase so that it contains a part in which Player 2 plays the action B:

$$B^* := \sum_{n=1}^{k} \big(k \times (C, C) + k \times (D, D)\big) + (k+1) \times (C, C) + k_3 \times (D, B).$$

The length of the last part in the Babbling Phase is k_3, so its length suffices to implement the last part of the base of the Regular Phase. The base of the Regular Phase is

$$R^* := k_1 \times (D, D) + k_2 \times (D, C) + k_3 \times (D, B).$$

By Lemma 19, the complexity of ω^* w.r.t. Player 1 satisfies $\text{comp}_1(\omega^*) \geq k^3 + 2k^2 + 1$ and its complexity w.r.t. Player 2 satisfies $\text{comp}_2(\omega^*) \geq k^3 + 2k^2 + k + 1 + k_3$.

We now describe a pure automaton of Player 1 that is compatible with ω^*. Let $j_1 \in \{1, 2, \ldots, k-1\}$ and let $H_1 = \{h_1, \cdots, h_{k_2}\}$ and $\widehat{H}_1 = \{\widehat{h}_1, \cdots, \widehat{h}_{k_3}\}$ be two sets of integers in the range $\{1, 2, \ldots, k\}$ that satisfy $h_1 = k_1 + 1$ and $\widehat{h}_1 = 1$. The set H_1 indicates the states in the $(j_1 + 1)$'th D-block that are reused to implement the part $k_2 \times (D, C)$. The set $\widehat{H}_1$ indicates the states in the j_1'th D-block that are reused to implement the part $k_3 \times (D, B)$ of both the Babbling Phase and the Regular Phase. The pure automaton $P_1^{j_1, H_1, \widehat{H}_1}$ for Player 1 contains $k^3 + 2k^2 + 1$ states and is constructed as follows.

- The $k^3 + 2k^2 + 1$ states of the automaton implement naively the Punishment Phase and the Babbling Phase, except of its last $k + k_3$ stages.
- The automaton now moves to the beginning of the j_1'th C-block, which is reused to implement the last k stages of the Babbling Phase:

$$g(k^3 + 2k^2 + 1, C) = k^3 + 2(j_1 - 1)k + 1.$$

- The automaton now implements the part $k_3 \times (D, B)$ by reusing states in the j_1'th D-block, as indicated by the set $\widehat{H}_1$:

$$g(k^3 + 2(j_1 - 1)k + k + \widehat{h}_l, B) \;=\; k^3 + 2(j_1 - 1)k + k + \widehat{h}_l \quad \forall l \in \{1, 2, \ldots, k_3 - 1\}$$

- The first k_1 stages of the (j_1+1)'th D-block are reused to implement the part $k_1 \times (D, D)$ of the Regular Phase:

$$g(k^3 + 2(j_1 - 1)k + k + \widehat{h}_{k_3}, B) = k^3 + 2j_1k + k + 1.$$

- States H_1 of the (j_1+1)'th D-block are reused to implement the part $k_2 \times (D, C)$ of the Regular Phase:

$$g(k^3 + 2j_1k + k + h_l, C) \;=\; k^3 + 2j_1k + k + h_{l+1}, \quad \forall l \in \{1, \ldots, k_2 - 1\}.$$

- Finally the automaton moves to the state that implements the first pair of $k_3 \times (D, B)$:

$$g(k^3 + 2j_1k + h_{k_2}, C) \;=\; k^3 + 2(j_1 - 1)k + k + \widehat{h}_1.$$

We now describe the construction of a pure automaton of Player 2 that is compatible with ω^*. Let $j_2 \in \{1, 2, \ldots, k-1\}$ and let $H_2 = \{h_1, \cdots, h_{k_2}\} \subset \{1, 2, \ldots, k\}$ with $h_1 = 1$. The set H_2 indicates the states in the (j_2+1)'th C-block that are reused to implement the part $k_2 \times (D, C)$. The pure automaton $P_2^{j_2, H_2}$ for Player 2 contains $k^3 + 2k^2 + k + 1 + k_3$ states and is constructed as follows.

