The Complexity of Games on Highly Regular Graphs (Extended Abstract)

Constantinos Daskalakis[*] Christos H. Papadimitriou[†]

July 7, 2005

Abstract

We present algorithms and complexity results for the problem of finding equilibria (mixed Nash equilibria, pure Nash equilibria and correlated equilibria) in games with extremely succinct description that are defined on highly regular graphs such as the d-dimensional grid; we argue that such games are of interest in the modelling of large systems of interacting agents. We show that mixed Nash equilibria can be found in time exponential in the succinct representation by quantifier elimination, while correlated equilibria can be found in polynomial time by taking advantage of the game's symmetries. Finally, the complexity of determining whether such a game on the d-dimensional grid has a pure Nash equilibrium depends on d and the dichotomy is remarkably sharp: it is solvable in polynomial time (in fact **NL**-complete) when $d = 1$, but it is **NEXP**-complete for $d \geq 2$.

[*]UC Berkeley, Computer Science Division, Soda Hall, Berkeley, CA 94720. Email: `costis@cs.berkeley.edu`. Part of this work was done while the first author was a senior undergraduate student at the National Technical University of Athens.

[†]UC Berkeley, Computer Science Division, Soda Hall, Berkeley, CA 94720. Supported by an NSF ITR grant and a Microsoft Research grant. Email: `christos@cs.berkeley.edu`.

1 Introduction

In recent years there has been some convergence of ideas and research goals between game theory and theoretical computer science, as both fields have tried to grapple with the realities of the Internet, a large system connecting optimizing agents. An important open problem identified in this area is that of computing a mixed Nash equilibrium; the complexity of even the 2-player case is, astonishingly, open (see, e.g., [9, 15]). Since a mixed Nash equilibrium is always guaranteed to exist, ordinary completeness techniques do not come into play. The problem does fall into the realm of "exponential existence proofs" [11], albeit of a kind sufficiently specialized that, here too, no completeness results seem to be forthcoming. On the other hand, progress towards algorithms has been very slow (see, e.g., [10, 7]).

We must mention here that this focus on complexity issues is not understood and welcome by all on the other side. Some economists are mystified by the obsession of our field with the complexity of a problem (Nash equilibrium) that arises in a context (rational behavior of agents) that is not computational at all. *We believe that complexity issues are of central importance in game theory,* and not just the result of professional bias by a few computer scientists. The reason is simple: Equilibria in games are important concepts of rational behavior and social stability, reassuring existence theorems that enhance the explanatory power of game theory and justify its applicability. An intractability proof would render these existence theorems largely moot, and would cast serious doubt on the modelling power of games. How can one have faith in a model predicting that a group of agents will solve an intractable problem? In the words of Kamal Jain: "If your PC cannot find it, then neither can the market."

However, since our ambition is to model by games the Internet and the electronic market, we must extend our complexity investigations well beyond 2-person games. This is happening: [2, 4, 5, 12, 10] investigate the complexity of multi-player games of different kinds. But there is an immediate difficulty: Since a game with n players and s strategies each needs ns^n numbers to be specified (see Section 2 for game-theoretic definitions) the input needed to define such a game is exponentially long. This presents with two issues: First, a host of tricky problems become easy just because the input is so large. More importantly, exponential input makes a mockery of claims of relevance: No important problem can need an astronomically large input to be specified (and we *are* interested in large n, and of course $s \geq 2$). Hence, all work in this area has focused on certain natural classes of *succinctly representable games*.

One important class of succinct games is that of the *graphical games* proposed and studied by Michael Kearns et al. [4, 5]. In a graphical game, we are given a graph with the players as nodes. It is postulated that an agent's utility depends on the strategy chosen by the player *and by the player's neighbors in the graph.* Thus, such games played on graphs of bounded degree can be represented by polynomially many (in n and s) numbers. Graphical games are quite plausible and attractive as models of the interaction of agents across a large network or market. There has been a host of positive complexity results for this kind of games. It has been shown, for example, that correlated equilibria (a sophisticated equilibrium concept defined in Section 2) can be computed in polynomial time for graphical games that are trees [4], later extended to all graphical games [10].

But if we are to study truly large systems of thousands or millions of interacting agents, it is unrealistic to assume that we know the arbitrarily complex details of the underlying interaction graph and of the behavior of every single player — the size of such description would be forbidding anyway. One possibility, explored brilliantly in the work of Roughgarden and Tardos [14], is to assume a continuum of behaviorally identical players. In the present paper we explore an alternative model of large populations of users, within the realm of graphical games. Imagine that the interaction graph is perhaps the $n \times n$ grid, and that all players are locally identical (our results apply to many highly regular topologies of graphs and the case of several player classes). The representation of such a game would then be extremely succinct: Just the game played at each locus, and n, the size of the grid. *Such games, called highly regular graph games, are the focus of* *this paper.* For concreteness and economy of description, we mainly consider the homogeneous versions (without boundary phenomena) of the highly regular graphs (cycle in 1 dimension, torus in 2, and so on); however, both our positive and negative results apply to the grid, as well as all reasonable generalizations and versions (see the discussion after Theorem 5.1).

We examine the complexity of three central equilibrium concepts: *pure Nash equilibrium*, *mixed Nash equilibrium* and the more general concept of *correlated equilibrium*. Pure Nash equilibrium may or may not exist in a game, but, when it does, it is typically much easier to compute than its randomized generalization (it is, after all, a simpler object easily identified by inspection). Remarkably, in highly regular graph games this is reversed: By a symmetry argument combined with quantifier elimination [1, 13], we can compute a (succinct description of a) mixed Nash equilibrium in a d-dimensional highly regular graph game in exponential time (see theorem 3.4; recall that the best known algorithms for even 2-player Nash equilibria are exponential in the worst case). In contrast, regarding pure Nash equilibria, we establish an interesting dichotomy: The problem is polynomially solvable (and **NL**-complete) for $d = 1$ (the cycle) but becomes **NEXP**-complete for $d \geq 2$ (the torus and beyond). The algorithm for the cycle is based on a rather sophisticated analysis of the cycle structure of the Nash dynamics of the basic game. **NEXP**-completeness is established by a generic reduction which, while superficially quite reminiscent of the tiling problem [6], relies on several novel tricks for ensuring faithfulness of the simulation. Finally, our main algorithmic result states that a succinct description of a correlated equilibrium in a highly regular game of any dimension can be computed in polynomial time.

2 Definitions

In a *game* we have *n players* $1, \ldots, n$. Each player p, $1 \leq p \leq n$, has a finite set of *strategies* or *choices*, S_p with $|S_p| \geq 2$. The set $S = \prod_{i=1}^{n} S_i$ is called *the set of strategy profiles* and we denote the set $\prod_{i \neq p} S_i$ by S_{-p}. The *utility* or *payoff* function of player p is a function $u_p : S \to \mathbb{N}$. The *best response function of player* p is a function $\mathrm{BR}_{u_p} : S_{-p} \to 2^{S_p}$ defined by

$$\mathrm{BR}_{u_p}(s_{-p}) \triangleq \{s_p \in S_p | \forall s'_p \in S_p : u_p(s_{-p}; s_p) \geq u_p(s_{-p}; s'_p)\}$$

that is, for every $s_{-p} \in S_{-p}$, $\mathrm{BR}_{u_p}(s_{-p})$ is the set of all strategies s_p of player p that yield the maximum possible utility given that the other players play s_{-p}.

To specify a game with n players and s strategies each we need ns^n numbers, an amount of information exponential in the number of players. However, players often interact with a limited number of other players, and this allows for much more succinct representations:

Definition 2.1 A *graphical game* is defined by:

- A graph $G = (V, E)$ where $V = \{1, \ldots, n\}$ is the set of players.
- For every player $p \in V$:
  - A non-empty finite set of *strategies* S_p
  - A *payoff function* $u_p : \prod_{i \in \mathcal{N}(p)} S_i \to \mathbb{N}$ (where $\mathcal{N}(p) = \{p\} \cup \{v \in V | (p, v) \in E\}$)

Graphical games can achieve considerable succinctness of representation. But if we are interested in modelling huge populations of players, we may need, and may be able to achieve, even greater economy of description. For example, it could be that the graph of the game is highly regular and that the games played at each neighborhood are identical. This can lead us to an extremely succinct representation of the game — logarithmic in the number of players. The following definition exemplifies these possibilities.

Definition 2.2 A d-*dimensional torus game* is a graphical game with the following properties:

- The graph $G = (V, E)$ of the game is the d-*dimensional torus*:
  - $V = \{1, \ldots, m\}^d$
  - $((i_1, \ldots, i_d), (j_1, \ldots, j_d)) \in E$ if there is a $k \leq d$ such that:

$$j_k = i_k \pm 1(\bmod m) \text{ and } j_r = i_r, \text{ for } r \neq k$$

- All the m^d players are identical in the sense that:
  - they have the same strategy set $\Sigma = \{1, \ldots, s\}$
  - they have the same utility function $u : \Sigma^{2d+1} \to \mathbb{N}$

Notice that a torus game with utilities bounded by u_{max} requires $s^{2d+1} \log |u_{max}| + \log m$ bits to be represented.

A torus game is *fully symmetric* if it has the additional property that the utility function u is symmetric with respect to the $2d$ neighbors of each node. Our negative results will hold even for this special case, while our positive results will apply to all torus games.

