Playing odds and evens with finite automata

Vladislav Makarov

May 12, 2020

Abstract

This paper is concerned with asymptotic behaviour of a repeated game of “odds and evens”, with strategies of both players represented by finite automata. It is proved that, for every \( n \), there is an automaton with \( 2^n \cdot \text{poly}(n) \) states which defeats every \( n \)-state automaton, in the sense that it wins all rounds except for finitely many. Moreover, every such automaton has at least \( 2^n \cdot (1 - o(1)) \) states, meaning that the upper bound is tight up to polynomial factors. This is a significant improvement over a classic result of Ben-Porath in the special case of “odds and evens”. Moreover, I conjecture that the approach can be generalised to arbitrary zero-sum games.

1 Introduction

This paper is concerned with so-called repeated games on finite automata. Informally, two automata play some simple game with incomplete information repeatedly, with their moves depending on the current state, and the next state depending on the current state and the move the opponent has just made.

This topic has been actively studied since 1980s, because finite automata exactly represent the set of strategies with finite memory. Moreover, limiting the set of strategies available to both players to those with finite memory makes it possible to rationalize co-operative behaviour in iterated prisoner’s dilemma and similar games, see Neyman [5, 6].

Informally, the number of states of an automaton is a measure of strategy’s complexity. Other notions of complexity have been studied as well, including ones that depend on “non-triviality” of the transition function of an automaton [2]. Further developments in this area led to the notion of strategic entropy, which characterises the complexity of the strategy and allows some properties of automatic strategies to be generalised to some wider classes [7].

Ben-Porath [3], and also Neyman and Okada [7] prove some results on the price of the repeated game, with pure strategies of the first and the second player being automata with \( n \) and \( f(n) \) states, respectively. The question is, how many states would allow the second
player to achieve a mean payoff that is consistently better than mean payoff of a single iteration, as \( n \) tends to infinity? As it turns out [3, Theorem 2], any subexponential number of states gives barely any improvement over using only \( n \) states. In fact, the improvement vanishes as \( n \) tends to infinity. More formally,

**Theorem A** (Ben-Porath [3]). Let \( G \) be any zero-sum game with two players, and \( G_{n,f(n)} \) be its repeated version, with strategies of the first player and the second player limited to automata with \( n \) and \( f(n) \) states, respectively. If \( f(n) \geq n \) and \( f(n) = 2^{o(n)} \), then,

\[
\lim_{n \to +\infty} \text{price}(G_{n,f(n)}) = \text{price}(G) = \lim_{n \to +\infty} \text{price}(G_{n,n}).
\]

We will consider only the simplest non-trivial case, when the base game is “odds and evens”. Instead of studying the usual game-theoretic notions, we consider a more combinatorial property: how many states do we need to win all except finitely many of rounds against any automaton from some class with \( n \) states.

The most interesting case is when this class is the class of all automata with \( n \) states. Then, by the aforementioned result of Ben-Porath, \( 2^{\Omega(n)} \) states are necessary to consistently win significantly more than half of the time, let alone almost always. On the other hand, as shown in Section 2, using the method of Ben-Porath, it is sufficient to use \( n^{2n+22} \) states.

The main result of this paper, established in Section 5, is that there is an automaton with \( 2^n \cdot \text{poly}(n) \) states that wins almost always. Furthermore, a lower bound of \( 2^n \cdot (1 - o(1)) \) states on the size of such an automaton is also presented in the Section 5.

The latter improvement arises from a more careful consideration of the constructions. Informally, both our construction and the construction for the general case at first try to determine which automaton \( \mathcal{A} \) from \( \mathcal{A} \) they are playing against, and then proceed to play against \( \mathcal{A} \) optimally. The upper bound for the general case is proved by enumerating the whole \( \mathcal{A} \) and checking all the possibilities, but our construction uses the fact that only small part of information about \( \mathcal{A} \) is actually required to play optimally against it.

The construction will use probabilistic method of Erdős, but only in the simplest possible variation, therefore no knowledge of Lovasz local lemma and other advanced results and methods is required. If you are interested in more information on probabilistic method, check the similarly named book by Alon and Spencer [1].

