Pure Nash Equilibria: Complete Characterization of Hard and Easy Graphical Games

Albert Xin Jiang
Department of Computer Science
University of British Columbia
Vancouver, Canada
jiang@cs.ubc.ca

MohammadAli Safari
Department of Computing Engineering
Sharif University of Technology
Tehran, Iran
msafari@sharif.edu

ABSTRACT

We consider the computational complexity of pure Nash equilibria in graphical games. It is known that the problem is NP-complete in general, but tractable (i.e., in P) for special classes of graphs such as those with bounded treewidth. It is then natural to ask: is it possible to characterize all tractable classes of graphs for this problem? In this work, we provide such a characterization for the case of bounded in-degree graphs, thereby resolving the gap between existing hardness and tractability results. In particular, we analyze the complexity of PURE-GG$(C, -)$, the problem of deciding the existence of pure Nash equilibria in graphical games whose underlying graphs are restricted to class C. We prove that, under reasonable complexity theoretic assumptions, for every recursively enumerable class C of directed graphs with bounded in-degree, PURE-GG$(C, -)$ is in polynomial time if and only if the reduced graphs (the graphs resulting from iterated removal of sinks) of C have bounded treewidth. We also give a characterization for PURE-CHG$(C, -)$, the problem of deciding the existence of pure Nash equilibria in colored hypergraphical games, a game representation that can express the additional structure that some of the players have identical local utility functions. We show that the tractable classes of bounded-arity colored hypergraphical games are precisely those whose reduced graphs have bounded treewidth modulo homomorphic equivalence. Our proofs make novel use of Grohe's characterization of the complexity of homomorphism problems.

Categories and Subject Descriptors

J.4 [Social and Behavioral Sciences]: Economics

General Terms

Theory, Economics, Algorithms

Keywords

game theory, graphical games, computational complexity, homomorphism problem, treewidth

1. INTRODUCTION

There has been recent interest in using game theory to model and analyze large multi-agent systems such as network routing, peer-to-peer fil sharing, auctions and other market mechanisms. One fundamental class of computational problems in game theory is the computation of *solution concepts* of a finit game, such as Nash equilibria. These kinds of computational tasks can be understood in the language of AI as *reasoning* about the game: what are the likely outcomes of the game, under certain models of rationality of the agents? The goal is to be able to efficientl carry out such reasoning for real-world systems.

Much of the existing game theoretic literature models simultaneous action games using the normal form (also known as the strategic form), i.e. a game's payoff function is represented as a matrix with one entry for each player's payoff each combination of the actions of all players. The size of this representation grows exponentially in the number of players. Computations that are "polynomial-time" in the input size are nevertheless impractical. As a result the normal form is unsuitable for representing large systems.

Fortunately, most real-world large games have highly-structured utility functions, which allows them to be represented compactly. A line of research thus exists looking for *compact game representations* that are able to succinctly describe structured games, and efficien algorithms for computing solution concepts that run in time polynomial in the size of the representation. An influentia compact representation of games is *graphical games* proposed by Kearns *et al.* [16]. A graphical game is associated with a graph whose nodes correspond to the players of the game and edges correspond to payoff influenc between players. In other words, each player's payoffs depend only on the actions of himself and his neighbors in the graph. The representation size of a graphical game is exponential in the size of its largest neighborhood. This can be exponentially smaller than the normal form representation of the same game, especially for sparse graphs.

A compact game representation is not very useful if we cannot perform computations that are efficien relative to its size. In this paper we focus on the problem of computing pure-strategy Nash equilibria (PSNE). Unlike mixed-strategy Nash equilibria, which are guaranteed to exist for finit games [18], in general pure Nash equilibria are not guaranteed to exist. Nevertheless, in many ways pure Nash equilibrium is a more attractive solution concept than mixed-strategy Nash equilibrium. First, pure Nash equilibrium can be easier to justify because it does not require the players to randomize. Second, it can be easier to analyze because of its discrete nature (see, e.g., [2]). Gottlob *et al.* [11] were the firs to analyze the problem of computing pure Nash equilibria in graphical games. They proved that the problem is NP-complete in general, even when the graphs have neighborhood size at most 3. On the other hand, for games with graphs of bounded hypertree-width there exists a dynamic-programming algorithm that determines the existence of pure Nash equilibria in polynomial time in the size

Cite as: Pure Nash Equilibria: Complete Characterization of Hard and Easy Graphical Games, Albert Xin Jiang and MohammadAli Safari, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 199-206

of the representation. Daskalakis and Papadimitriou [7] reduced the problem of findin pure strategy Nash equilibrium in graphical games to a Markov Random Field (MRF), and then applied the standard clique tree algorithm to the resulting MRF. Among their results they showed that for graphical games on graphs with log-sized treewidth, bounded neighborhood size and bounded number of actions per player, deciding the existence of pure Nash equilibria is in polynomial time.

A natural question arises: are there other tractable classes of graphical games? Such a tractable class can be define by restrictions over the graph structure as well as the local utility functions. In this paper, we analyze the complexity of PURE-GG$(C, -)$, the problem of determining the existence of pure Nash equilibria in graphical games whose underlying digraphs[1] are restricted to class C (while other aspects of the game representation can be arbitrary). We say C is *tractable* if PURE-GG$(C, -)$ is in polynomial time.

Throughout the paper we make the restriction that the graphical games have bounded neighborhood size (i.e. bounded in-degree). Graphical games with large in-degree have the same problem as normal form games: even if we fin polynomial-time algorithms for them, that would still be impractical due tb the large input size.

Previous results [11, 7] do not answer whether bounded treewidth is the sole measure of tractability of PURE-GG$(C, -)$. For example, it was unknown whether games with log-sized treewidth and unbounded number of actions per player are tractable. Furthermore, there are other graph parameters that affect the tractability of certain computational problems on directed graphs, e.g. directed tree-width [15], D-width [20], DAG-width [19], and Kelly-width [13]. Since these parameters take advantage of the directionality of the edges, they could potentially give a better characterization of the tractability of PURE-GG$(C, -)$.

In this paper we give a complete characterization of the tractable classes of bounded-indegree graphs, thereby resolving the gap between existing tractability and hardness results. Our results are summarized below.

1. Bounded-treewidth graphs are *not* the only tractable kind of digraphs. One example is graphical games on DAGs, for which pure equilibria always exist and can be computed efficientl . More generally, whenever there is a sink (a vertex with out-degree zero), the utilities for that sink player do not affect the existence of pure equilibria.