We could also define torus games with *unequal sides* and *grid games:* torus games where the graph does not wrap around at the boundaries, and so $d + 1$ games must be specified, one for the nodes in the middle and one for each type of boundary node. Furthermore, there are the fully symmetric special cases for each. It turns out that very similar results would hold for all such kinds. We sketch the necessary modifications of the proofs whenever it is necessary and/or expedient.

Consider a game G with n players and strategy sets $S_1, \ldots, S_n$. For every strategy profile s, we denote by s_p the strategy of player p in this strategy profile and by s_{-p} the $(n-1)$-tuple of strategies of all players but p. For every $s'_p \in S_p$ and $s_{-p} \in S_{-p}$ we denote by $(s_{-p}; s'_p)$ the strategy profile in which player p plays s'_p and all the other players play according to s_{-p}. Also, we denote by $\Delta(A)$ the set of probability distributions over a set A and we'll call the set $\prod_{i=1}^{n} \Delta(S_i)$ *set of mixed strategy profiles* of the game G. For a mixed strategy profile σ and a mixed strategy σ'_p of player p, the notations σ_p, σ_{-p} and $(\sigma_{-p}; \sigma'_p)$ are analogous to the corresponding notations for the strategy profiles. Finally, by $\sigma(s)$ we'll denote the probability distribution in product form $\sigma_1(s_1)\sigma_2(s_2)\ldots\sigma_n(s_n)$ that corresponds to the mixed strategy profile σ.

Definition 2.3 A strategy profile s is a *pure Nash equilibrium* if for every player p and strategy $t_p \in S_p$ we have $u_p(s) \geq u_p(s_{-p}; t_p)$.

Definition 2.4 A mixed strategy profile σ of a game $G = \langle n, \{S_p\}_{1 \leq p \leq n}, \{u_p\}_{1 \leq p \leq n} \rangle$ is a *mixed Nash equilibrium* if for every player p and for all mixed strategies $\sigma'_p \in \Delta(S_p)$ the following is true: $\mathbb{E}_{\sigma(s)}[u_p(s)] \geq \mathbb{E}_{(\sigma_{-p}, \sigma'_p)(s)}[u_p(s)]$, where by $\mathbb{E}_{f(s)}[u_p(s)]$, $f \in \Delta(S)$, we denote the expected value of the payoff function $u_p(s)$ under the distribution f.

Definition 2.5 A probability distribution $f \in \Delta(S)$ over the set of strategy profiles of a game G is a *correlated equilibrium* iff for every player p, $1 \leq p \leq n$, and for all $i, j \in S_p$, the following is true:

$$\sum_{s_{-p} \in S_{-p}} [u_p(s_{-p}; i) - u_p(s_{-p}; j)] f(s_{-p}; i) \geq 0$$

Every game has a mixed Nash Equilibrium [8] and, therefore, a correlated equilibrium, since, as can easily be checked, a mixed Nash equilibrium is a correlated equilibrium in product form. However, it may or may not have a pure Nash equilibrium.

The full description of an equilibrium of any kind in a torus game would require an exponential (doubly exponential in the correlated case) number of bits. Accordingly, our algorithms shall always output some kind of succinct representation of the equilibrium, from which one can generate the equilibrium in output polynomial time. In other words, a *succinct representation* of an equilibrium (or any other object) x is a string y such that $|y|$ is polynomial in the input size and $x = f(y)$ for some function f computable in time polynomial in $|x| + |y|$.

3 Mixed Nash Equilibria

We start with a theorem due to Nash ([8]).

Definition 3.1 An *automorphism* of a game $G = \langle n, \{S_p\}, \{u_p\}\rangle$ is a permutation ϕ of the set $\bigcup_{p=1}^{n} S_p$ along with two induced permutations of the players ψ and of the strategy profiles χ, with the following properties:

- $\forall p, \forall x, y \in S_p$ there exists $p' = \psi(p)$ such that $\phi(x) \in S_{p'}$ and $\phi(y) \in S_{p'}$
- $\forall s \in S, \forall p : u_p(s) = u_{\psi(p)}(\chi(s))$

Definition 3.2 A mixed Nash equilibrium of a game is *symmetric* if it is invariant under all automorphisms of the game.

Theorem 3.3 *[8]Every game has a symmetric mixed Nash equilibrium.*

Now we can prove the following.

Theorem 3.4 *For any $d \geq 1$, we can compute a succinct description of a mixed Nash equilibrium of a d-dimensional torus game in time polynomial in $(2d)^s$, the size of the game description, and the number of bits of precision required, but independent of the number of players.*

Proof: Suppose we are given a $d-dimensional$ torus game $G = \langle m, \Sigma, u\rangle$ with $n = m^d$ players. By theorem 3.3, game G has a symmetric mixed Nash equilibrium σ. We claim that in σ all players play the same mixed strategy. Indeed for every pair of players p_1, p_2 in the torus, there is an automorphism (ϕ, ψ, χ) of the game such that $\psi(p_1) = p_2$ and ϕ maps the strategies of player p_1 to the same strategies of player p_2. (In this automorphism, the permutation ψ is an appropriate d-dimensional cyclic shift of the players and permutation ϕ always maps strategies of one player to the same strategies of the player's image.) Thus in σ every player plays the same mixed strategy.

It follows that we can describe σ succinctly by giving the mixed strategy σ_x that every player plays. Let's suppose that $\Sigma = \{1, 2, \ldots, s\}$. For all possible supports $T \subseteq 2^{\Sigma}$, we can check if there is a symmetric mixed Nash equilibrium σ with support T^n as follows. Without loss of generality let's suppose that $T = \{1, 2, \ldots, j\}$ for some $j, j \leq s$. We shall construct a system of polynomial equations and inequalities with variables $p_1, p_2, \ldots, p_j$, the probabilities of the strategies in the support.

Let us call E_l the expected payoff of an arbitrary player p if s/he chooses the pure strategy l and every other player plays σ_x. E_l is a polynomial of degree $2d$ in the variables $p_1, p_2, \ldots, p_j$. Now σ_x is a mixed

Nash equilibrium of the game if and only if the following conditions hold (because of the symmetry, if they hold for one player they hold for every player of the torus):

$$E_l = E_{l+1}, \forall l \in \{1, \ldots, j-1\}$$
$$E_j \geq E_l, \forall l \in \{j+1, \ldots, s\}$$

We need to solve s simultaneous polynomial equations and inequalities of degree $2d$ in $O(s)$ variables. It is known — see [13] — that this problem can be solved in time polynomial in $(2d)^s$, the number of bits of the numbers in the input and the number of bits of precision required. Since the number of bits required to define the system of equations and inequalities is polynomial in the size of the description of the utility function, we get an algorithm polynomial in $(2d)^s$, the size of the game description and the number of bits of precision required, but independent of the number of players. ■

4 An Algorithm for Correlated Equilibria

We show that we can compute a succinct description of a correlated equilibrium of any torus game in time polynomial in the size of the game description.

Theorem 4.1 *Given any torus game, we can compute a succinct representation of a correlated equilibrium in time polynomial in the description of the game.*

Proof: It is obvious from the definition of correlated equilibrium that computing one requires computing s^{m^d} numbers. To achieve polynomial time we will not compute a correlated equilibrium f, but the marginal probability of a correlated equilibrium in the neighborhood of one player p. We then will show that the computed marginal can be extended in a systematic way (and in fact in time polynomial in the output complexity) to a correlated equilibrium.

The construction is easier when m is a multiple of 2d+1, and thus we shall assume first that this is the case. Let us rewrite the defining inequalities of a correlated equilibrium as follows:

$$\forall i, j \in \Sigma: \sum_{s_{neigh} \in \Sigma^{2d}} [u(s_{neigh}; i) - u(s_{neigh}; j)] \sum_{s_{oth} \in \Sigma^{m^d - 2d - 1}} f(s_{oth}; s_{neigh}; i) \geq 0 \quad (1)$$

$$\Leftrightarrow \forall i, j \in \Sigma: \sum_{s_{neigh} \in \Sigma^{2d}} [u(s_{neigh}; i) - u(s_{neigh}; j)] f_p(s_{neigh}; i) \geq 0 \quad (2)$$

where f_p is the marginal probability corresponding to player p and the $2d$ players in p's neighborhood. Now, if $x^{(p)}$ is the $s^{2d+1} \times 1$ vector of the unknown values of the marginal f_p, then by appropriate definition of the $s^2 \times s^{2d+1}$ matrix U we can rewrite inequalities (2) as follows:

$$Ux^{(p)} \geq 0$$

and we can construct the following linear program:

$$\max \sum_i x_i^{(p)}$$
$$Ux^{(p)} \geq 0$$
$$1 \geq x^{(p)} \geq 0$$

whose solution $x^{(p)}$ defines an unnormalized distribution that might be the marginal distribution of player p's neighborhood in a correlated equilibrium of the game. We note that a non-zero solution of the linear program is guaranteed from the existence of a correlated equilibrium.