## 2 Definitions

**Definition.** A zero-sum game is a triple \( G = (\Sigma_1, \Sigma_2, R) \), where \( \Sigma_1 \) and \( \Sigma_2 \) are finite sets of possible moves for the first and the second player, respectively and \( R: \Sigma_1 \times \Sigma_2 \to \mathbb{R} \) is a payoff function. One round consists of the first player choosing some move \( x \) from \( \Sigma_1 \) and the second player choosing some move \( y \) from \( \Sigma_2 \). In each round, players choose their moves independently. The first player gets \( R(x, y) \) points from such a round, the second gets \(-R(x, y)\).

Consider the “odds and evens” game \( G = (\Sigma, \Sigma, R) \), where \( \Sigma = \{0, 1\} \) and \( R((0, 1)) = R((1, 0)) = 1 \), \( R((0, 0)) = R((1, 1)) = -1 \). If \( R(x, y) = 1 \), this is a win for the first player
and a loss for the second, and, vice versa, $R(x, y) = -1$ is a loss for the first player and a win for the second. In simple words, if $x$ and $y$ are equal, the second player wins, otherwise the first player wins.

Technically speaking, both automata choose the next state depending on the current state and the opponent’s move ($0$ or $1$). But, because the output of the automaton is fully determined by the state it is currently in, we can label transitions not with numbers $0$ or $1$, but with symbols $a$ and $b$, that correspond to the first automaton or the second automaton winning this round, respectively.

**Definition.** An automaton for the game of “odds and evens” is a tuple $A = (Q, \delta, f, q_1)$, where $Q$ is a finite set of states, $\delta: Q \times \{a, b\} \to Q$ is a transition function, $f: Q \to \Sigma$ is an output function and $q_1 \in Q$ is an initial state.

**Definition.** For an automaton $A = (Q, \delta, f, q_1)$ and its state $q \in Q$, call the value of $f(q)$ the label of the state $q$.

**Definition.** For the first automaton, let us call (any) cycle in the directed graph of $b$-transitions (that is, the transitions corresponding to the loss of the first automaton) a losing cycle.

**Definition.** For the second automaton, let us call a cycle in the directed graph of $b$-transitions a $b$-cycle, or a winning cycle.

How does the repeated game between automata proceed? Let $A = (P, \gamma, f, p_1)$ be the automaton of the first player and $B = (Q, \delta, g, q_1)$ be the automaton of the second player. Before the first round, $A$ is in the state $p_1$ and $B$ is in the state $q_1$. Suppose that before an $r$-th round $A$ is in the state $p$ and $B$ is in the state $q$. Automaton $B$ beats $A$ in the $r$-th round if $R(g(q_r), f(p_r)) = +1$. In this case $A$ goes to the state $\gamma(p, b)$ and $B$ goes to the state $\delta(q, b)$. Otherwise, $B$ loses to $A$, and the automata go to the states $\gamma(p, a)$ and $\delta(q, a)$, respectively. Note that the definition is not symmetric with respect to swapping $A$ and $B$.

**Definition.** $B$ beats $A$ if and only if $B$ wins against $A$ in all rounds, except for a finite number of them.

Consider the following class of questions. Let $A$ be some class of automata for “odds and evens”, with all automata from $A$ having no more than $n$ states. What is the minimum number of states in an automaton that beats $A$, in a sense that it beats every automaton from $A$?

If $A$ is the set of all automata with $n$ states, there is a more general result of Ben-Porath

**Theorem B** (Ben-Porath [3]). Suppose that $H$ is a zero-sum game with $k$ possible moves for the first player and $h$ moves for the second player, and $f(n) \geq n^2 k^n n^{nh}$. Then, there exists an automaton $B$ for the second player with $f(n)$ states, such that $B$ achieves mean payoff $\maxmin(H)$ or better against any automaton of the first player with $n$ states.

3
Applying this result to the case when \( H \) is “odds and evens” immediately implies that \( n^{2n+2}2^n \) states are enough to beat \( A \). Theorem 5 is an improvement over this “naive” upper bound in our special case.