2. Given a digraph G, let its *reduced graph* be the graph obtained by iterated removal of sinks. We prove that, under reasonable complexity theoretic assumptions, for every recursively enumerable class C of directed graphs with bounded in-degree, PURE-GG$(C, -)$ is in polynomial time if and only if the reduced graphs of C have bounded treewidth.

3. We consider *colored hypergraphical games*, a new game representation based on colored hypergraphs, which can express the additional structure that some of the players have identical local utility functions. For the pure equilibrium problem on this representation PURE-CHG$(C, -)$, we show that a class of colored hypergraphs is tractable if and only if its reduced graphs have bounded treewidth modulo homomorphic equivalence. This is a wider family of tractable games compared to the graphical game representation. That is, by incorporating more information about the structure of the game into the graph, we are able to identify new tractable classes of games.

[1]We defin graphical games on directed graphs (whereas Daskalakis and Papadimitriou's [7] definitio is based on undirected graphs). The definitio with directed graphs is more general, as graphical games on undirected graphs can be thought of as games on directed graphs with bi-directional edges. Our result applies to undirected graphs as a special case.

Our results for PURE-GG$(C, -)$ follow as a corollary to our results for PURE-CHG$(C, -)$. Another corollary is that if the graphical games are represented as undirected graphs, then the tractable classes of undirected graphs are precisely those with bounded treewidth.

We prove these results by connecting PURE-GG$(C, -)$ and PURE-CHG$(C, -)$ to *homomorphism problems*, which given colored hypergraphs G and H, ask whether there exists a homomorphism from G to H. We then make use of Grohe's [12] breakthrough result that characterizes the tractable classes of *HOM*$(\mathcal{C}, -)$, homomorphism problems with restricted left-hand side. This is (as far as we know) a novel application of Grohe's result for homomorphism problems to computational problems in game theory. We prove our main tractability result by reducing an arbitrary instance of PURE-CHG$(C, -)$ to an instance of the homomorphism problem. This reduction has a similar structure as [11]'s formulation of graphical games as constraint satisfaction problems. On the other hand, our proof for our hardness result is quite unlike the existing NP-hardness proof for graphical games [11]. At a high level, this is because the previous approach would construct graphical games on graphs with a certain specifi structure. This is sufficien for proving NP-hardness, but not for our purposes, because we want to characterize the complexity for PURE-CHG$(C, -)$ for arbitrary C, which implies that we had to instead construct our graphical/colored hypergraphical game on an arbitrarily given digraph/colored hypergraph. In other words, we only have control over the utility functions (but not the graph structure), and need to set the utilities such that there is a solution to the given homomorphism problem if and only if the game has a pure equilibrium. This makes our task more technically challenging. We think our proof techniques may have wider interest; for example, it might be possible to extend these techniques to prove similar results for action-graph games [1], another compact game representation.

These results increase our understanding of the power and limitations of the graphical game representation, and have immediate practical impact. Specificall , they imply that if the systems we are interested in have large-treewidth reduced graphs when modeled as graphical games, then the resulting graphical games are unlikely to admit an polynomial-time algorithm for pure Nash equilibria, *even if the graphs have other types of structure*. Nevertheless, if some of the players have identical utility functions, we might be able to get around this limitation of graphical games by representing the systems as colored graphical games instead. If the corresponding reduced graphs have bounded treewidth modulo homomorphism equivalence, pure Nash equilibria can be found efficientl by transforming to the corresponding homomorphism problems which have known polynomial-time algorithms [5, 12].

2. PRELIMINARIES

2.1 Graphical Games

A (simultaneous-move) *game* is a tuple $(N, \{S_i\}_{i \in N}, \{u_i\}_{i \in N})$ where $N = \{1, \ldots, n\}$ is the set of agents; for each agent i, S_i is the set of i's actions. S_i is nonempty. An action profil $\mathbf{s} \in \prod_{i \in N} S_i$ is a tuple of actions of the n agents. $u_i : \prod_{j \in N} S_j \to \mathbb{R}$ is i's utility function, which specifie i's utility given any action profile

For every action profil $\mathbf{s}$, let s_i be the action of agent i under this action profile and s_{-i} be the $(n-1)$-tuple of the actions of agents other than i under this action profile For each action $s'_i \in S_i$, let (s'_i, s_{-i}) be the action profil where agent i plays s'_i and all the other agents play according to s_{-i}.

A game representation is a data structure that stores all information needed to specify a game.

DEFINITION 1. *A* graphical game *representation is a tuple* $(G, \{U_i\}_{i\in N})$ *where*

- $G = (N, E)$ *is a directed graph, with the set of vertices corresponding to the set of agents.* E *is a set of ordered tuples corresponding to the arcs of the graph, i.e.* $(i, j) \in E$ *means there is an arc from* i *to* j*. Vertex* j *is a* neighbor *of* i *if* $(j, i) \in E$.
- *for each* $i \in N$*, a* local utility function $U_i : \prod_{j\in\mathcal{N}(i)} S_j \to \mathbb{R}$ *where* $\mathcal{N}(i) = \{i\} \cup \{j \in N|(j, i) \in E\}$ *is the* neighborhood *of* i.

Each local utility function U_i is represented as a matrix of size $\prod_{j\in\mathcal{N}(i)} |S_j|$. Since the size of the local utility functions dominates the size of the graph G, the total size of the representation is $O(nm^{(\mathcal{I}+1)})$ where $\mathcal{I}$ is the maximum in-degree of G and $m = \max_{j\in N} |S_j|$.

A graphical game $(G, \{U_i\})$ specifie a game $(N, \{S_i\}, \{u_i\})$ where each S_i is specifie by the domain of agent i in U_i, and for all $i \in N$ and all action profil $\mathbf{s}$ we have $u_i(\mathbf{s}) \equiv U_i(s_{\mathcal{N}(i)})$, where $s_{\mathcal{N}(i)} = (s_j)_{j\in\mathcal{N}(i)}$.

2.2 Colored Hypergraphical Games

We now consider graphical games with a certain additional structure. Specificall , some players may have identical local utility functions.[2]

To represent this kind of structure, we not only need to specify which players have the same local utility function, we also need to specify an ordering of the vertices in each neighborhood. We express this kind of structure graphically using colored hypergraphs.