- The $k^3 + 2k^2 + k + 1 + k_3$ states of the automaton implement naively the Punishment Phase and the Babbling Phase.

- The last k_1 stages of the j_2'th D-block are reused to implement the part $k_1 \times (D, D)$ of the Regular Phase.

$$g(k^3 + 2k^2 + k + 1 + k_3, D) = k^3 + 2j_2k - k_1 + 1.$$

- States H_1 of the (j_2+1)'th C-block are reused to implement the part $k_2 \times (D, C)$ of the Regular Phase:

$$g(k^3 + 2j_2k + h_l, D) \;=\; k^3 + 2j_2k + h_{l+1}, \quad \forall l \in \{1, 2, \ldots, k_2 - 1\}.$$

- The last $k_3 \times (D, B)$ of the Regular Phase is reused to implement the part $k_3 \times (D, B)$ of the Regular Play

$$g(k^3 + 2j_2k + h_{k_2}, D) = k^3 + 2k^2 + k + 2.$$

The mixed automaton M_2 is defined as in Configuration (C2). The rest of the proof is similar to that for Configuration (C3).

A.6 Configuration C5: (D,D), (D,C), and (C,B).

The Babbling Phase changes to

$$B^* := \sum_{n=1}^{k} \big(k \times (C,C) + k \times (D,D)\big) + (k+1) \times (C,C) + k_3 \times (C,B).$$

and the base of the Regular Phase is

$$R^* := k_1 \times (D,D) + k_2 \times (D,C) + k_3 \times (C,B).$$

By Lemma 19 the complexity of ω^* w.r.t. Player 1 satisfies $\mathrm{comp}_1(\omega^*) \geq k^3 + 2k^2 + 2$ and the complexity of ω^* w.r.t. Player 2 satisfies $\mathrm{comp}_2(\omega^*) \geq k^3 + 2k^2 + k + 1 + k_3$.

We now explain the construction of a pure automaton of Player 1 that is compatible with ω^*. Let $j_1 \in \{1, 2, \ldots, k-1\}$ and let $H_1 = \{h_1, \cdots, h_{k_2}\}$ and $\widehat{H}_1 = \{\widehat{h}_2, \widehat{h}_3, \cdots, \widehat{h}_{k_3}\}$ be two sets of indices in the range $\{1, 2, \ldots, k\}$. The index j_1 will determine the blocks of the reused states. The set H_1 will indicate the states in the j_1'th D-block that are reused to implement the part $k_2 \times (D,C)$; we require that $h_1 = k_1 + 1$. The set $\widehat{H}_1$ will indicate the states in the (j_1+1)'th C-block that are reused to implement both the part $k_3 \times (C,B)$ of the Babbling and the Regular Phases.

The pure automaton $P_1^{j_1, H_1, \widehat{H}_1}$ for Player 1 contains $k^3 + 2k^2 + 2$ states and is constructed as follows.

- The $k^3 + 2k^2 + 2$ states of the automaton implement naively the Punishment Phase and the Babbling Phase, except of its last $k - 1 + k_3$ stages.

- The automaton now moves to the beginning of the j_1'th C-block, which is reused to implement the part $(k-1) \times (C,C) + 1 \times (C,B)$ of the Babbling Phase:

$$g(k^3 + 2k^2 + 2, C) = k^3 + 2(j_1 - 1)k + 1.$$

- States $\widehat{H}_1$ of the (j_1+1)'th C-block are reused to implement the part $(k_3 - 1) \times (C,B)$ of the Babbling Phase:

$$\begin{aligned} g(k^3 + 2(j_1-1)k + k - 1, B) &= k^3 + 2j_1 k + \widehat{h}_2, \\ g(k^3 + 2j_1 k + \widehat{h}_l, B) &= k^3 + 2(j_1-1)k + \widehat{h}_{l+1}, \ \forall l \in \{2, \ldots, k_3 - 1\}. \end{aligned}$$