On a high level, Ben-Porath’s proof of Theorem B works in the following way: enumerate possible automata of the first player (in our case, there are \( n^{2n+1}2^n \) of them) and proceed to play against the first of them optimally. In case of a loss, proceed to play against the second of them optimally, and so on. Basically, there are at most \( n^{2n+1}2^n \) steps and on \( i \)-th step the second automaton either starts winning indefinitely, or refutes the “conjecture” that the first player is using the \( i \)-th automaton. Hence, the second automaton will eventually start winning after considering the automaton that the first player actually uses.

The improvement comes from the fact that knowing the whole automaton \( A \) is hardly required for playing optimally against it. For example, a-transitions of \( A \) do not matter if we “do not intend to lose”.

3 Unknown initial state

Let us recall the following “folklore” result:

**Theorem 1** (folklore). Let \( A = (Q, \delta, f, q_1) \) be an automaton with \( n \) states. Then, denote by \( \mathcal{A} \) the set of all automata with the same output and transition functions, but with all possible initial states \( (n \) possibilities). Then there exists an automaton \( B \) with at most \( n^2 \) states that beats \( A \).

*Folklore proof.* Let \( Q = \{q_1, q_2, \ldots, q_n\} \). We will construct \( B \) in the following way: take \( n \) independent copies of the automaton \( A \) and leave only b-transitions in them. Denote the state \( q_k \) in the \( i \)-th copy of \( A \) by \( (i, q_k) \). The initial state of \( B \) will be \( (1, q_1) \). Then \( B \) beats \( A \) with initial state \( q_1 \) without ever leaving first copy of \( A \): we are always in the same state as \( A \), hence always print the same symbol, hence always win.

What will happen if \( A \) starts in \( q_2 \)? Then \( B \) either always wins, or loses at some moment. When we construct the automaton \( B \), we know when this loss will happen. Suppose that when this loss happens, \( B \) is in the state \( (1, r) \) and \( A \) goes to a state \( p \) along a-transition exactly after that. Then, add an edge from state \( (1, r) \) of \( B \) to the state \( (2, p) \). Then, \( B \) will infinitely beat \( A \) with initial state \( q_2 \) in the second copy, because we specifically aligned the losing transition of \( B \) from the first copy of \( A \) to the second copy of \( A \).

Now, if \( A \) starts in \( q_3 \), we similarly know for sure when \( B \) will lose for the first time. Handle this case by adding a losing edge to the state when it will happen, as long as there was not such an edge (otherwise just move to the next copy). Repeat this process \( n \) times, on the \( i \)-th step we extend \( B \) in a way that it beats \( A \) with initial state being one of \( \{q_1, q_2, \ldots, q_i\} \).

In other words, \( n^2 \) states are enough to beat an automaton that we know almost completely except for the initial state. The bound is in no way tight, and it is possible
but tedious to optimize the number of states in $B$ non-asymptotically, to something like $\frac{n^2}{2} + O(n)$. But numeric evidence for small $n$ (from 1 to 4) suggests that optimal $B$ has at most $n$ states! It is quite surprising, because beating a single fixed automaton with $n$ states does indeed require $n$ states. Hence, if these empirical findings are actually true and are not just an example of law of small numbers, hiding the initial state does not really help.

**Conjecture 1.** The upper bound on the number of states in $B$ in the Theorem 1 can be replaced by $n$, or, at least, by some linear function of $n$.

## 4 Unknown winning transitions

Now, assume that we know the set of states and b-transitions of $A$, but do not know a-transitions, initial state and the output function. Formally speaking, for a given directed graph $G$ on $n$ vertices with all outdegrees 1, the set $\mathcal{A} = \mathcal{A}(G)$ consists of all automata with $n$ states and b-transitions forming the graph $G$. What can we prove in this case?

We will need a few simple statements about strings over the binary alphabet. All these results are in no way new, of course.

**Definition.** Two binary strings $s$ and $t$ of the same length $n$ are equivalent ($s \sim t$) if one of them is a cyclic shift of another. An equivalence class of this equivalence relation is called a cyclic string of length $n$.

Let $c_n$ be the number of cyclic strings of length $n$.

**Definition.** Let us call a cyclic string $s$ simple if there is no such string $t$ and integer $k > 1$, such that $s = t^k$. It is easy to see that this definition does not depend on the choice of the representative.