A *colored hypergraph* $H = (V, E, C)$ consists of a set of vertices V, a set of edges E where every edge $e \in E$ is an ordered tuple of vertices in V,[3] and a color function $C : E \to \tau$ that maps each edge to its color. In other words, each edge $e \in E$ is labeled with a color $C(e)$. We denote as $V(H)$, $E(H)$ and C_H the set of vertices, set of edges and the color function of colored hypergraph H, respectively.

We are now ready to defin colored hypergraphical games. Intuitively, in a colored hypergraphical game, the players affecting player i's utility are represented as a colored hyperedge consisting of these players' vertices, with i being the firs element. If two hyperedges have the same color, it means that their corresponding local utility functions are identical.

DEFINITION 2. *A* colored hypergraphical game *is a tuple* $(G, \{U_c\}_{c\in\tau})$*, where*

- $G = (N, E, C)$ *is a colored hypergraph with the set of colors* τ*;*
- *the set of vertices* $V(G) = N$ *corresponds to the set of players;*
- *for each vertex* $v \in N$*, there exists exactly one edge* $e \in E$ *that has* v *as the first element. Denote this edge as* e_v*.*
- *for each color* $c \in \tau$*, edges of color* c *have the same arity*[4] $\mathcal{I}_c$*.*
- *each player has* m *actions. Let* $[m] = \{1, \ldots, m\}$*.*
- *for each color* c*,* $U_c : [m]^{\mathcal{I}_c} \to R$*.*

A colored hypergraphical game $(G, \{U_i\})$ specifie a game $(N, \{S_i\}, \{u_i\})$ where each $S_i = [m]$ and for each $i \in N$ and each action profil $\mathbf{s}$, $u_i(\mathbf{s}) = U_{C(e_i)}(s_{e_i})$.

Unlike graphical games, where given an arbitrary digraph G there is a graphical game on G, not all colored hypergraphs have corresponding colored hypergraph games. Let Σ be the set of colored hypergraphs of colored hypergraph games.

Given $G \in \Sigma$, we defin its *induced digraph* $\mathcal{D}(G)$ to be a digraph on the same set of vertices; and for each hyperedge $(v, v_1, \ldots, v_r)$ in G we create directed edges $(v_1, v), \ldots, (v_r, v)$ in $\mathcal{D}(G)$.

Graphical games can be thought of as special cases of colored hypergraph games where each neighborhood has a different local utility function, i.e. a different color. Given a directed graph $G = (V, E)$, we defin its *induced colored hypergraph* $\mathcal{H}(G) = (V, \mathcal{E}, C)$ such that its set of colors is V and for each vertex $v \in V$, there is a hyperedge $e \in \mathcal{E}$ of color v, consisting of vertices in $\mathcal{N}(i)$, with v being the firs element in the tuple e. The rest of the vertices in e is sorted in a pre-determined order over V. In particular, if the vertices correspond to the agents $1, \ldots, n$ in a game, we require these vertices to be sorted in ascending order of the agents. By construction, $\mathcal{D}(\mathcal{H}(G)) = G$ for all digraph G. Given a graphical game $\Gamma = (G, \{U_i\}_{i\in N})$, its *induced colored hypergraphical game* is $\mathcal{H}(\Gamma) = (\mathcal{H}(G), \{U_i\}_{i\in N})$. It is straightforward to verify that Γ and $\mathcal{H}(\Gamma)$ represent the same game.

For notational convenience, given a class of directed graphs $\mathcal{C}$, let $\mathcal{H}(\mathcal{C})$ be the class of induced hypergraphs of the directed graphs in $\mathcal{C}$.

There is one graph often associated with any hypergraph: the *primal* graph. Given a colored hypergraph H, its primal graph *pri*(H) is an undirected graph obtained by making a clique out of the vertices in every edge in H.

There are a couple of previous papers on the computational properties of graphical games with different notions of identical utility functions. Daskalakis *et al.* [6] analyzed the complexity of find ing pure and mixed Nash equilibria of graphical games on highly regular graphs (namely the d-dimensional grid), in which all local payoff functions are identical. Brandt *et al.* [3] instead analyzed graphical games on arbitrary graphs, but with several stronger notions of symmetry. In contrast, the colored hypergraphical game formulation places the least amount of restrictions and is thus more likely to occur in practice. In fact, these previous formulations can be thought of as special cases of colored hypergraphical games.

2.3 Best Response and Pure Nash Equilibrium

Given a game $(N, \{S_i\}_{i\in N}, \{u_i\}_{i\in N})$ and s_{-i}, agent i's *best response* to s_{-i} is $BR_i(s_{-i}) = \arg\max_{s_i\in S_i} u_i(s_i, s_{-i})$. Since S_i is nonempty, i has at least one best response given any s_{-i}. Note that in a graphical game, the best response of i depends only on the actions of i's neighbors.

[2]For simplicity of presentation, we assume each player have the same number of actions. We can convert an arbitrary game to our setting by adding dummy actions. Since we are only focusing on graphical games with bounded in-degree, this would only increase the representation size by a polynomial factor.

[3]Note that the definitio we use is slightly different from the common definitio of hypergraphs in which each edge is an unordered set of vertices.

[4]The *arity* of an edge e is its size, i.e. number of elements.

DEFINITION 3. *An action profile* $\mathbf{s} \in \prod_{i \in N} S_i$ *is a* pure Nash equilibrium *of the game* $(N, \{S_i\}_{i \in N}, \{u_i\}_{i \in N})$ *if each agent* $i \in N$ *is playing a best response to* s_{-i}*, i.e.* $s_i \in BR_i(s_{-i})$.

We defin PURE-GG to be the following decision problem: given a graphical game $(G, \{U_i\}_{i \in N})$, decide whether there exists a pure Nash equilibrium. Gottlob *et al.* [11] have shown that this problem is NP-complete in general. Given a class C of digraphs and class $\mathcal{U}$ of local utility functions, let PURE-GG(C,$\mathcal{U}$) be the pure Nash equilibrium decision problem on graphical games when the graphs of the input game are taken only from class C and the local utility functions are taken only from class $\mathcal{U}$. In this paper we are interested in problems of the form PURE-GG($C, -$), which means that the local utility functions are unconstrained, other than the requirement that the input is a well-formed graphical game, i.e. that each U_i takes $|\mathcal{N}(i)|$ arguments.

Similarly, we defin the problem PURE-CHG($C, -$) to be the pure Nash equilibrium decision problem on colored hypergraphical games, with colored hypergraphs restricted to class C.