In order to have a guarantee that the distribution corresponding to the solution of the linear program can be extended to a correlated equilibrium, we shall add some further constraints to our linear program requiring from the solution to have some symmetry. For now, let's assume that we add $O(s^{2d+1} \cdot (2d+1)!)$ symmetry constraints so that the distribution defined by the solution of the linear program is symmetric with respect to its arguments. It will be clear later in the proof that in fact we don't need full symmetry and that $O(s^{2d+1} \cdot 2d)$ symmetry constraints are enough. The new linear program will be the following:

$$\begin{aligned} &\max \sum_i x_i^{(p)} \\ &Ux^{(p)} \geq 0 \\ &(\textit{symmetry constraints}) \\ &1 \geq x^{(p)} \geq 0 \end{aligned}$$

As noted in section 3, game G possesses a symmetric mixed Nash equilibrium in which all players play the same mixed strategy and, thus, a correlated equilibrium that is in product form and symmetric with respect to all the players of the game. Therefore, our linear program has at least one non-zero solution.

Let $x^{(p)*}$ be the solution of the linear program after normalization. Solution $x^{(p)*}$ defines a probability distribution $g(s_0, s_1, \dots, s_{2d})$ over the set Σ^{2d+1} which is symmetric with respect to its arguments. We argue that every such distribution can be extended in a systematic way (in fact, by an algorithm polynomial in the output complexity) to a correlated equilibrium for the game G, ***provided m is a multiple of 2d+1***. We, actually, only need to show that we can construct a probability distribution $f \in \Delta(\Sigma^{m^d})$ with the property that the marginal probability of the neighborhood of every player is equal to the probability distribution g. Then, by the definition of g, inequalities (1) will hold and, thus, f will be a correlated equilibrium of the game.

We first give the intuition behind the construction. Probability distribution g can be seen as the joint probability of $2d+1$ random variables $X_0, X_1, \dots, X_{2d}$. If we can "tile" the d-dimensional torus with the random variables $X_0, X_1, \dots, X_{2d}$ in such a way that all the $2d+1$ random variables appear in the neighborhood of every player (as in figure 1), then we can define probability distribution f as the probability distribution that first draws a sample $(l_0, l_1, ..., l_{2d})$ according to g and then assigns strategy l_i to all players that are tiled with X_i for all $i \in \{0, 1, \dots, 2d\}$. This way the marginal distribution of f in every neighborhood will be the same as g (since g is symmetric).

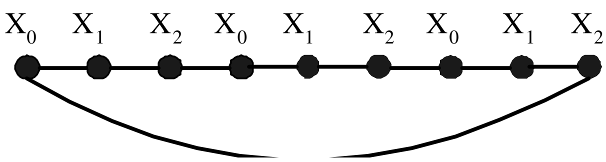

Figure 1: A tiling of the ring with the random variables X_0, X_1, X_2. We generalize this "periodic tiling" in d-dimensions.

We now show how this can be done in a systematic way. For convenience, let us fix an arbitrary player of the d-dimensional torus as the origin and assign orientation to each dimension, so that we can label each player x of the torus with a name $(x_1, x_2, \dots, x_d), x_i \in \{1, \dots, m\}, \forall i$. The configurations of strategies in the support of f will be in a one-to-one correspondence with the configurations in the support of g. More specifically, for every configuration $s = (l_0^s, l_1^s, \ldots, l_{2d}^s)$ in the support of g, we include in the support of f the configuration in which every player $(x_1, x_2, \ldots, x_d)$ plays strategy $l^s_{(x_1+2x_2+3x_3+\ldots+dx_d \textbf{ mod } 2d+1)}$*. So we define the support of the distribution f to be:

$$\mathcal{S}_f = \{s \in \Sigma^{m^d} | \exists (l_0^s, l_1^s, \ldots, l_{2d}^s) \in \mathcal{S}_g \text{ s.t. } s_x = l_{(x_1+2x_2+3x_3+\ldots+dx_d \textbf{ mod } 2d+1)}\}$$

and the distribution itself to be:

$$f(s) = \begin{cases} 0, & \text{if } s \notin \mathcal{S}_f \\ g(l_0^s, l_1^s, \ldots, l_{2d}^s), & \text{if } s \in \mathcal{S}_f \end{cases}$$

By the symmetry of function g and the definition of the support $\mathcal{S}_f$, it is not difficult to see that, if m is a multiple of $2d+1$, the distribution f has the property that the marginal distribution of every player's neighborhood is equal to the distribution g†. This completes the proof. Additionally, we note that for our construction to work we don't need g to be fully symmetric. We only need it to be equal to $O(2d)$ out of the total number of $O((2d+1)!)$ functions that result by permuting its arguments. For example, in the case of the ring presented in figure (1) it suffices that:

$$g(x_0, x_1, x_2) = g(x_1, x_2, x_0) = g(x_2, x_0, x_1)$$

So the number of symmetry constraints we added to the linear program can be reduced to $O(s^{2d+1} \cdot 2d)$.

To generalize this result to the case in which m is not a multiple of 2d+1, let us first reflect on the reason why the above technique fails in the case where m is not a multiple of $2d+1$. If m is not a multiple of $2d+1$, then, if we are given an arbitrary probability distribution g defined in the neighborhood of one player, we cannot always find a probability distribution on the torus so that its marginal in the neighborhood of every player is the same as g, even if g is symmetric‡. As an illustration (but not a proof) of this, observe that the periodic tiling scheme that we used above fails if the size of the ring is not a multiple of 3 (see figure 2).

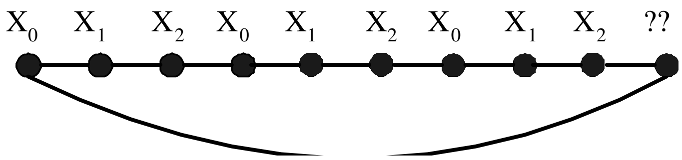

Figure 2: An attempt to tile the ring with the random variables X_0, X_1, X_2 using the periodic tiling scheme. The tiling fails when m is not a multiple of 3(=$2d+1$).

Thus, instead of starting of with the computation of a probability distribution in the neighborhood of one player, we compute a probability distribution h with an augmented number of arguments. Let's call $\upsilon = m \textbf{ mod } 2d+1$. Probability distribution h will have the following properties:

*In terms of tiling the d-dimensional torus with the random variables $X_0, X_1, \ldots, X_{2d}$, the construction that we describe assigns variable X_0 to the origin player and then assigns variables periodically with step 1 in the first dimension, step 2 in the second dimension etc.

†To prove this, we only need to observe that for every player $(x_1, x_2, \ldots, x_d)$ and for every $k \in \{0, 1, \ldots, 2d\}$ there exists an index $i \in \{1, 2, \ldots, d\}$ such that

$$k = x_1 + 2x_2 + 3x_3 + \ldots + ix_i' + \ldots + dx_d \textbf{ mod } 2d+1$$

where $x_i' \in \{x_i, x_i + 1 \textbf{ mod } m, x_i' = x_i - 1 \textbf{ mod } m\}$, if m is a multiple of $2d+1$.

‡There is actually a very simple counterexample that we omit here.

- it will have $2 \times (2d+1)$ arguments
- it will be symmetric with respect to its arguments (we'll relax this requirement later in the proof to get fewer symmetry constraints for our linear program)
- the marginal distribution f_p of the first $2d+1$ variables will satisfy inequalities (2); then, because of the symmetry of the function, the marginal of every ordered subset of 2d+1 arguments will satisfy inequalities (2) (this will also be relaxed)

Again, the existence of a probability distribution h with the above properties follows from the existence of a symmetric mixed Nash equilibrium in which all players play the same mixed strategy. Moreover, h can be found in time polynomial in the size of the game description: let $y^{(p)}$ be the $s^{2\times(2d+1)} \times 1$ vector of variables corresponding to the probability distribution h and $x^{(p)}$ the $s^{2d+1} \times 1$ vector of variables corresponding to the marginalization of the first $2d+1$ arguments of h. Also, let U be the same matrix we used above. Then we can find an unnormalized distribution h that satisfies the above properties by solving the following linear program:

$$\max \sum_i y_i^{(p)}$$
$$Ux^{(p)} \geq 0$$
$$(\textit{symmetry constraints for vector } y^{(p)})$$
$$(\textit{constraints ``marginalizing'' vector } y^{(p)} \textit{ to vector } x^{(p)})$$
$$1 \geq y^{(p)} \geq 0$$

By normalizing the solution of the linear program we get a probability distribution h with the properties that we stated.

To conclude the proof we show how we can use h to produce a correlated equilibrium of the game in a systematic way (and, in fact, with polynomial output complexity). Before doing so, we make the following observations (again let us consider an arbitrary player of the torus as the origin, and let us assign orientation to each dimension so that every player x of the torus has a name $(x_1, x_2, \ldots, x_d)$ where $x_i \in \{1, \ldots, m\}, \forall i$):

- Probability distribution h can be seen as the joint probability distribution of $2 \times (2d+1)$ random variables $X_0, X_1, \ldots, X_{2d}, Z_0, Z_1, ..., Z_{2d}$. If we can "tile" the d-dimensional torus with these random variables in such a way that the neighborhood of every player contains $2d+1$ distinct random variables, then this tiling implies a probability distribution on the torus with the property that the marginal of every player's neighborhood is equal to f_p and thus is a correlated equilibrium.
- Given $2d+1$ random variables, we can use the tiling scheme described above to tile **any d-dimensional grid** in such a way that every neighborhood of size i has i distinct random variables. However, if we "fold" the grid to form the torus this property might not hold if m is not a multiple of $2d+1$; there might be player with at least one coordinate equal to 1 or m (who before was at the boundary of the grid) whose neighborhood does not have $2d+1$ distinct random variables.