Denote the set of all simple cyclic strings of length $n$ by $P_n$.

**Lemma 1** (OEIS A001037). $|P_n| = \frac{2^n}{n} \cdot (1 - o(1))$.

**Definition.** For non-empty string $s$ over some alphabet, denote the infinite string $ssss\ldots$ by $s^\omega$.

**Lemma 2** (Fine and Wilf [4, Theorem 3.1]). If $p^\omega = q^\omega$ for a simple string $p$, then $q = p^k$ for some positive integer $k$.

**Theorem 2.** There exists a directed graph $G$ on $n$ vertices with all outdegrees equal to 1, such that any automaton $B$ that beats $\mathcal{A}(G)$ has at least $2^n \cdot (1 - o(1))$ states.

**Proof.** Let $G$ be simply a directed cycle with $n$ vertices. Then, every automaton from $\mathcal{A}(G)$ has exactly one losing cycle, and all its states belong to this cycle. Consider $\mathcal{A}_p$ — the set of automata from $\mathcal{A}(G)$, such that the labels of the states, when read along the losing cycle and then interpreted as a cyclic string, form a simple cyclic string $p \in P_n$. 

5
When $B$ infinitely beats an $A \in \mathcal{A}_p$, it also repeatedly goes through some b-cycle. Suppose that $B$ prints some string $q$ while going along the aforementioned cycle. Because $B$ beats $A$, there exists a string $p_0$, equivalent to $p$, such that the infinite strings $p_0^\omega$ and $q^\omega$ coincide. Therefore, by Lemma 2, $q = p_0^k$ for some positive integer $k$. Moreover, by Lemma 2, powers of non-equivalent simple strings cannot be equivalent. Because every two b-cycles of $B$ either do not intersect, or coincide, there should be at least $|P_n|$ non-intersecting b-cycles in $B$, with each having length at least $n$ (the length of the cycle is the $|q| = |p_0^k| = k|p_0| = kn \geq n$). Hence, $B$ has at least $n|P_n| = n \cdot (2^n \cdot (1-o(1))) = 2^n \cdot (1-o(1))$ states.

**Theorem 3.** For any $G$ there exists an automaton $B$ that beats $\mathcal{A}(G)$ and has at most $2^n \cdot r(n)$ states, where $r$ is some polynomial.

**Proof.** A stronger statement (Theorem 5) will be established in the next section.

## 5 Unknown automaton

Now, suppose that only the number of states is known about $A$. In other words, $\mathcal{A}$ is the set of all automata with $n$ states.

**Theorem 4.** Any $B$ that beats $\mathcal{A}$ has at least $2^n(1-o(1))$ states.

**Proof.** A consequence of Theorem 2.

**Definition.** A device is a deterministic unary automaton with all states reachable from the initial state and states labeled with symbols from $\Sigma$ instead of being classified as accepting or not. Because $\Sigma = \{0, 1\}$ in our case, the latter difference is purely semantical. Moreover, we consider devices only up to a permutations of vertices that preserve transitions, labels and the initial state.

In other words, for any state $q$ in an automaton $A \in \mathcal{A}$, there is a corresponding device (let us call it $\text{device}(A, q)$) with at most $n$ states, such that $\text{device}(A, q)$ beats $A$ if interpreted as an automaton of the second player, with transitions of $\text{device}(A, q)$ corresponding to b-transitions. Constructing such a device is simple: just go along b-transitions of $A$, starting from the state $q$, until the states start repeating themselves. The states we visited, along with their labels and outgoing b-transitions, will constitute exactly what we need. There are no a-transitions, because devices “do not intend to lose”.

Let $L_n$ be the set of all devices with $n$ states. The main reason for considering devices at all is that there are relatively few of them, as compared to the number of possible automata of the first player, which is $n^{2n+1}2^n$.

**Lemma 3.** $|L_n| \leq 2^{n+1}n$.

**Proof.** There are two types of devices on $n$ vertices:
1) Some non-empty aperiodic part and a cycle. Then, there is only one possible initial state, at most \( n \) possible lengths of the aperioidic parts and \( 2^n \) ways to choose the labels on the states.

2) A big cycle. In this case, there are \( n \) ways to choose the initial state and at most \( 2^n \) ways to choose the labels on the states.