- The automaton now moves to the beginning of the j_1'th D-block to implement the part $k_1 \times (D,D)$:

$$g(k^3 + 2j_1 k + \widehat{h}_{k_3}, B) = k^3 + 2(j_1 - 1)k + k + 1.$$

- States H_1 of the j_1'th D-block are reused to implement the part $k_2 \times (D, C)$ of the Regular Phase:

$$g(k^3 + 2(j_1 - 1)k + k + h_l, C) \;=\; k^3 + 2(j_1 - 1)k + k + h_{l+1}, \quad \forall l \in \{1, 2, \ldots, k_2 - 1\}.$$

- Finally the automaton moves to the last state in the j_1'th C-block to implement the part $k_3 \times (C, B)$:

$$g(k^3 + 2(j_1 - 1)k + k + h_{k_2}, C) = k^3 + 2(j_1 - 1)k + k - 1.$$

We now describe a pure automaton of Player 2 that is compatible with ω^*. Let $j_2 \in \{1, 2, \ldots, k - 1\}$ and let $H_2 = \{h_1, \cdots, h_{k_2}\}$ be a set of distinct elements from the set $\{1, 2, \ldots, k\}$ satisfying $h_1 = 1$. Define the following pure automaton $P_2^{j_2, H_2}$ with $k^3 + 2k^2 + k + 1 + k_3$ states as follows.

- The $k^3 + 2k^2 + k + 1 + k_3$ states of the automaton implement naively the Punishment Phase and the Babbling Phase.

- After the Babbling Phase ends, the automaton reuses states in the j_2'th D-block to implement the part $k_1 \times (D, D)$ of the Regular Phase:

$$g(k^3 + 2k^2 + k + 1 + k_3, C) = k^3 + 2j_2k - k_1 + 1.$$

- States in the $(j_2 + 1)$'th C-block are used to implement the part $k_2 \times (D, C)$ of the Regular Phase, and the last reused state in this C-block points to state $k^3 + 2k^2 + k + 2$, where the unique B-block starts.

$$\begin{aligned} g(k^3 + 2j_2k + h_l, D) &= k^3 + 2j_2k + h_{l+1}, \quad l \in \{1, 2, \ldots, k_3 - 1\}, \\ g(k^3 + 2j_2k + h_{k_3}, D) &= k^3 + 2k^2 + k + 2. \end{aligned}$$

The rest of the proof is similar to that for Configuration (C4).

A.7 Configuration C6: (D, D), (C, C), and (C, B).

The Babbling Phase is similar to that in Configuration (C5) and the base of the Regular Phase is

$$R^* := k_1 \times (D, D) + k_2 \times (C, C) + k_3 \times (C, B).$$

By Lemma 19 the complexity of ω^* w.r.t. Player 1 satisfies $\text{comp}_1(\omega^*) \geq k^3 + 2k^2 + k + 2$ and its complexity w.r.t. Player 2 satisfies $\text{comp}_2(\omega^*) \geq k^3 + 2k^2 + k + 1 + k_3 + k_1$.

We now describe a pure automaton for Player 1 that is compatible with ω^*. Let $j_1 \in \{1, 2, \ldots, k-1\}$ and let $H_1 = \{h_1, \cdots, h_{k_3}\}$ be a set of distinct elements from the set $\{1, 2, \ldots, k\}$ satisfying $h_1 = k_2 + 1$. The index j_1 determines the blocks that will be reused to implement the Regular Phase, and the set H_1 will indicate the states that will be reused to implement the part $k_3 \times (C, B)$. Define the following pure automaton $P_1^{j_1, H_1}$ with $k^3 + 2k^2 + k + 2$ states.