Following this line of thought, if we can partition the players of the d-dimensional torus in disjoint d-dimensional grids and we tile every grid with a set of random variables so that every two neighboring grids are tiled with disjoint sets of random variables, then we automatically define a tiling with the required properties.

To do so, we partition the d-dimensional torus in 2^d grids Γ_t, $t \in \{0,1\}^d$, where grid Γ_t is the subgraph of the d-dimensional torus defined by players with names $(x_1, x_2, \ldots, x_d)$, where $x_i \in \{1, 2, \ldots, m-(2d+1+v)\}$ if the i-th bit of t is 0 and $x_i \in \{m-(2d+1+v)+1, \ldots, m\}$ if the i-th bit of t is 1. For every grid Γ_t, let's call $n(\Gamma_t)$ the number of 1's in t[§]. The following observation is key to finishing the proof:

Lemma 4.2 *If two grids Γ_t, $\Gamma_{t'}$ are neighboring then $n(\Gamma_t) = n(\Gamma_{t'}) + 1$ or $n(\Gamma_t) = n(\Gamma_{t'}) - 1$.*

Using lemma 4.2, we can tile the d-dimensional torus as follows: if a grid Γ_t has even $n(\Gamma_t)$ then use random variables $X_0, X_1, \ldots, X_{2d}$ and the tiling scheme described above to tile it; otherwise use random variables $Z_0, Z_1, \ldots, Z_{2d}$ to tile it. This completes the proof. Additionally, we note that for our construction to work we don't need h to be fully symmetric: we only need it to be equal to $O(d)$ out of the total number of $O((2 \times (2d+1))!)$ functions that result by permuting its arguments. So we need a polynomial in the game complexity number of symmetry constraints. ■

5 The Complexity of Pure Nash Equilibria in d Dimensions

In this section we show our dichotomy result: Telling whether a d-dimensional torus game has a pure Nash equilibrium is **NL**-complete if $d = 1$ and **NEXP**-complete if $d > 1$.

5.1 The Ring

Theorem 5.1 *Given a 1-dimensional torus game we can check whether the game has a pure Nash equilibrium in polynomial time, and in fact in nondeterministic logarithmic space.*

Proof:

Given such a game we construct a directed graph $T = (V_T, E_T)$ as follows.

$$V_T = \{(x, y, z) \mid x, y, z \in \Sigma : y \in \mathrm{BR}_u(x, z)\}$$
$$E_T = \{(v_1, v_2) \mid v_1, v_2 \in V_T : v_{1y} = v_{2x} \wedge v_{1z} = v_{2y}\}$$

It is obvious that the construction of the graph T can be done in time polynomial in s and that $|V_T| = O(s^3)$. Thus the adjacency matrix A_T of the graph has $O(s^3)$ rows and columns.

We now prove the following lemma:

Lemma 5.2 *G has a pure Nash equilibrium iff there is a closed walk of length m in the graph T*

Proof: ($\Rightarrow$) Suppose that game G has a pure Nash equilibrium $s = (s_0, s_2, ..., s_{m-1})$. From the definition of the pure Nash equilibrium it follows that, for all $i \in [m]$, $s_i \in \mathrm{BR}_u(s_{i-1 \bmod m}, s_{i+1 \bmod m})$. Thus from the definition of graph T it follows that: $t_i = (s_{i-1 \bmod m}, s_i, s_{i+1 \bmod m}) \in V_T$ for all $i \in [m]$ and moreover that $(t_i, t_{i+1 \bmod m}) \in E_T$ for all $i \in [m]$ (note that the t_i's need not be distinct).

It follows that the sequence of nodes $t_1, t_2, \ldots, t_n, t_1$ is a closed walk of length m in the graph T.

($\Leftarrow$) Suppose that there is a closed walk $v_1, v_2, \ldots, v_m, v_1$ in the graph T. Since $v_1, v_2, \ldots, v_m \in V_T$ it follows that each v_i is of the form (x_i, y_i, z_i) and $y_i \in \mathrm{BR}_u(x_i, z_i)$. Moreover, since $(v_i, v_{i+1 \bmod m}) \in E_T$ we have $y_i = x_{i+1 \bmod m} \wedge z_i = y_{i+1 \bmod m}$. It follows that the strategy profile $\langle y_0, y_1, \ldots, y_m \rangle$ is a pure Nash equilibrium. ■

[§] $n(\Gamma_t)$ is the number of dimensions of the grid in which the coordinates take values from the set $\{m-(2d+1+v)+1, \ldots, m\}$

It follows that, in order to check whether the game has a pure Nash equilibrium, it suffices to check whether there exists a closed walk of length m in graph T. This can be easily done in polynomial time, for example by finding A_T^m using repeated squaring.

We briefly sketch why it can be done in nondeterministic logarithmic space. There are two cases: If m is at most polynomial in s (and thus in the size of T), then we can guess the closed walk in logarithmic space, counting up to m. The difficulty is when m is exponential in s, and we must guess the closed walk of length m in space $\log \log m$, that is, without counting up to m. We first establish that every such walk can be decomposed into a short closed walk of length $q \leq s^6$, plus a set of cycles, all connected to this walk, and of lengths $c_1, \ldots, c_r$ (each cycle repeated a possibly exponential number of times) such that the greatest common divisor of $c_1, \ldots, c_r$ divides $m - q$. We therefore guess the short walk of T of length q, guess the r cycles, compute the greatest common divisor of their lengths $g = \gcd(c_1, \ldots, c_r)$ in logarithmic space, and then check whether g divides $m - q$; this latter can be done by space-efficient long division, in space $\log g$ and without writing down $m - q$. Finally, **NL**-completeness is shown by a rather complicated reduction from the problem of telling whether a directed graph has an odd cycle, which can be easily shown to be **NL**-complete. ■

The same result holds when the underlying graph is the 1-dimensional grid (the path); the only difference is that we augment the set of nodes of T by two appropriate node-sets to account for the "left" and "right" -for an arbitrary direction on the path- boundary players; we are now seeking a path of length $m-1$ between those two distinguished subsets of the nodes.

We note that the problem has the same complexity for several generalizations of the path topology such as the *ladder graph* and many others.

5.2 The Torus

The problem becomes **NEXP**-complete when $d > 1$, even in the fully symmetric case. The proof is by a generic reduction.

Theorem 5.3 *For any $d \geq 2$, the problem of deciding whether there exists a pure Nash equilibrium in a fully symmetric d-dimensional torus game is* **NEXP**-*complete.*
Proof: A non-deterministic exponential time algorithm can choose in $O(m^d)$ nondeterministic steps a pure strategy for each player, and then check the equilibrium conditions for each player. Thus, the problem belongs to the class **NEXP**.

Our **NEXP**-hardness reduction is from the problem of deciding, given a one-tape nondeterministic Turing machine M and an integer t, whether there is a computation of M that halts within $5t - 2$ steps. We present the $d = 2$ case, the generalization to $d > 2$ being trivial. Given such M and t, we construct the torus game $G_{M,\,t}$ with size $m = 5t + 1$. Intuitively, the strategies of the players will correspond to states and tape symbols of M, so that a Nash equilibrium will spell a halting computation of M in the "tableau" format (rows are steps and columns are tape squares). In this sense, the reduction is similar to that showing completeness of the *tiling* problem (see figure 3): Can one tile the $m \times m$ square by square tiles of unit side and belonging to certain types, when each side of each type comes with a label restricting the types that can be used next to it? Indeed, each strategy will simulate a tile having on the horizontal sides labels of the form **(*symbol, state*)** whereas on the vertical sides **(*state, action*)**. (Furthermore, as shown in figure 3, each strategy will also be identified by a pair (i, j) of integers in $[5]$, standing for the coordinates modulo 5 of the node that plays this strategy; the necessity of this, as well as the choice of 5, will be come more clear later in the proof).

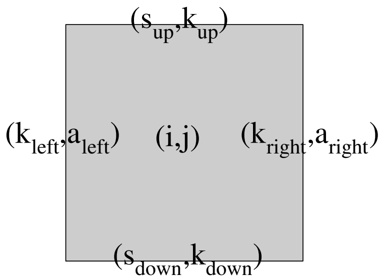

Figure 3: A pure strategy as a tile

Superficially, the reduction now seems straightforward: Have a strategy for each tile type, and make sure that the best response function of the players reflects the compatibility of the tile types. There are, however, several important difficulties in doing this, and we summarize the principal ones below.

difficulty 1 In tiling, the compatibility relation can be partial, in that no tile fits at a place when the neighbors are tiled inappropriately. In contrast, in our problem the best response function must be total.

difficulty 2 Moreover, since we are reducing to fully symmetric games, the utility function must be symmetric with respect to the strategies of the neighbors. However, even if we suppose that for a given 4-tuple of tiles (strategies) for the neighbors of a player there is a matching tile (strategy) for this player, that tile does not necessarily match every possible permutation-assignment of the given 4-tuple of tiles to the neighbors. To put it otherwise, symmetry causes a *lack of orientation*, making the players unable to distinguish among their 'up', 'down', 'left' and 'right' neighbors.

difficulty 3 The third obstacle is the *lack of boundaries* in the torus which makes it difficult to define the strategy set and the utility function in such a way as to ensure that some "bottom" row will get tiles that describe the initial configuration of the Turing machine and the computation tableau will get built on top of that row.