In total, there are at most \( 2^{n+1}n \) devices on \( n \) vertices. \qed

**Theorem 5.** For some polynomial \( r \), there exists an automaton \( B \) with at most \( 2^n r(n) \) states, that beats \( A \).

**Proof.** In an ideal scenario, the proof of this Theorem would be similar to the proofs of the Theorems B and 1: enumerate all possible conjectures about how the chosen automaton of the first player \( A \in A \) “looks in general” and then proceed to play against \( A \) optimally, with improvement over the Theorem B coming from dealing with several possible \( A \) at the same time.

By Lemma 3 there are relatively few devices compared to the number \( |A| = n^{2n+1}2^n \) of possible automata on \( n \) vertices. Moreover, as mentioned above, in order to play against \( A \) optimally, it is sufficient to know \( \text{device}(A, q) \) and the knowledge of the entire \( A \) is not necessary. One could say that \( \text{device}(A, q) \) captures the “losing essence” of \( A \) with initial state \( q \). In some sense, devices “contain less information” than automata, because a lot of irrelevant information is omitted from device description.

However, not everything is so simple. Up until now, all upper bounds in this paper were proved by using a deterministic “guess-and-check” strategy. In other words, each losing move refutes our current conjecture, but we know, how exactly the game unfolded before if the next conjecture is true. Now, because devices don’t contain any information about \( a \)-transitions, we willingly forget the information that makes us know the current state of the game under the current conjecture. Therefore, reasoning about any fixed construction appears to be very difficult, but we only need to prove an existence of a necessary construction. The probabilistic method \([1]\) is one of go-to ways to deal with such a situation. Moreover, because of the randomness, we will need to repeat each device several times. These observations should lead to the following construction.

Consider all devices from \( L = \bigcup_{i=1}^n L_i \). Build an automaton \( B \), where each device from \( L \) appears \( nk \) times (we will choose \( k \) later), with transitions of each device corresponding to \( b \)-transitions of \( B \). Consider a random permutation of all these devices (with multiplicity) and unite them in that order in one big cycle with \( nk|L| \) device copies in total. Moreover, direct all \( a \)-transitions from each device to the initial vertex of the next device along the cycle. Denote the set of all automata that we could get with non-zero probability by \( B \).

Proving that there exists a \( B \in B \), such that it beats every \( A \in A \) is enough to establish the theorem. To prove the last claim, it is enough to prove that

\[
P(\text{random } B \in B \text{ beats all } A \in A) \geq 1 - \sum_{A \in A} P(\text{random } B \in B \text{ does not beat } A \in A) > 0
\]
Indeed, let us estimate the probability that a random $B \in \mathcal{B}$ does not beat some fixed $A \in \mathcal{A}$.

For $B$ to win, it is enough for $A$ to reach some state $q$ in the same round with $B$ reaching one of the $nk$ copies of device$(A,q)$. Denote the set \( \{ \text{device}(A,q_i) \mid 1 \leq i \leq n \} \) by $L_A$. Because we consider devices only up to some kind of isomorphism, this set may contain less than $n$ elements.

Consider $(s + 1)$-st time when $B$ just entered into a copy of some device from $L_A$. Then, this is a random not yet visited copy of devices from $L_A$, because we connected device copies into a cycle in a random order. In particular, the probability of this copy being a copy of device$(A,q)$, is at least \( \frac{nk - s}{nk|L_A| - s} \). Indeed, before that moment we visited exactly $s$ copies of devices from $L_A$ and at most $s$ copies of device$(A,q)$. Hence, there are exactly $nk|L_A| - s$ yet unvisited copies of device from $L_A$, and at least $nk - s$ of them are copies of device$(A,q)$.

Now, consider first $nk$ such moments. In $(s + 1)$-st of them, $B$ “guesses” the correct device from $L_A$, and, therefore, enters never-ending winning spree against $A$, with probability at least \( \frac{nk - s}{nk|L_A| - s} \). Therefore, the probability of $B$ not winning in any of first $nk$ such moments is at most \( \prod_{s=0}^{nk-1} (1 - \frac{nk - s}{nk}) \), because $nk$ events did not happen, with probability of $(s + 1)$-st being at least $\frac{nk - s}{nk}$. Substituting $i := nk - s$, we can simplify the last expression a bit: \( \prod_{s=0}^{nk-1} (1 - \frac{nk - s}{nk}) = \prod_{i=1}^{nk} (1 - \frac{i}{nk} \cdot \frac{i}{nk}) \).