2.4 Treewidth

Due to space constraints we omit the standard definitio of treewidth for undirected graphs (see, e.g., [17]). The treewidth of a digraph G is the treewidth of the undirected version of G. The treewidth of a colored hypergraph is the treewidth of its primal graph.

2.5 Homomorphism

Let G and H be two colored hypergraphs. A *homomorphism* from G to H is a mapping h from the vertex set of G to the vertex set of H that preserves both adjacency and color, i.e. for every edge $e = (a_1, a_2, \cdots, a_k) \in E_G, h(e) = (h(a_1), h(a_2), \cdots, h(a_k)) \in E_H$ and $C_G(e) = C_H(h(e))$. In a *homomorphism problem*, we are given G and H and have to decide whether there exists a homomorphism from G to H.

For two classes $\mathcal{C}$ and $\mathcal{D}$ of colored hypergraphs let *HOM*($\mathcal{C}, \mathcal{D}$) be the homomorphism problem when the input colored hypergraphs are taken only from classes $\mathcal{C}$ and $\mathcal{D}$. When an input class is the class of all colored hypergraphs, we use the notation '$-$' instead.

Two hypergraphs G and H are *homomorphically equivalent* if there is a homomorphism from G to H and vice versa. A class C has bounded treewidth modulo homomorphic equivalence if there exists some constant k such that every hypergraph in C is homomorphically equivalent to a hypergraph with treewidth at most k. For example the class of bipartite graphs have bounded treewidth modulo homomorphic equivalence as they are homomorphically equivalent to an edge. We use *modulo-treewidth*(G) to indicate the minimum k for which G is homomorphically equivalent to a hypergraph of treewidth k.

EXAMPLE 4. *We describe a class of colored hypergraphical games with bounded treewidth modulo homomorphism equivalence. Each game has* m^2+2m *players. There are 4 colors* $\{L, R, X, Y\}$*. We have* m *players labeled* $l_1 \ldots l_m$*, and* m *players labeled* $r_1, \ldots, r_m$*. For each* $i, j \in \{1, \ldots, m\}$ *we have a player* x_{ij} *and a player* y_{ij}*. For each player* x_{ij}*, we have an hyperedge* $(x_{ij}, y_{ij}, l_i, r_j)$ *of color* X*; for each player* y_{ij}*, we have an hyperedge* $(y_{ij}, x_{ij}, l_i, r_j)$ *of color* Y*. Furthermore for each player* l_i *we have a hyperedge* (l_i) *of color* L *and for each player* r_i *we have a hyperedge* (r_i) *of color* R*. The colored hypergraph is homomrophically equivalent to the fragment involving only the vertices* l_1, r_1, x_{11}, y_{11} *and their corresponding hyperedges. Therefore these colored hypergraphical games have modulo-treewidth 3, while the treewidth of each hypergraph is at least* m.

2.6 Parameterized Complexity Theory

Our results make use of certain concepts from he theory of parameterized complexity developed by Downey and Fellows [9]. They are not essential for understanding our reductions. We briefl mention the relevant concepts here and refer the reader to [9, 10] for more details.

Given a decision problem $P \subseteq \Sigma^*$, a parameterization of P is a mapping $k : \Sigma^* \rightarrow N$ that define the paramterized problem (P, k). A paramterized problem (P, k) is *fixed parameter tractable* if there is a computable function $f : N \rightarrow N$ and an algorithm that decides if a given instance $x \in \Sigma^*$ belongs to P in time $f(k(x))|x|^{O(1)}$ for some function f depending only on k. The class of all fi ed parameter tractable problems is denoted by FPT.

Downey and Fellows [8] define the paramterized complexity class $W[1]$, which can be seen as an analogue of NP in parametrized complexity theory, and conjectured that FPT is a proper subset of W[1]. This conjecture is widely believed to be true.

Let *p-HOM*($C, -$) be the parametrized version of *HOM*($C, -$), with the parameter being the representation size of the left colored hypergraph. Similarly we defin p-PURE-GG($C, -$) and p-PURE-CHG($C, -$) to be the parametrized versions of PURE-GG($C, -$) and PURE-CHG($C, -$), with the parameters being the representation sizes of the directed graph and the colored hypergraph, respectively.

2.7 Complexity of Homomorphism Problems

Grohe [12], in a breakthrough paper, characterizes the tractable instances of the homomorphism problem when we restrict the left input graphs.[5]

THEOREM 5 (GROHE [12]). *Assume FPT* $\neq$ *W[1]. Then for every recursively enumerable class* C *of colored hypergraphs with bounded arity the following statements are equivalent.*

1. *HOM*($C, -$) *is in polynomial time.*
2. *p-HOM*($C, -$) *is fixed-parameter tractable.*
3. C *has bounded modulo-treewidth.*

Under a slightly stronger assumption of *nonuniform-FPT* $\neq$ *nonuniform-W[1]*, this result holds for arbitrary (not necessarily recursively enumerable) class C.

3. MAIN RESULT

3.1 Digraphs with sinks

One's firs intuition is to try and show a correspondence between PURE-GG($C, -$) and $HOM(\mathcal{H}(C), -)$ for arbitrary classes of bounded-degree graphs. In fact such correspondence does not exist for arbitrary graphs. For example, a graphical game on a directed acyclic graph (DAG) always has a PSNE, which can be computed efficientl by a greedy algorithm that goes through vertices in the topological order, from sources to sinks. Consider the class D_k of k-bounded in-degree DAGs. $\mathcal{H}(D_k)$ has unbounded modulo-treewidth, however PURE-GG($D_k, -$) is in polynomial time. This is just an example of a more general phenomenon in graphical games. Let $GG = (G, \{U_i\}_{i \in N})$ be a graphical game. If G has a sink u (i.e. u has out-degree zero) then the action of u does not affect any other player. This means we can simply solve the game without player u and the resulting game has a pure Nash equilibrium if and only if GG has one.

[5]Grohe stated his result on *relational structures* instead of colored hypergraphs. The two formulations are equivalent.

Intuitively, this is because in a graphical game (and any game in general) each player has at least one action, and as a result, whatever actions others chose, each player has at least one best response.

We formalize this intuition as the following classificatio of digraphs into reducible and irreducible graphs:

DEFINITION 6. *A directed graph G is* irreducible *if it does not have a sink (a vertex with out-degree zero). Otherwise G is* reducible*.*

It is often helpful to consider the strongly connected components (SCCs) of a directed graph. In particular, we can characterize irreducible graphs by their *terminal SCCs*.