It is these difficulties that require our reduction to resort to certain novel stratagems and make the proof rather complicated. Briefly, we state here the essence of the tricks (we'll omit some technicalities):

solution 1 To overcome the first difficulty, we introduce three special strategies (set $\mathcal{K}$ in the appendix) and we define our utility function in such a way that (a) these strategies are the best responses when we have no tile to match the strategies of the neighbors and (b) no equilibria of the game can contain any of these strategies.

solution 2 To overcome the second difficulty we attach coordinates modulo 5 to all of the tiles that correspond to the interior of the computation tableau (set S_1 in the appendix) and we define the utility function in such a way that in every pure Nash equilibrium a player who plays a strategy with coordinates modulo 5 equal to (i, j) has a neighbor who plays a strategy with each of the coordinates $(i \pm 1, j \pm 1 \bmod 5)$. This implies (through a nontrivial graph-theoretic argument, see Lemma A.2 in the appendix) that the torus is "tiled" by strategies respecting the counting modulo 5 in both dimensions.

solution 3 To overcome difficulty 3, we define the side of the torus to be $5t + 1$ and we introduce strategies that correspond to the boundaries of the computation tableau (set S_2 in the appendix) and are best responses only in the case their neighbors' coordinates are not compatible with the counting modulo 5. The choice of side length makes it impossible to tile the torus without using these strategies and, thus, ensures that one row and one column (at least) will get strategies that correspond to the boundaries of the computation tableau.

We postpone further details to an Appendix. ■

6 Discussion

We have classified satisfactorily the complexity of computing equilibria of the principal kinds in highly regular graph games. We believe that the investigation of computational problems on succinct games, of which the present paper as well as [3] and [12, 10] are examples, will be an active area of research in the future. One particularly interesting research direction is to formulate and investigate games on highly regular graphs in which the players' payoffs depend in a natural and implicit way on the players' location on the graph, possibly in a manner reflecting routing congestion, facility location, etc., in a spirit similar to that of non-atomic games [14].

References

[1] G. E. Collins "Quantifier elimination for real closed fields by cylindrical algebraic decomposition," *Springer Lecture Notes in Computer Science, 33*, 1975, 515–532.

[2] A. Fabrikant, C. H. Papadimitriou, K. Talwar "The Complexity of Pure Nash Equilibria,"*STOC*, 2004, 604–612.

[3] L. Fortnow, R. Impagliazzo, V. Kabanets, C. Umans "On the complexity of succinct zero-sum games," *IEEE Conference on Computational Complexity*, 2005.

[4] S. Kakade, M. Kearns, J. Langford, and L. Ortiz "Correlated Equilibria in Graphical Games," *ACM Conference on Electronic Commerce*, 2003.

[5] M. Kearns, M. Littman, S. Singh "Graphical Models for Game Theory," *Proceedings of UAI*, 2001.

[6] H. Lewis, C. H. Papadimitriou "Elements of the Theory of Computation," *Prentice-Hall*, 1981.

[7] R. J. Lipton, E. Markakis "Nash Equilibria via Polynomial Equations," *LATIN*, 2004, 413–422.

[8] J. Nash "Noncooperative games," *Annals of Mathematics, 54*, 289–295, 1951.

[9] C. H. Papadimitriou "Algorithms, Games, and the Internet," *STOC*, 2001, 749–753.

[10] C. H. Papadimitriou "Computing correlated equilibria in multiplayer games," *STOC*, 2005.

[11] C. H. Papadimitriou "On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence," *J. Comput. Syst. Sci., 48(3)*, 1994, 498–532.

[12] C. H. Papadimitriou, T. Roughgarden "Computing equilibria in multiplayer games," *SODA*, 2005.

[13] J. Renegar "On the Computational Complexity and Geometry of the First-Order Theory of the Reals, I, II, III," *J. Symb. Comput., 13(3)*, 1992, 255–352.

[14] T. Roughgarden, E. Tardos "How bad is selfish routing?," *J. ACM, 49(2)*, 2002, 236-259.

[15] R. Savani, B. von Stengel "Exponentially many steps for finding a Nash equilibrium in a bimatrix game," *FOCS*, 2004.

[16] H. Wang, "Proving theorems by pattern recognition II," *Bell Systems Technical Journal 40*, 1961, 1–42.

A Missing Proofs

Proof of theorem 5.3: We present here some of the omitted technicalities of the proof. Let us define the following language which is not difficult to verify that is **NEXP**-complete (by NTM we denote the set of *one-tape non-deterministic Turing Machines*):

$$L = \{(\langle M \rangle, t), M \in NTM, t \in \mathbb{N} | \text{on empty input } M \text{ halts within } 5t-2 \text{ steps}\}$$

We shall show a reduction from language L to the problem of determining whether a 2-dimensional fully symmetric torus game has a pure Nash equilibrium. That is, given $(\langle M \rangle, t)$ we will construct a 2-dimensional fully symmetric torus game $G_{M,t} = \langle m, \Sigma, u \rangle$ so that:

$$(\langle M \rangle, t) \in L \Leftrightarrow \text{game } G_{M,t} \text{ has a pure Nash Equilibrium} \tag{3}$$

Since NTM is the set of *one tape non-deterministic Turing Machines*, a machine $M \in NTM$ can be described by a tuple $\langle M \rangle = \langle K, \mathcal{A}, \delta, q_0 \rangle$, where K is the set of states, $\mathcal{A}$ is the alphabet, $\delta : K \times \mathcal{A} \to 2^{K \times \mathcal{A} \times \{\to, \leftarrow, -\}}$ is the transition function and q_0 is the initial state of the machine. For proof convenience, we'll make two non-restrictive assumptions for our model of computation:

1. The transition function δ satisfies $(q, a, -) \in \delta(q, a), \forall q \in K, a \in \mathcal{A}$ (transitions *"do nothing"*).
2. The tape of the machine contains in its two leftmost cells the symbols $\rhd \rhd'$ throughout all of the computation; furthermore the head of the machine points initially at the cell that contains $\rhd'$ and the head never reaches the leftmost cell of the tape.

Now let's construct the game $G_{M,t} = \langle m, \Sigma, u \rangle$.

Size of torus: We choose $m = 5t + 1$

Set of strategies Σ: Every pure strategy, except for some special ones that we'll define later, will be an 11-tuple of the form:

$$\sigma = (i, j, s_{down}, k_{down}, s_{up}, k_{up}, a_{left}, k_{left}, a_{right}, k_{right}, v)$$

where (let's give to ω the connotation of "*empty field*"):

- $i, j \in [5] \cup \{\omega\}$ (counters or "empty")
- $s_{down}, s_{up} \in \mathcal{A} \cup \{\omega\}$ (alphabet symbols or "empty")
- $k_{down}, k_{left}, k_{up}, k_{down} \in K \cup \{\omega\}$ (state symbols or "empty")
- $a_{right}, a_{left} \in \{\to, \leftarrow, \omega\}$ (action symbols or "empty")
- $v \in \{B_{\rhd}, B_{\sqcup}, \llcorner, \omega\}$ (special labels or "empty")

We shall henceforth refer to a specific field of a pure strategy using the label of the field as a subscript, for example $\sigma|_i$, $\sigma|_j$ and $\sigma|_{s_{up}}$. Also, it will be helpful to think of pure strategies we can think of them as ***tiles*** (see figure 3 in section 5).

Now we chose $\Sigma = S_1 \cup S_2 \cup \mathcal{K}$ where:

- Set S_1 contains the strategies-tiles that, in a high level, we intend to fill the interior of the computation tableau:
  - $\forall a \in \mathcal{A}, \forall i, j \in [5]$ strategy: $(i, j, a, \omega, a, \omega, \omega, \omega, \omega, \omega, \omega)$
  - $\forall q, p \in K, \forall a, b \in \mathcal{A}$ s.t. $(p, b, -) \in \delta(q, a)$, $\forall i, j \in [5]$ strategy: $(i, j, a, q, b, p, \omega, \omega, \omega, \omega, \omega)$
  - $\forall q, p \in K, \forall a, b \in \mathcal{A}$ s.t. $(p, b, \rightarrow) \in \delta(q, a)$, $\forall \gamma \in \mathcal{A}$, $\forall i, j \in [5]$ strategies:

    $$(i, j, a, q, b, \omega, \omega, \omega, \rightarrow, p, \omega) \text{ and } (i, j, \gamma, \omega, \gamma, p, \rightarrow, p, \omega, \omega, \omega)$$
  - $\forall q, p \in K, \forall a, b \in \mathcal{A}$ s.t. $(p, b, \leftarrow) \in \delta(q, a)$, $\forall \gamma \in \mathcal{A}$, $\forall i, j \in [5]$ strategies:

    $$(i, j, a, q, b, \omega, \leftarrow, p, \omega, \omega, \omega) \text{ and } (i, j, \gamma, \omega, \gamma, p, \omega, \omega, \leftarrow, p, \omega)$$
  - $\forall a \in \mathcal{A}$, $\forall i, j \in [5]$ strategy: $(i, j, a, h, a, h, \omega, \omega, \omega, \omega, \omega)$ (h is the halting state)
  - strategy: $(1, 0, \rhd', \omega, \rhd', q_0, \omega, \omega, \omega, \omega, \omega)$
- Set S_2 contains the strategies-tiles that, in a high level, we intend to span the boundaries of the computation tableau:
  - $\forall i \in [5]$ strategy: $(i, \omega, \omega, \omega, \omega, \omega, \omega, \omega, \omega, \omega, B_{\sqcup})$ (abbreviated: $s^i_{\sqcup}$)