Now, it is enough to prove that, for correct choice of $k$, $|\mathcal{A}| \cdot \prod_{i=1}^{nk} (1 - \frac{i}{nk}) < 1$. Indeed, $|\mathcal{A}| = n^{2n+1}2^n$ ($n$ ways to choose initial state, $n^2$ ways to choose $a$-transitions and $b$-transitions and $2^n$ ways to choose output function), hence $|\mathcal{A}| \cdot \prod_{i=1}^{nk} (1 - \frac{i}{nk}) < 2^n n^{2n+1} \prod_{i=nk/2}^{nk} (1 - \frac{i}{nk}) < 2^n n^{2n+1}(1 - \frac{1}{2n})^{nk/2} < 2^n n^{2n+1}e^{-k/4} < 1$, for $k > 4(2n + 1) \ln n + 4n \ln 2$.

Finally, because we copied each of $|L| = \sum_{i=1}^{n} |L_i| < \sum_{i=1}^{n} 2^{i+1}i < 2^{n+2}n$ devices $nk$ times, each $B \in \mathcal{B}$ has at most $2^n r(n)$ states for some polynomial $r$. \( \square \)

6 Conclusion

It seems that the Theorem 5 can be generalised to the case of general zero-sum game. Indeed, for a game $G = (\Sigma_1, \Sigma_2, R)$ define a function response: $\Sigma_1 \to \Sigma_2$, such that response$(x) = \text{argmin}_{y \in \Sigma_2} R(x, y)$ (if there are several possible values of $y$ with minimal possible $R(x, y)$, choose any of them). That is, response$(x)$ is the best response of the second player to the move $x$ of the first player. By definition, maxmin$(G) = \max_{x \in \Sigma_1} R(x, \text{response}(x))$. Call a situation, where the first player made a move $x \in \Sigma_1$
and the second player made a move response\((x)\), a win for the second player.

Then, from the second player’s perspective, winning all rounds, except for finitely many of them, is an easier task compared to achieving mean payoff \(\text{maxmin}(G)\) or better.

It seems that the proof of the Theorem 5 will not change much if we replace the set of moves \(\Sigma = \{0, 1\}\) with the set \(\Sigma_1\) for the first player and the set \(\text{response}(\Sigma_1)\) for the second player.

Hence, we can state the following conjecture:

**Conjecture 2.** \(n^2k^n.n^n.h\) in the Theorem B can be replaced with \(\min(k, h)n^n\cdot r(n, \log k, \log h)\), for some polynomial \(r\) of three variables.

The above reasoning seems to prove that this conjecture is true, but more thorough studying of possible pitfalls is needed to conclude that the conjecture is indeed proven.

On the other hand, either proving or refuting the conjecture 1 seems to be completely out of reach now. Admittedly, I do not know any plausible explanation of the observed phenomena, moreover the numerical evidence is very fickle, so the conjecture may as well be incorrect.

References

[1] N. Alon, J. H. Spencer, *The probabilistic method*, Wiley-Interscience, 2000

[2] J. S. Banks, R. K. Sundaram, “Repeated games, finite automata and complexity”, *Games and economic behaviour*, 2 (1990), 97–117

[3] E. Ben-Porath, “Repeated games with finite automata”, *Journal of economic theory*, 59 (1993), 17–32

[4] B. Jean, K. Juhani, “Combinatorics on Words - A Tutorial”, *Bulletin of the EATCS*, 79 (2003)

[5] A. Neyman, “Bounded complexity justifies cooperation in the finitely repeated prisoners’ dilemma”, *Economics letters*, 19 (1985), 227–229

[6] A. Neyman, “Finitely repeated games with finite automata”, *Mathematics of operations research*. 23:3 (1998), 513–551

[7] A. Neyman, D. Okada, “Repeated games with bounded entropy”, *Games and economic behaviour*, 30 (2000), 228–247