DEFINITION 7. *A strongly connected component(SCC) π of G is* terminal *if there is no outgoing edges from π. A terminal SCC is by definition a maximal SCC.*

LEMMA 8. *If G is irreducible then all its terminal SCCs have size at least 2.*

This is because otherwise, a vertex in a terminal SCC with only one vertex would have out-degree zero.

It turns out that for our purposes, given an arbitrary digraph we can focus on its subgraph resulting from iterative removal of sinks.

DEFINITION 9. *Given a directed graph G, its reduced graph red(G) is the result of the following algorithm:*

1. *repeat until G does not change:*
   (a) *remove all vertices with out-degree zero as well as their incoming edges.*
2. *return G*

DEFINITION 10. *Given a graphical game $GG = (G, \{U_i\}_{i\in N})$, its* reduced game *red$(GG)$ is $(red(G), \{U_i\}_{i\in V(red(G))})$, i.e. the game obtained by removing all agents corresponding to reducible vertices of G.*

red(GG) is well define because for all $v \in$ *red*(G), vertices that are neighbors of i in G are not reducible vertices, so they are still present in *red*(G).

LEMMA 11. *A graphical game GG has a pure Nash equilibrium if and only if its reduced game red(GG) has a pure Nash equilibrium.*

LEMMA 12. *Suppose $\mathcal{C}$ is a recursively enumerable class of graphs with bounded in-degree, such that PURE-GG(red$(\mathcal{C}), -)$ is in P. Then PURE-GG$(\mathcal{C}, -)$ is in P.*

For the other direction, we would like to prove that if graphical games on a class of graphs $\mathcal{C}$ is tractable, then graphical games on *red*$(\mathcal{C})$ is also tractable. This is not trivial, because although the reducible vertices of graphs in $\mathcal{C}$ do not affect the existence of pure equilibria, the subgraphs on these vertices could potentially carry information (similar to advice strings in complexity theory) such that PURE-GG$(\mathcal{C}, -)$ is easier than PURE-GG(*red*$(\mathcal{C}), -)$. It turns out that if we consider the parameterized version of the problem, then if p-PURE-GG$(\mathcal{C}, -)$ is in FPT then p-PURE-GG(*red*$(\mathcal{C}), -)$ is in FPT. This will be sufficien for our purposes. The proof of the following lemma is given in Appendix A.

LEMMA 13. *Suppose $\mathcal{C}$ is a recursively enumerable class of graphs with bounded in-degree, such that p-PURE-GG$(\mathcal{C}, -)$ is in FPT. Then p-PURE-GG(red$(\mathcal{C}), -)$ is in FPT.*

We can defin analogous concepts for colored hypergraphs and colored hypergraphical games, by looking at their induced digraphs. A colored hypergraph $G \in \Sigma$ is irreducible if and only if its induced digraph is irreducible. Given $G \in \Sigma$, its reduced colored hypergraph *red*(G) is obtained by removing all reducible vertices of the induced digraph of G and all hyperedges that include these reducible vertices. Reduced colored hypergraphical games can be define similarly. Lemmas 12 and 13 can be straightforwardly extended to colored hypergraphical games.

3.2 Main Theorems

The above implies that for the complexity of PURE-GG$(C, -)$ and PURE-CHG$(C, -)$, it is sufficien to consider irreducible graphs. The complexity for a general class C then correspond to the complexity for *red*(C). It turns out that if we restrict to irreducible graphs, there exists a correspondence between PURE-CHG$(C, -)$ and $HOM(\mathcal{H}(C), -)$. We are now ready to state our main result, which will be proved in the rest of Section 3:

THEOREM 14. *Assume FPT $\neq$ W[1]. Then for every recursively enumerable class of bounded arity colored hypergraphs $\mathcal{C} \subseteq \Sigma$, the following statements are equivalent.*

1. *PURE-CHG$(C, -)$ is in polynomial time.*
2. *p-PURE-CHG$(C, -)$ is fixed-parameter tractable.*
3. *for every $G \in \mathcal{C}$, red(G) has bounded modulo-treewidth.*

The direction $1 \rightarrow 2$ is trivial; the "tractability" direction $3 \rightarrow 1$ is proved in Section 3.3; the "hardness" direction $2 \rightarrow 3$ is proved in Sections 3.4 and 3.5.

We then obtain as a corollary the characterization for the complexity of PURE-GG$(C, -)$. We make use of the following lemma on the modulo-treewidth of $\mathcal{H}(G)$.

LEMMA 15. *Given a digraph G, the modulo-treewidth of $\mathcal{H}(G)$ equals the treewidth of $\mathcal{H}(G)$.*

Furthermore, we can relate the treewidth of a digraph G to the treewidth of $\mathcal{H}(G)$. Daskalakis and Papadimitriou [7] showed that given an undirected graph H with bounded degree, the treewidth of its induced hypergraph $\mathcal{H}(H)$ and the treewidth of H are within a constant factor of each other. This result cannot be directly applied to digraphs, because the induced hypergraph of the undirected version of a digraph G can be different from $\mathcal{H}(G)$. Nevertheless, their proof can be relatively straightforwardly adapted to digraphs, yielding the following lemma.

LEMMA 16. *Given a digraph G with bounded in-degree, the treewidth of $\mathcal{H}(G)$ and the treewidth of the undirected version of G are within a constant factor of each other.*

This means for our purposes bounded treewidth of $\mathcal{H}(G)$ implies bounded treewidth of G and vice versa. We are now ready to state the characterization for PURE-GG$(C, -)$, which is a direct consequence of Theorem 14 and Lemmas 15 and 16.

COROLLARY 17. *Assume FPT $\neq$ W[1]. Then for every recursively enumerable class C of digraphs with bounded in-degree the following statements are equivalent.*

1. *PURE-GG$(C, -)$ is in polynomial time.*
2. *p-PURE-GG$(C, -)$ is fixed-parameter tractable.*
3. *for every $G \in C$, red(G) has bounded treewidth.*

Comparing Theorem 14 and Corollary 17, CHGs gives a wider family of tractable games compared to graphical games. For example, the class of CHGs described in Example 4 has bounded modulo-treewidth but unbounded treewidth. Thus they would be intractable if represented as graphical games.

We also obtain as a corollary the characterization for the pure equilibrium problem for graphical games define on undirected graphs. Defin PURE-UGG$(C, -)$ to be the problem of deciding existence of pure equilibrium on such undirected graphical games, restricted to the class of graphs C. Then under the same assumptions, C is tractable if and only if its graphs have bounded treewidth.

