Nash Equilibria in Symmetric Games with Partial Observation

Patricia Bouyer, Nicolas Markey, Steen Vester

February 2014

Research report LSV-14-01 (Version 1)

Laboratoire Spécification & Vérification

École Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Nash Equilibria in Symmetric Games with Partial Observation

Patricia Bouyer Nicolas Markey
LSV, CNRS & ENS Cachan, France
{bouyer,markey}@lsv.ens-cachan.fr

Steen Vester
DTU, Kgs. Lyngby, Denmark
stve@dtu.dk

We investigate a model for representing large multiplayer games, which satisfy strong symmetry properties. This model is made of multiple copies of an arena; each player plays in his own arena, and can partially observe what the other players do. Therefore, this game has partial information and symmetry constraints, which make the computation of Nash equilibria difficult. We show several undecidability results, and for bounded-memory strategies, we precisely characterize the complexity of computing pure Nash equilibria (for qualitative objectives) in this game model.

1 Introduction

Multiplayer games. Games played on graphs have been intensively used in computer science as a tool to reason about and automatically synthesize interacting reactive systems [11]. Consider a server granting access to a printer and connected to several clients. The clients may send requests to the server, and the server grants access to the printer depending on the requests it receives. The server could have various strategies: for instance, never grant access to any client, or always immediately grant access upon request. However, it may also have constraints to satisfy (which define its winning condition): for instance, that no two clients should access the printer at the same time, or that any request must eventually be granted. A strategy for the server is then a policy that it should apply in order to achieve these goals.

Until recently, more focus had been put on the study of purely antagonistic games (a.k.a. zero-sum games), which conveniently represent systems evolving in a (hostile) environment: the aim of one player is to prevent the other player from achieving his own objective.

Non-zero-sum games. Over the last ten years, computer scientists have started considering games with non-zero-sum objectives: they allow for conveniently modelling complex infrastructures where each individual system tries to fulfill its own objectives, while still being subject to uncontrollable actions of the surrounding systems. As an example, consider a wireless network in which several devices try to send data: each device can modulate its transmitting power, in order to maximize its bandwidth or reduce energy consumption as much as possible. In that setting, focusing only on optimal strategies for one single agent is too narrow. Game-theoreticians have defined and studied many other solution concepts for such settings, of which Nash equilibrium [13] is a prominent one. A Nash equilibrium is a strategy profile where no player can improve the outcome of the game by unilaterally changing his strategy. In other terms, in a Nash equilibrium, each individual player has a satisfactory strategy. Notice that Nash equilibria need not exist or be unique, and are not necessarily optimal: Nash equilibria where all players lose may coexist with more interesting Nash equilibria. Finding constrained Nash equilibria (*e.g.*, equilibria in which some players are required to win) is thus an interesting problem for our setting.

Networks of identical devices. Our aim in this paper is to handle the special case where all the interacting systems (but possibly a few of them) are identical. This encompasses many situations involving

Part of this work was sponsored by ERC Starting Grant EQualIS and by EU FP7 project Cassting.

computerized systems over a network. We propose a convenient way of modelling such situations, and develop algorithms for synthesizing a single strategy that, when followed by all the players, leads to a global Nash equilibrium. To be meaningful, this requires symmetry assumptions on the arena of the game (the board should look the same to all the players). We also include *imperfect observation* of the other players, which we believe is relevant in such a setting.

Our contributions. We propose a convenient model for representing large interacting systems, which we call *game structure*. A game structure is made of multiple copies of a single arena (one copy per player); each player plays on his own copy of the arena. As mentioned earlier, the players have imperfect information about the global state of the game (they may have a perfect view on some of their "neighbours", but may be blind to some other players). In *symmetric* game structures, we additionally require that any two players are in similar situations: for every pair of players (A,B), we are able to map each player C to a corresponding player D with the informal meaning that 'player D is to B what player C is to A'. Of course, winning conditions and imperfect information should respect that symmetry. We present several examples illustrating the model, and argue why it is a relevant model for computing symmetric Nash equilibria.

We show several undecidability results, in particular that the parameterized synthesis problem (aiming to obtain one policy that forms a Nash equilibrium when applied to any number of participants) is undecidable. We then characterize the complexity of computing (constrained) pure symmetric Nash equilibria in symmetric game structures, when objectives are given as LTL formulas, and when restricting to memoryless and bounded-memory strategies. This problem with no memory bound is then proven undecidable.