    (these strategies are intended to span the "horizontal" boundary of the computation tableau)
  - $\forall j \in [5]$ strategy: $(\omega, j, \omega, \omega, \omega, \omega, \omega, \omega, \omega, \omega, B_{\rhd})$ (abbreviated: $s^j_{\rhd}$)

    (these strategies are intended to span the "vertical" boundary of the computation tableau)
  - strategy: $(\omega, \omega, \omega, \omega, \omega, \omega, \omega, \omega, \omega, \omega, \llcorner)$ (abbreviated: $s_{\llcorner}$)

    (this strategy is intended to appear at the corners of the computation tableau)
- Set $\mathcal{K}$ contains three special strategies $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3$ that we invent to overcome the first difficulty we stated in the abstract description of the proof (see section 5 and lemma A.1)

It's easy to see that the set Σ we defined can be computed in time $O(|\langle M\rangle|)$, where $|\langle M\rangle|$ is the size of description of the machine M.

Utility Function u: In order to be more concise, we'll define function u in an undirect way. In fact, we'll give some properties that the best response function BR_u must have. It will be obvious that we can construct in polynomial time a function u so that BR_u has these properties. Before stating the properties we note that since u is symmetric with respect to the neighbors, BR_u must be symmetric in all its arguments, so instead of writing $\text{BR}_u(x, y, z, t)$ we can write $\text{BR}_u(\{\{x, y, z, t\}\})$ (by $\{\{x, y, z, t\}\}$ we denote the multiset with elements x, y, z, t). The properties that we require from BR_u are given bellow. We give in comments the intuition behind the definition of each property. The claims that we state, however, should not be taken as proofs. The correctness of the reduction is only established by lemmata A.1 through A.8.

/* *the following properties ensure that lemma A.1 will hold* */

1. If $\mathcal{K}_1 \in \{w, y, z, t\}$ then $\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{\mathcal{K}_2\}$
2. If $w = y = z = t = \mathcal{K}_2$ then: $\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{\mathcal{K}_3\}$

/* *the following property says: one player can play the strategy that stands for the corner of the computation tableau if s/he has two neighbors that play strategies standing for pieces of the horizontal boundary and two neighbors that play strategies standing for pieces of the vertical boundary (note that we try to encode a computation of the machine M on the torus so the horizontal boundary starting from the corner player eventually meets the corner player from the other side and the same holds for the vertical boundary; note also that according to lemma A.6 there can by multiple horizontal and vertical boundaries, in which case the property we stated should hold as well)* */

3. If $w = s^0_{\sqcup}$, $y = s^4_{\sqcup}$, $z = s^0_{\rhd}$, $t = s^4_{\rhd}$ then: $\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{s_{\llcorner}\}$

/* *the following properties make possible the formation of a row encoding the horizontal boundary of the computation tableau; such rows (they can be more than one, see lemma A.6) will serve as down and up boundaries of encoded computations of the machine M on the torus*/

4. If $w = s_{\llcorner}$, $y = s^1_{\sqcup}$, $z = (0, 0, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)$, $t = (0, 4, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)$ then:
$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{s^0_{\sqcup}\}$$
5. If $w = s_{\llcorner}$, $y = s^3_{\sqcup}$, $z = (4, 0, \sqcup, \omega, \sqcup, \omega, \omega, \omega, \omega, \omega, \omega)$, $t = (4, 4, \lambda, \kappa, \lambda, \kappa, \omega, \omega, \omega, \omega, \omega)$, for some $\lambda \in \mathcal{A}$ and $\kappa \in \{\omega, h\}$ then:
$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{s^4_{\sqcup}\}$$
6. If for some $l \in [5]$, $w = s^{(l-1) \mod 5}_{\sqcup}$, $y = s^{(l+1) \mod 5}_{\sqcup}$, $z|_i = l \wedge z|_j = 0$ and $t = (l, 4, \lambda, \kappa, \lambda, \kappa, \omega, \omega, \omega, \omega, \omega)$ for some $\lambda \in \mathcal{A}$ and $\kappa \in \{\omega, h\}$ then:
$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{s^l_{\sqcup}\}$$

/* *the following properties make possible the formation of a column encoding the vertical boundary of the computation tableau; such columns (they can be more than one, see lemma A.6) will serve as left and right boundaries of encoded computations of the machine M on the torus*/

7. If $w = s_{\llcorner}$, $y = s^1_{\rhd}$, $z = (0, 0, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)$, $t = (4, 0, \sqcup, \omega, \sqcup, \omega, \omega, \omega, \omega, \omega, \omega)$ then:
$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{s^0_{\rhd}\}$$
8. If $w = s_{\llcorner}$, $y = s^3_{\rhd}$, $z = (0, 4, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)$, $t = (4, 4, \lambda, \kappa, \lambda, \kappa, \omega, \omega, \omega, \omega, \omega)$ for some $\lambda \in \mathcal{A}$ and $\kappa \in \{\omega, h\}$ then
$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{s^4_{\rhd}\}$$
9. If for some $l \in [5]$, $w = s^{(l-1) \mod 5}_{\rhd}$, $y = s^{(l+1) \mod 5}_{\rhd}$, $z = (0, l, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)$, $t|_i = 4 \wedge t|_j = l$ then:
$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{s^l_{\rhd}\}$$

/** *INTERIOR OF THE COMPUTATION TABLEAU* **/

/* *column encoding the leftmost cell of the tape through an encoded computation of M* */

10. If $w = s^0_\sqcup$, $y = s^0_\rhd$, $z = (0, 1, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)$, $t|_i = 1 \wedge t|_j = 0$ then:

$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{(0, 0, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)\}$$

11. If $w = (0, 3, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)$, $y = s^0_\sqcup$, $z = s^4_\rhd$, $t|_i = 1 \wedge t|_j = 4$ then:

$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{(0, 4, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)\}$$

12. If for some $l \in [5]$:

- $w = (0, (l-1) \mod 5, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)$
- $y = (0, (l+1) \mod 5, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)$
- $z = (\omega, l, \omega, \omega, \omega, \omega, \omega, \omega, \omega, \omega, B_\rhd)$
- $t|_i = 1 \wedge t|_j = l$

then: $\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{(0, l, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)\}$

/* *row encoding the initial configuration of the tape of M*/

13. If $w = s^1_\sqcup$, $y = (0, 0, \rhd, \omega, \rhd, \omega, \omega, \omega, \omega, \omega, \omega)$, $z|_i = 1 \wedge z|_j = 1$, $t|_i = 2 \wedge t|_j = 0$ then:

$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{(1, 0, \rhd', \omega, \rhd', q_0, \omega, \omega, \omega, \omega, \omega)\}$$

14. If for some $l \in [5]$, $w = s^l_\sqcup$, $y|_i = l \wedge y|_j = 1$, $z|_i = (l-1) \mod 5 \wedge z|_j = 0 \wedge z|_{s_{up}} \neq \rhd$, $t|_i = (l+1) \mod 5 \wedge z|_j = 0$ then:

$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{(l, 0, \sqcup, \omega, \sqcup, \omega, \omega, \omega, \omega, \omega, \omega)\}$$

/* *row encoding the configuration of the tape after the last step of the computation; note that we require that the field k_{up} of one of the neighbors must be either h or empty; in this way we force the encoded computation to be halting and the halting state to be reached within at most $5t-2$ steps (lemma A.8)* */

15. If for some $l \in [5]$:

- $w = s^l_\sqcup$
- $y|_i = l \wedge y|_j = 3 \wedge y|_{k_{up}} \in \{h, \omega\}$
- $z = ((l-1) \mod 5, 4, \lambda_1, \kappa_1, \lambda_1, \kappa_1, \omega, \omega, \omega, \omega, \omega)$, where $\lambda_1 \in \mathcal{A}$, $\kappa_1 \in \{\omega, h\}$
- $t = ((l+1) \mod 5, 4, \lambda_2, \kappa_2, \lambda_2, \kappa_2, \omega, \omega, \omega, \omega, \omega)$, where $\lambda_2 \in \mathcal{A}$, $\kappa_2 \in \{\omega, h\}$

then: $\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{(l, 4, y|_{s_{up}}, y|_{k_{up}}, y|_{s_{up}}, y|_{k_{up}}, \omega, \omega, \omega, \omega, \omega)\}$

16. If for some $l \in [5]$:

- $w = (l, \omega, \omega, \omega, \omega, \omega, \omega, \omega, \omega, \omega, B_\sqcup)$
- $y|_i = (l-1) \mod 5 \wedge y|_j = 4$
- $z|_i = (l+1) \mod 5 \wedge z|_j = 4$
- $t|_i = l \wedge t|_j = 3$

then: $\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{(l, 4, t|_{s_{up}}, t|_{k_{up}}, t|_{s_{up}}, t|_{k_{up}}, \omega, \omega, \omega, \omega, \omega)\}$