3.3 Proof of Tractability Result

We use the following lemma that reduces a colored hypergraphical game to a homomorphism problem instance. The tractability direction of Theorem 14 then follows straightforwardly.

LEMMA 18. *Let $\Gamma = (G, \{U_i\}_{i\in N})$ be a colored hypergraphical game. It is possible to construct in polynomial time an instance (G', H') of homomorphism problem such that Γ has a pure equilibrium if and only if there exists a homomorphism from G' to H'. Furthermore if G has bounded arity and bounded modulo-treewidth then so does G'.*

PROOF. Given a colored hypergraphical game $\Gamma = (G, \{U_c\}_{c\in\tau})$, each player having m actions, we construct the instance (G', H') of the homomorphism problem as follows. Let $G' = G$. H' consists of m vertices, one for each action in $[m]$. For each color c, for each action tuple $(a, a_1, a_2, \cdots, a_r)$ such that

$$a \in \arg\max_{a'\in[m]} U_c(a', a_1, a_2, \cdots, a_r)$$

(i.e. a is a best response for a player with utility function U_c given neighbor actions $(a_1, a_2, \cdots, a_r)$), create an hyperedge $(a, a_1, a_2, \cdots, a_r)$ of H' with color c.

If Γ has a pure Nash equilibrium then the mapping that maps each vertex u to the vertex a, where a is the action chosen by u in the pure Nash equilibrium, is a homomorphism. For the other direction, if H' is a homomorphism of G' and the corresponding mapping function is ℓ then $\ell(u)$ corresponds to an action of u, and for every edge $e^u = (u, u_1, u_2, \cdots, u_r)$ of color c in G', $\ell(e^u) = (\ell(u), \ell(u_1), \ell(u_2), \cdots, \ell(u_r))$ must be an edge of color c in H'. This implies that $\ell(u)$ is a best response action of player u against his neighbors' actions. Therefore, the mapping ℓ corresponds to a pure Nash equilibrium.

Since G' is identical to G, both maximum arity and modulo-treewidth remain unchanged. ☐

3.4 Hardness for graphical games

We firs consider the hardness result for graphical games. In Section 3.5 we extend our approach to colored hypergraph games.

As mentioned in the introduction, applying the hardness proof approach of [11] to our setting would create graphical games with a particular structure, which is not sufficien for our purpose because we want to characterized the complexity of PURE-GG$(C, -)$ given an *arbitrary* class C.We thus use a different construction in our proof of the hardness direction, which starts with an arbitrary class C of irreducible digraphs, constructs a bijective mapping to a class C' of colored hypergraphs, and then show that for any instance (G, H) of $HOM(C', -)$ we can construct an equivalent instance of PURE-GG$(C, -)$. We can then apply Theorem 5 to get the hardness result.

The key step of the proof is the following lemma. Recall that given digraph G, $\mathcal{H}(G)$ is the colored hypergraph with hyperedge e_i (the edge that corresponds to vertex i and its neighbors) being colored with color i.

LEMMA 19. *Let G be an irreducible digraph. Then for any colored hypergraph H, there exists a graphical game $GG = (G, \{U_i\}_{i\in N})$ such that there is a homomorphism from $\mathcal{H}(G)$ to H if and only if GG has a PSNE.*

The reduction is outlined as follows. (We give a detailed proof of the lemma in Appendix B.) Each player's action set consists of $V(H)$ plus some "failure actions", in this case T and B. We defin the utility for i, given a local strategy profil over $\mathcal{N}(i)$, such that if the local strategy profil correspond to a hyperedge in H of color i, then i gets a high payoff (say 100), such that if there exists a homomorphism from $\mathcal{H}(G)$ to H, then the corresponding strategy profil is a PSNE.

If the local strategy profil does not correspond to an edge of right color in H, we set the utilities so that player i is forced to play one of the failure actions. This implies that out-neighbors of i are forced to play failure actions, and so on. Now we just need to set utilities such that if at least one player plays failure actions, then no PSNE exists. Recall that if G was a DAG, then there always exists a PSNE; i.e. a game construction with no PSNE must contain a cycle. Fortunately G is assumed to be irreducible, which means that all of its terminal SCCs has a directed cycle of length at least 2. For each of the terminal SCCs, fi a cycle and set the utilities of players on that cycle (given failure actions of their neighbors) to be a generalization of the Matching Pennies game: one of the players is incentivized to play the opposite failure action as his predecessor in the cycle, while all other players on the cycle are incentivized to imitate their predecessors.

If there is no homomorphism, then for any strategy profile there must be one player forced to play failure actions, which implies that at least one terminal SCC play failure actions, which implies that one of these cycles are playing the generalized Matching Pennies game, which does not have a PSNE.

Using Lemma 19, given an FPT algorithm for p-PURE-GG$(C, -)$ we can construct an FPT algorithm for *p-HOM*$(C', -)$ where $C' = \{\mathcal{H}(G) | G \in$ *red*$(C)\}$. This implies the hardness direction for graphical games.

3.5 Hardness for colored hypergraphical games

To prove the hardness direction of Theorem 14, it is sufficien to extend Lemma 19 to colored hypergraphical games:

LEMMA 20. *Let $G \in \Sigma$ be an irreducible colored hypergraph. Then for any colored hypergraph H, there exists a colored hypergraphical game $\Gamma = (G, \{U_c\}_{c\in\tau})$ such that there is a homomorphism from G to H if and only if Γ has a PSNE.*

We sketch a proof of the lemma in this section. At a high level, the main difficult when extending our proof of Lemma 19 to colored hypergraphical games is that players with the same color must have the same utility function. Instead of being able to specify the utility function for each player in the graphical game case, now we need to defin one utility function U_c for each color c. In fact, our generalized Matching Pennies construction for the graphical game case cannot be directly applied to colored hypergraphical game, and our proof of Lemma 20 instead uses a different construction involving $2n + 1$ failure actions for each player.

Part of the hardness proof for graphical games can be relatively easily adapted to colored hypergrahical games: each player's action set still consists of $V(H)$ plus some failure actions (to be specifie later). We set the utility function U_c so that if the input action tuple corresponds to a hyperedge of color c in H, then the utility is

100. This will ensure that if there exists a homomorphism, then the corresponding strategy profil is a PSNE. This concludes the proof of the "if" direction of Lemma 20.