- The Punishment and Babbling Phases (except the last $k - 1 + k_3$ pairs) are implemented naively.
- To implement the next k action pairs, which are $(k-1) \times (C, C) + 1 \times (C, B)$, the automaton moves to the beginning of the j_1'th C-block:
$$g(k^3 + 2k^2 + k + 2, C) = k^3 + 2(j_1 - 1)k + 1.$$
- To implement the last part of the Babbling Phase $(k_3 - 1) \times (C, B)$ the automaton reuses states H_1 of the $(j_1 + 1)$'th C-block:
$$\begin{aligned} g(k^3 + 2(j_1 - 1)k + k, B) &= k^3 + 2j_1k + h_2, \\ g(k^3 + 2j_1k + h_l, B) &= k^3 + 2j_1k + h_{l+1}, \quad l \in \{2, 3, \ldots, k_3 - 1\}. \end{aligned}$$
- To implement the part $k_1 \times (D, D)$ of the Regular Phase we reuse states from the j_1'th D-block:
$$g(k^3 + 2j_1k + h_{k_3}, B) = k^3 + 2j_1k - k_1 + 1.$$
- The first k_2 states in the $(j_1 + 1)$'th C-block implement the part $k_2 \times (C, C)$. Since the $(j_1 + 1)$'th C-block also implemented the part $(k_3 - 1) \times (C, B)$, to be able to implement the last part of the Regular Play, $k_3 \times (C, B)$, we need to add to this block another state that implements the action pair (C, B). We use the $(k_2 + 1)$'th state for this purpose, where the automaton is at the end of the part $k_2 \times (C, C)$:
$$g(k^3 + 2j_1k + h_1, B) = k^3 + 2j_1k + h_2.$$

Note that the first pair the part $k_3 \times (C, B)$ in the Babbling Phase is implemented at the last state of the j_1'th C-block, while first pair the part $k_3 \times (C, B)$ in the Regular Phase is implemented at state h_1 of the $(j_1 + 1)$'th C-block. The rest of the $k_3 - 1$ action pairs are implemented in the same states.

In this configuration, Player 2 does not need to reuse states and therefore he uses a pure automaton, rather than a mixed automaton as he did in previous configurations. The automaton P_2 of Player 2 contains $k^3+2k^2+k+1+k_3+k_1$ states and is constructed as follows.

- The Punishment and Babbling Phases are implemented naively, as well as the part $k_1 \times (D, D)$ of the Regular Phase:

$$\begin{aligned}
f(q) &= D, \quad q \in Q^P \cup Q^D, \\
f(q) &= C, \quad q \in Q^C, \\
f(q) &= B, \quad k^3 + 2k^2 + k + 2 \leq q \leq k^2 + 2k^2 + k + 1 + k_3, \\
f(q) &= D, \quad k^3 + 2k^2 + k + 1 + k_3 + 1 \leq q \leq k^3 + 2k^2 + k + 1 + k_3 + k_1, \\
g(q, f(q)) &= q + 1, \quad q \in \{1, 2, \ldots, k^3 + 2k^2 + 1\}, \\
g(q, C) &= q + 1, \quad q \in \{k^3 + 2k^2 + 2, k^3 + 2k^2 + 3, \cdots, k^3 + 2k^2 + 1 + k_3\}, \\
g(q, D) &= q + 1, \quad q \in \{k^3 + 2k^2 + 1 + k_3 + 1, \cdots, k^3 + 2k^2 + 1 + k_3 + k_1 - 1\},
\end{aligned}$$

- From the last state the automaton moves to the last C-block, to complete the implementation of the Regular Phase:

$$g(k^3 + 2k^2 + k + 1 + k_3 + 1, D) = k^3 + 2k^2 + k + 1 - k_2 + 1.$$

The rest of the proof for Player 2 is similar to that for Configuration (C3). Because Player 2 uses a pure automaton, the fact that the complexity of ω^* for Player 1 is the size of his automaton ensures that he cannot gain by deviating to a smaller or larger automaton.