/* *column encoding the rightmost cell of the tape that the machine reaches in an encoded computation* */

17. If $w = s^4_{\sqcup}$, $y = s^0_{\triangleright}$, $z = (3, 0, \sqcup, \omega, \sqcup, \omega, \omega, \omega, \omega, \omega, \omega)$, $t = (4, 1, \sqcup, \omega, \sqcup, \omega, \omega, \omega, \omega, \omega, \omega)$ then:

$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{(4, 0, \sqcup, \omega, \sqcup, \omega, \omega, \omega, \omega, \omega, \omega)\}$$

18. If $w = s^4_{\sqcup}$, $y = s^4_{\triangleright}$, $z|_i = 4 \wedge z|_j = 3 \wedge z|_{k_{up}} \in \{\omega, h\}$, $t = (3, 4, \lambda, \kappa, \lambda, \kappa, \omega, \omega, \omega, \omega, \omega)$, for some $\lambda \in \mathcal{A}$, $\kappa \in \{\omega, h\}$ then:

$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{(4, 4, z|_{s_{up}}, z|_{k_{up}}, z|_{s_{up}}, z|_{k_{up}}, \omega, \omega, \omega, \omega, \omega)\}$$

19. If, for some $r \in [5]$, $a, b \in \mathcal{A}$, $q, p \in K$, $w|_i = 4$, $w|_j = (r-1) \mod 5$, $w|_{s_{up}} = a$, $w|_{k_{up}} = q$, $y|_i = 4$, $y|_j = (r+1) \mod 5$, $y|_{s_{down}} = b$, $y|_{k_{down}} = p$, $z|_i = 3$, $z|_j = r$, $z|_{a_{right}} = \omega$, $z|_{k_{right}} = \omega$, $t = s^r_{\triangleright}$ then:

$$\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \begin{cases} \{(4, r, a, q, b, p, \omega, \omega, \omega, \omega, \omega)\}, & \text{if } (p, b, -) \in \delta(q, a) \\ \{\mathcal{K}_1\}, & \text{otherwise} \end{cases}$$

20. If for some $r \in [5]$, $a \in \mathcal{A}$ and $p \in K$:

   - $w|_i = 4 \wedge w|_j = (r-1) \mod 5 \wedge w|_{s_{up}} = a \wedge w|_{k_{up}} = \omega$
   - $y|_i = 4 \wedge y|_j = (r+1) \mod 5 \wedge y|_{s_{down}} = a \wedge y|_{k_{down}} = p$
   - $z|_i = 3 \wedge z|_j = r \wedge z|_{a_{right}} =\rightarrow \wedge z|_{k_{right}} = p$
   - $t = s^r_{\triangleright}$

   then: $\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \begin{cases} \{(4, r, a, \omega, a, p, \rightarrow, p, \omega, \omega, \omega)\}, & \text{if } (p, z|_{s_{up}}, \rightarrow) \in \delta(z|_{k_{down}}, a) \\ \{\mathcal{K}_1\}, & \text{otherwise} \end{cases}$

/** *Cells between leftmost and rightmost* **/

/* *if a cell of the tape is not pointed to by the head at a particular step, its value will remain the same* */

21. If for some $l, r \in [5]$, $a \in \mathcal{A}$:

   - $w|_i = l \wedge w|_j = (r-1) \mod 5 \wedge w|_{s_{up}} = a \wedge w|_{k_{up}} = \omega$
   - $y|_i = l \wedge y|_j = (r+1) \mod 5 \wedge y|_{s_{down}} = a \wedge y|_{k_{down}} = \omega$
   - $z|_i = (l-1) \mod 5 \wedge z|_j = r \wedge z|_{a_{right}} = \omega \wedge z|_{k_{right}} = \omega$
   - $t|_i = (l+1) \mod 5 \wedge t|_j = r \wedge t|_{a_{left}} = \omega \wedge t|_{k_{left}} = \omega$

   then: $\mathrm{BR}_u(\{\{w, y, z, t\}\}) = \{(l, r, a, \omega, a, \omega, \omega, \omega, \omega, \omega, \omega)\}$

/* *change the symbol of the cell at which the head of the machine points; if for a particular choice of states and symbols this cannot happen according to the machine's transition function the best response is $\mathcal{K}_1$ (see lemma A.1)* */

22. If for some $l, r \in [5]$, $a, b \in \mathcal{A}$, $q, p \in K$:

   - $w|_i = l \wedge w|_j = (r-1) \mod 5 \wedge w|_{s_{up}} = a \wedge w|_{k_{up}} = q$

- $y|_i = l \wedge y|_j = (r+1) \mod 5 \wedge y|_{s_{down}} = b \wedge y|_{k_{down}} = p$
- $z|_i = (l-1) \mod 5 \wedge z|_j = r \wedge z|_{a_{right}} = \omega \wedge z|_{k_{right}} = \omega$
- $t|_i = (l+1) \mod 5 \wedge t|_j = r \wedge t|_{a_{left}} = \omega \wedge t|_{k_{left}} = \omega$

then: $\mathrm{BR}_u(\{\{w,y,z,t\}\}) = \begin{cases} \{(l,r,a,q,b,p,\omega,\omega,\omega,\omega,\omega)\}, & \text{if } (p,b,-) \in \delta(q,a) \\ \{\mathcal{K}_1\}, & \text{otherwise} \end{cases}$

/* *the following two properties make possible the encoding of a right transition of the machine* */

23. If for some $l, r \in [5]$, $a \in \mathcal{A}$, $p \in K$:
    - $w|_i = l \wedge w|_j = (r-1) \mod 5 \wedge w|_{s_{up}} = a \wedge w|_{k_{up}} = \omega$
    - $y|_i = l \wedge y|_j = (r+1) \mod 5 \wedge y|_{s_{down}} = a \wedge y|_{k_{down}} = p$
    - $z|_i = (l-1) \mod 5 \wedge z|_j = r \wedge z|_{a_{right}} = \rightarrow \wedge z|_{k_{right}} = p$
    - $t|_i = (l+1) \mod 5 \wedge t|_j = r \wedge t|_{a_{left}} = \omega \wedge t|_{k_{left}} = \omega$

    then: $BR_u(\{\{w,y,z,t\}\}) = \begin{cases} \{(l,r,a,\omega,a,p,\rightarrow,p,\omega,\omega,\omega)\}, & \text{if } (p, z|_{s_{up}}, \rightarrow) \in \delta(z|_{k_{down}}, a) \\ \{\mathcal{K}_1\}, & \text{otherwise} \end{cases}$
24. If for some $l, r \in \{0,1,2,3,4\}$, $a, b \in \mathcal{A}$, $p, q \in K$:
    - $w|_i = l \wedge w|_j = (r-1) \mod 5 \wedge w|_{s_{up}} = a \wedge w|_{k_{up}} = q$
    - $y|_i = l \wedge y|_j = (r+1) \mod 5 \wedge y|_{s_{down}} = b \wedge y|_{k_{down}} = \omega$
    - $z|_i = (l-1) \mod 5 \wedge z|_j = r \wedge z|_{a_{right}} = \omega \wedge z|_{k_{right}} = \omega$
    - $t|_i = (l+1) \mod 5 \wedge t|_j = r \wedge t|_{a_{left}} = \rightarrow \wedge t|_{k_{left}} = p$

    then: $BR_u(\{\{w,y,z,t\}\}) = \begin{cases} \{(l,r,a,q,b,\omega,\omega,\omega,\rightarrow,p,\omega)\}, & \text{if } (p,b,\rightarrow) \in \delta(q,a) \\ \{\mathcal{K}_1\}, & \text{otherwise} \end{cases}$

/* *the following two properties make possible the encoding of a left transition of the machine* */

25. If for some $l, r \in [5]$, $a, b \in \mathcal{A}$, $p, q \in K$:
    - $w|_i = l \wedge w|_j = (r-1) \mod 5 \wedge w|_{s_{up}} = a \wedge w|_{k_{up}} = q$
    - $y|_i = l \wedge y|_j = (r+1) \mod 5 \wedge y|_{s_{down}} = b \wedge y|_{k_{down}} = \omega$
    - $z|_i = (l-1) \mod 5 \wedge z|_j = r \wedge z|_{a_{right}} = \leftarrow \wedge z|_{k_{right}} = p$
    - $t|_i = (l+1) \mod 5 \wedge t|_j = r \wedge t|_{a_{left}} = \omega \wedge t|_{k_{left}} = \omega$

    then: $BR_u(\{\{w,y,z,t\}\}) = \begin{cases} \{(l,r,a,q,b,\omega,\leftarrow,p,\omega,\omega,\omega)\}, & \text{if } (p,b,\leftarrow) \in \delta(q,a) \\ \{\mathcal{K}_1\}, & \text{otherwise} \end{cases}$
26. If for some $l, r \in [5]$, $a \in \mathcal{A}$, $p \in K$:
    - $w|_i = l \wedge w|_j = (r-1) \mod 5 \wedge w|_{s_{up}} = a \wedge w|_{k_{up}} = \omega$
    - $y|_i = l \wedge y|_j = (r+1) \mod 5 \wedge y|_{s_{down}} = a \wedge y|_{k_{down}} = p$
    - $z|_i = (l-1) \mod 5 \wedge z|_j = r \wedge z|_{a_{right}} = \omega \wedge z|_{k_{right}} = \omega$
    - $t|_i = (l+1) \mod 5 \wedge t|_j = r \wedge t|_{a_{left}} = \leftarrow \wedge t|_{k_{left}} = p$

then: $BR_u(\{\{w,y,z,t\}\}) = \begin{cases} \{(l,r,a,\omega,a,p,\omega,\omega,\leftharpoondown,p,\omega)\}, & \text{if } (p,t|_{s_{up}},\leftharpoondown) \in \delta(t|_{k_{down}},a) \\ \{\mathcal{K}_1\}, & \text{otherwise} \end{cases}$