The "only if" direction is more difficult In particular, it is difficul to defin the utilities for the failure actions in a way that respects the color constraints. For one, we would not be able to express the generalized Matching Pennies game now: in the worst case all players may have the same color. Also, we cannot specify a cycle and then defin utility functions on that cycle in a way that ignore all edges not in the cycle: this would also require player-specifi utility functions.

Thus we want the utilities given failure actions to not depend on the player. For the simple case of a single cycle, the following construction is sufficient (For notational convenience, we only specify the best response function BR, which maps a tuple of actions of the neighbors to a single action as the best response. Given the BR function the utilities can be define straightforwardly.)

LEMMA 21. *Given a colored hypergraph G, whose induced digraph consists of just one cycle with length n, the following colored hypergraphical game on G does not have PSNE:*

- *each player's actions are the integers* $0, \dots, p-1$*;*
- *let* $BR(a) = (a+1) \mod p$ *where* $p \geq n+1$.

We omit the straightforward proof. If we think of BR as arcs from a to $BR(a)$, then the digraph on actions form a p-cycle.

This can be extended to strongly connected digraphs, by the the following construction:

LEMMA 22. *Given a colored hypergraph G, whose induced digraph is strongly connected, the following colored hypergraphical game on G does not have PSNE:*

- *each player's actions are the integers* $0, \dots, n$*;*
- *given neighbors' actions* $(s_1, \dots, s_m)$*, let* $BR(s_1, \dots, s_m) = (\max\{s_1, \dots, s_m\} + 1) \mod (n+1)$.

The intuition is that for each strategy profile at least one neighbor is "activated" in the following sense: Given digraph $G = (V, E)$, strat profil $\mathbf{s}$, we say an edge $(u, v) \in E$ is active if

$$u \in \arg \max_{u':(u,v) \in E} s_{u'},$$

i.e. u's action under $\mathbf{s}$ is maximal among v's neighbors. Let G' be the subgraph of G where we only keep the active edges, i.e. for each player i, only keep the edge from the neighbor playing the highest action among neighbors. We claim that G' must contain a cycle, i.e. is not a DAG. This is because G is strongly connected, which means it has no source, i.e. all vertices of G have positive number of incoming edges. This implies that all vertices of G' have positive number of incoming edges, i.e. G' has no source. Therefore G' is not a DAG.

Since G' must contain a cycle, on that cycle $BR(a) = (a+1) \mod (n+1)$, which implies that at least one player on that cycle is not playing a best response. Therefore $\mathbf{s}$ must not be a PSNE.

The above construction does not directly work for the general case of digraphs with no sinks: now G' could be a DAG. It turns out that we can indeed fi the construction to work for all digraphs with no sinks. At a high level, instead of forming a best-response cycle with the actions, we form a ρ shape with a cycle and a tail.

We now complete the specificatio of the utility functions for our construction for Lemma 20. The failure actions are $1, \dots, 2n+1$. If the input action tuple of U_c does not correspond to a hyperedge in H with the same color c, then:

- if no neighbors are playing failure actions, then utility of playing failure action 1 is +1, all others -100;
- otherwise, let $f_{\max}$ be the max failure action of neighbors. If $f_{\max} < 2n+1$, then $BR = f_{\max} + 1$; otherwise (i.e. $f_{\max} = 2n+1$), let $BR = n+1$.

We now sketch the "only if" direction of Lemma 20. If there is no homomorphism, then for any strategy profile some player must play failure actions, which implies that some SCC (and all SCCs reachable form there) must play failure; the other SCCs must not play failure actions. Given a strategy profile consider the "earliest reached" non-singleton SCC, as define by the following process: go through SCCs in topological order, in the direction of the edges; return the firs non-singleton SCC whose players choose failure actions. All earlier SCCs are either not playing failure actions, or a singleton SCC that is playing failure action $a <= n$.

Given strategy profil $\mathbf{s}$, consider the graph G_f, which is the subgraph of G restricted over the earliest non-singleton SCC and all earlier singleton SCCs playing failure actions:

First, we claim that if one player j in the non-singleton SCC is playing an action less or equal to n, then $\mathbf{s}$ must not be a PSNE. This is because for such an action $b \leq n$ to be a best response, all incoming neighbors within the SCC must be playing even lesser failure actions. If we iteratively follow an incoming neighbor within the SCC, thus with decreasing actions, we either encounter j again, with action less than b, a contradiction, or a player k playing action 1. For 1 to be a BR, all neighbors must not play failure actions; but k is in a non-singleton SCC with all players playing failure actions, so there must be at least one neighbor playing failure actions, a contradiction.

Therefore, in order for $\mathbf{s}$ to be an PSNE, all players in the non-singleton SCC must play failure actions greater than n. We claim that if this is the case, then each vertex in the non-trivial SCC must have an active neighbor in the same SCC. This is because if an edge from a singleton SCC in G_f to a player i the non-singleton SCC in G_f is active, then because the player in the singleton SCC is playing some action $a <= n$, the target player i in the non-singleton SCC must have an inactive neighbor in that SCC, i.e. some player j in the non-singleton SCC is playing $b < a \leq n$. We have argued above that this would contradict with $\mathbf{s}$ being a PSNE. Therefore, each vertex in the non-trivial SCC must have an active neighbor in the same SCC. By the same argument as for the strongly connected digraph case, there must exist an active cycle in the SCC. Since every player on that active cycle is playing an action greater than n, they are playing a shifted version of the game in Lemma 22. This implies that $\mathbf{s}$ cannot be a PSNE.

4. DISCUSSION AND FUTURE WORK

Our results can be understood as establishing an equivalence between PSNE problems and homomorphism problems. Such a equivalence relation is closer than the kind of equivalence between two NP-complete problems: we are in fact showing a family of equivalences, between PURE-CHG$(C, -)$ for an arbitrary class C and *HOM*(*red*$(C), -)$. On the other hand, our results also show that there are certain differences between the two problems: because in a graphical/colored hypergraphical game each player has at least one best response regardless of her neighbors' actions, we can iteratively remove sinks without affecting the answer, whereas the same does not hold for homomorphism problems in general.

We have focused on the decision problem on the existence of pure Nash equilibria. Related problems include counting the number of pure Nash equilibria and findin one such equilibrium if one exists. On the homomorphism problem side, Dalmau and Jonsson

[4] gave a characterization of the complexity of the counting version of *HOM*$(C, -)$, while the characterization for the construction problem is still open. An interesting direction is to adapt our reductions to the counting and construction versions of these problems, as well as to cases with unbounded in-degree. Another direction is to use similar techniques to prove characterizations for other game representations such as action graph games [1, 14].

5. REFERENCES

[1] N. Bhat and K. Leyton-Brown. Computing Nash equilibria of action-graph games. In *UAI*, 2004.

[2] F. Brandt, F. Fischer, and M. Holzer. Symmetries and the complexity of pure Nash equilibrium. In *STACS*, 2007.

[3] F. Brandt, F. A. Fischer, and M. Holzer. Equilibria of graphical games with symmetries. In *WINE*, pages 198–209, 2008.

[4] V. Dalmau and P. Jonsson. The complexity of counting homomorphisms seen from the other side. *Theor. Comput. Sci.*, 329(1-3):315–323, 2004.

[5] V. Dalmau, P. G. Kolaitis, and M. Y. Vardi. Constraint satisfaction, bounded treewidth, and finite- ariable logics. In *CP*, pages 310–326, 2002.

[6] C. Daskalakis and C. Papadimitriou. The complexity of games on highly regular graphs. In *the 13th Annual European Symposium on Algorithms*, 2005.

[7] C. Daskalakis and C. Papadimitriou. Computing pure Nash equilibria via Markov random fields In *ACM-EC*, 2006.

[8] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness i: Basic results. *SIAM J. Comput.*, 24(4):873–921, 1995.

[9] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[10] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[11] G. Gottlob, G. Greco, and F. Scarcello. Pure Nash equilibria: Hard and easy games. In *TARK*, 2003.

[12] M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1):1–24, 2007.

[13] P. Hunter and S. Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theoretical Computer Science*, 399(3):206–219, 2008.

[14] A. X. Jiang and K. Leyton-Brown. A polynomial-time algorithm for Action-Graph Games. In *AAAI*, 2006.

[15] T. Johnson, N. Robertson, P. Seymour, and R. Thomas. Directed tree-width. *Journal of Combinatorial Theory, Series B*, 82(1):138–154, 2001.

[16] M. Kearns, M. Littman, and S. Singh. Graphical models for game theory. In *UAI*, 2001.

[17] T. Kloks. *Treewidth: Computations and Approximations*. Springer-Verlag, Berlin, 1994.

[18] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.

[19] J. Obdržálek. DAG-width: connectivity measure for directed graphs. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, page 821. ACM, 2006.

[20] M. Safari. D-Width: A more natural measure for directed tree width. In *MFCS*, pages 745–756. Springer Verlag, 2005.

APPENDIX

A. PROOF OF LEMMA 13

PROOF. Given class $\mathcal{C}$ with in-degree bounded by $\mathcal{I}$, and an FPT algorithm for p-PURE-GG$(\mathcal{C}, -)$, we now construct a fi ed-parameter tractable algorithm for p-PURE-GG(*red*$(\mathcal{C}), -)$. Given $G' \in$ *red*$(\mathcal{C})$, we run the following algorithm:

1. Enumerate the class $\mathcal{C}$ until we fin a graph $G \in \mathcal{C}$ such that $G' =$ *red*(G).
2. Run algorithm for p-PURE-GG$(\mathcal{C}, -)$ on G.

We claim that Step 1 always terminates and its running time is bounded by a computable function on the size of G'. This is because the class C is recursively enumerable, and because by definition, for each $G' \in$ *red*$(\mathcal{C})$, there exists a graph $G \in \mathcal{C}$ such that $G' =$ *red*(G). Therefore we have a fi ed-parameter tractable algorithm for p-PURE-GG(*red*$(\mathcal{C}), -)$. □

B. PROOF OF LEMMA 19

PROOF. Let G's terminal SCCs be $\pi_1, \ldots, \pi_m$. For each π_j, fi a cycle $i_0^j \to \ldots \to i_{r_j}^j \to i_0^j$. This is always possible since G is irreducible and, hence, every terminal SCC has at least two vertices. These cycles are disjoint since the terminal SCCs are maximal SCCs.

The graphical game GG is constructed as follows.

- Each player i's action set is $V(H) \cup T \cup B$. T and B are the "failure actions".
- Player i's utility: by the definitio of graphical game, his utility depends on the actions chosen by him and his neighbors. Let $p_i \in \prod_{j \in \mathcal{N}(i)} S_j$ be the tuple of actions chosen by i and its neighbors.
  1. If p_i corresponds to a hyperedge in H with the same color as e_i (the edge corresponding to $\mathcal{N}(i)$), then i's utility is 100.
  2. Otherwise:
     (a) If i is not playing one of the failure actions T or B, then i gets -100.
     (b) Otherwise, if $i = i_k^j$ in one of the pre-define cycles:
        i. If $k > 0$, then $i = i_k^j$'s payoff depends only on the actions of herself and i_{k-1}^j. If i_{k-1}^j plays other than T or B, i gets 0 by playing either T or B. Otherwise, i gets 1 if both she and i_{k-1}^j plays T or both plays B, and -1 otherwise.
        ii. If $k = 0$, then $i = i_0^j$'s payoff depends only on the actions of herself and $i_{r_j}^j$. If $i_{r_j}^j$ plays other than T or B, i gets 0 by playing either T or B. Otherwise, i gets -1 if both she and $i_{r_j}^j$ plays T or both plays B, and 1 otherwise.
     (c) Otherwise, i gets 0 (by playing either T or B).

We claim that this graphical game has a PSNE if and only if there is a homomorphism from $\mathcal{H}G$ to H.

if part: if there exists a homomorphism from $\mathcal{H}G$ to H whose mapping function is h, then in the graphical game, the strategy in which each player i plays $h(i)$ is a Nash equilibrium.

only if part: A PSNE where everyone gets 100 corresponds to a homomorphism. Furthermore, the only PSNE of the graphical game are ones where every player gets 100. This is because if some player i fails to get 100, then he has to play T or B to avoid the -100 penalty. This makes all his (outgoing) neighbors fail to get 100 as well, so they also have to play T or B. i is either part of a terminal π_j or there is a path to a vertex in a terminal π_j. Since i plays failure actions all players in π_j must play failure actions as well. Therefore the pre-define cycle $i_0^j \to \ldots \to i_{r_j}^j \to i_0^j$ play failure actions. The utilities are set up so that the players in this cycle are playing a game similar to Matching Pennies, and it is straightforward to verify that there is no PSNE if they play the failure actions. Therefore there's no PSNE unless everyone gets 100. □