/* *the property that follows in combination with lemma A.1 makes all the rest-undesired 4-tuples of neighbors' strategies impossible**/

27. In all other cases: $BR_u(\{\{w,y,z,t\}\}) = \{\mathcal{K}_1\}$

Proof that: $(\langle\mathbf{M}\rangle, \mathbf{t}) \in \mathbf{L} \Leftrightarrow$ **game** $\mathbf{G_{M,\,t}}$ **has a pure Nash Equilibrium**
($\Rightarrow$) Supposing that machine M halts on empty input within $5t-2$ steps, we can construct a Nash Equilibrium of the game $G_{M,\,t}$. In order to do so, we pick an arbitrary row and column of the torus and assign to the player at the intersection (that we'll call *corner player*) the strategy $s_{\llcorner}$. Then we assign to the other players of the row strategies of the form $s^i_{\sqcup}$, so that the player 'right' from the corner player (with respect to an arbitrary orientation of the rows) gets strategy $s^0_{\sqcup}$, the player next to him $s^1_{\sqcup}$ and so forth. Similarly, we assign to the player 'up' from the corner player (again with respect to an arbitrary orientation of the torus) strategy $s^0_{\triangleright}$, to the player next to him strategy $s^1_{\triangleright}$ and so forth. Now since $m = 5t+1$, the row and column we chose contain a $5t \times 5t$ square area of players. These players will get strategies from the set S_1. The mapping of the tiles (strategies) of set S_1 to the players of the square area is essentially the same as in the **NEXP**-hardness proof of the tiling problem (see [6]). The only difference now is that the tiles (strategies) have the extra *coordinates modulo 5 labels* which will be filled as follows. We assign coordinates $(0,0)$ to the unique player with neighbors $s^0_{\triangleright}$ and $s^0_{\sqcup}$, coordinates $(1,0)$ to its 'right' neighbor and coordinates $(0,1)$ to its 'up' neighbor. We continue to assign coordinates modulo 5 in the obvious way. Based on the properties of function BR_u, it's not difficult to prove that the assignment of strategies that we have described is a pure Nash equilibrium of the game $G_{M,\,t}$.

($\Leftarrow$) Suppose now that game $G_{M,\,t}$ has a pure Nash equilibrium. We'll prove that $(\langle M\rangle, t) \in L$. The proof goes through the following lemmas that are not very hard to prove.

Lemma A.1 *In a pure Nash equilibrium no player plays a strategy from the set* $\mathcal{K}$*.*

Lemma A.1 ensures that, in a pure Nash equilibrium, there will be no undesired patterns of strategies on the torus and, therefore, establishes the solution of the first difficulty in our reduction (see section 5.2). To see that, note that while defining the best response function BR_u we gave as best response to every undesired configuration of the neighbors' strategies the strategy $\mathcal{K}_1$.

Lemma A.2 *Imagine a set of players that induces a connected subgraph of the torus. If all the players of the set play strategies from* S_1 *then the coordinates of their strategies must form a 2-dimensional counting modulo 5.*

Since we chose $m = 5t+1$, lemma A.2 immediately implies the following.

Lemma A.3 *There is no pure Nash equilibrium in which players have only strategies from the set* S_1*.*

Lemmata A.1 and A.3 imply that, in a pure Nash equilibrium, there is at least one player that plays a strategy from the set S_2. Actually, something stronger is true as implied by the following lemmata.

Lemma A.4 *In a pure Nash equilibrium there is at least one player that plays the strategy* $s_{\llcorner}$*.*

Lemma A.5 *If two players are diagonal one to the other, then they cannot play both* $s_{\llcorner}$ *in a pure Nash equilibrium.*

Lemma A.6 *Suppose that in a pure Nash equilibrium k players play strategy $s_{\llcorner}$. Then there is a divisor ϕ of k, a set of rows of the torus R, with $|R| = \phi$, and a set of columns of the torus C, with $|C| = k/\phi$, such that:*

1. *only the players that are located in the intersection of a row from set R and a column from set C play strategy $s_{\llcorner}$*
2. *for all $r_1, r_2 \in R$ the distance between r_1, r_2 is a multiple of 5*
3. *for all $c_1, c_2 \in C$ the distance between c_1, c_2 is a multiple of 5*
4. *the players that are located in a row of set R or a column of set C but not at an intersection play strategies from the set $S_2 \setminus \{s_{\llcorner}\}$ and furthermore one of the following is true for these players:*
   - *either all the players of the rows have strategies of the form $s^i_{\sqcup}$ and all the players of the columns have strategies of the form $s^j_{\triangleright}$*
   - *or vice versa*
5. *all the other players of the torus have strategies from the set S_1*

Now let us assign an arbitrary orientation to each dimension and let us label each neighbor of a player as 'up', 'down', 'left' and 'right' with respect to that orientation. We call **ordered tuple of neighbors' strategies** of a player the 4-tuple of strategies that has as its first field the strategy of the 'up' neighbor, second field the strategy of the 'down' neighbor, third field the strategy of the 'left' and fourth field the strategy of the 'right' neighbor. The following is true.

Lemma A.7 *In a pure Nash equilibrium all players that play the strategy $s_{\llcorner}$ have the same ordered tuple of neighbors' strategies.*

By combining lemmas (A.1) through (A.7) it follows that the rows of set R and the columns of set C (see lemma (A.6)) define rectangular areas of players on the torus that play strategies from the set S_1. One such rectangular area might look as in figure 4 or might be a rotation or mirroring of that. Now that we have a visual idea of what is happening on the torus we can state the final lemma.

Lemma A.8 *Suppose a pure Nash equilibrium on the torus and one of the rectangular areas of players that are defined by the rows of set R and the columns of set C (see lemma (A.6). The strategies of the players inside the rectangular area correspond to a computation of the non-deterministic machine M on empty input that halts within so many steps as the "height" of the rectangular area minus 2 (where by height of the rectangular area we mean half of its neighbors that play strategies of the form $s^i_{\triangleright}$).*

In a pure Nash equilibrium, the height of each of the rectangular areas that are defined is at most $5t$ since $m = 5t + 1$. Thus, from lemma (A.8), if there is a pure Nash equilibrium, then there is a computation of the machine M that halts within $5t - 2$ steps. ■

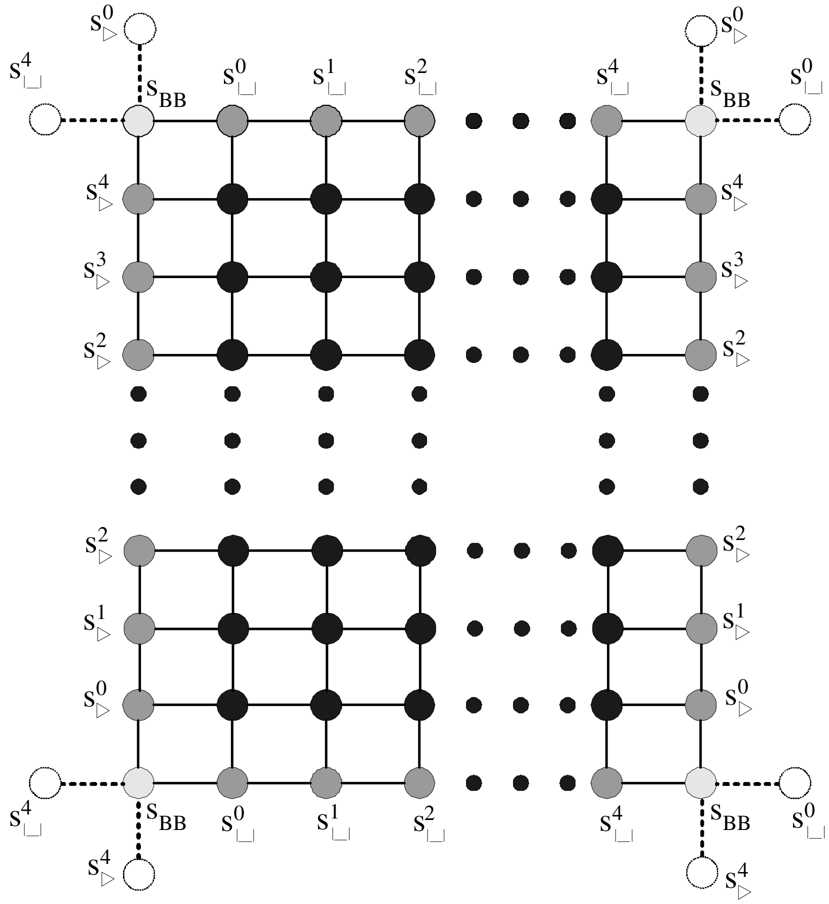

Figure 4: The boundary of every rectangular area of players