Related work. Game theory has been a very active area since the 1940's, but its applications to computer science *via* graph games is quite recent. In that domain, until recently more focus had been put on zero-sum games [11]. Some recent works have considered multi-player non-zero-sum games, including the computation of (constrained) equilibria in turn-based and in concurrent games [6, 16, 3] or the development of temporal logics geared towards non-zero-sum objectives [5, 7].

None of those works distinguish symmetry constraints in strategy profiles nor in game description. Still, symmetry has been studied in the context of normal-form games [14, 8]: in such a game, each player has the same set of actions, and the utility function of a player only depends on his own action and on the number of players who played each action (it is independent on 'who played what'). Finally, let us mention that symmetry was also studied in the context of model checking, where different techniques have been developped to deal with several copies of the same system [10, 9, 1].

By lack of space, most of the technical developments could not be included in this extended abstract. They are available in the technical report [4].

2 Nash equilibria in Symmetric Games with Partial Observation

2.1 Definitions

Preliminary definitions. For any $k \in \mathbb{N} \cup \{\infty\}$, we write $[k]$ for the set $\{i \in \mathbb{N} \mid 0 \leq i < k\}$ (in particular, $[\infty] = \mathbb{N}$). Let $s = (p_i)_{i\in[n]}$ be a sequence, with $n \in \mathbb{N} \cup \{\infty\}$ being the length $|s|$ of s. Let $j \in \mathbb{N}$ s.t. $j-1 < n$. The jth element of s, denoted s_{j-1}, is the element p_{j-1} (so that a sequence $(p_i)_{i\in[n]}$ may be named p when no ambiguity arises). The jth prefix $s_{<j}$ of s is the finite sequence $(p_i)_{i\in[j]}$. If s is finite, we write $\mathsf{last}(s)$ for its last element $s_{|s|-1}$.

Given a mapping $f\colon A \to B$, $a \in A$, and $b \in B$, we write $f[a \mapsto b]$ for the mapping g s.t. $g(a) = b$ and $g(\alpha) = f(\alpha)$ for all $\alpha \in A \setminus \{a\}$. Given a mapping $f\colon A \to B$ and a mapping $g\colon B \to C$, the composition of g and f is the mapping $g \circ f\colon A \to C$ defined as $g \circ f(a) = g(f(a))$. Seeing a sequence $s = (p_i)_{i\in[n]}$ as a mapping with domain $[n]$, and given a mapping $f\colon [m] \to [n]$, we write $s \circ f$ for the sequence $(p_{f(j)})_{j\in[m]}$.

Concurrent games. Concurrent games are used in the verification community as a tool to reason about and automatically synthesize interacting reactive systems. The model of concurrent games is defined as follows:

Definition 1 *A* concurrent game *is a tuple* $\langle \mathit{States}, \mathit{Agt}, \mathit{Act}, \mathit{Mov}, \mathit{Tab} \rangle$ *where*

- States *is a finite set of states;*
- Agt *is a finite set of agents (also named players);*
- Act *is a finite set of actions;*
- $\mathit{Mov}\colon \mathit{States} \times \mathit{Agt} \to 2^{\mathit{Act}} \setminus \{\emptyset\}$ *is the set of actions available to a given player in a given state;*
- $\mathit{Tab}\colon \mathit{States} \times \mathit{Act}^{\mathit{Agt}} \to \mathit{States}$ *is a transition function that specifies the next state, given a state and an action of each player.*

The evolution of such a game is as follows: from a state s, each player A concurrently selects an available action $m_A \in \mathsf{Mov}(s,A)$. The successor state of s under the *move* $(m_A)_{A\in\mathsf{Agt}}$ is then looked up in Tab. A *path* in $\mathscr{G}$ from state s is a sequence $(s_i)_{i\geq 0}$ of states such that $s_0 = s$ and for all $i \geq 0$, there is a move $(m_A)_{A\in\mathsf{Agt}}$ such that $s_{i+1} \in \mathsf{Tab}(s_i, (m_A)_{A\in\mathsf{Agt}})$. Finite paths are called *histories*, while infinite paths are called *plays*. We write Path (resp. Hist, Play) for the set of paths (resp. histories, plays) in $\mathscr{G}$ (from any state).

Let $A \in \mathsf{Agt}$. A *strategy* for A is a mapping $\sigma_A\colon \mathsf{Hist} \to \mathsf{Act}$ such that for any $\rho \in \mathsf{Hist}$, $\sigma_A(\rho) \in \mathsf{Mov}(\mathsf{last}(\rho), A)$. Given a set of players $C \subseteq \mathsf{Agt}$, a strategy for C is a mapping σ assigning to each $A \in C$ a strategy for A (we will write σ_A instead of $\sigma(A)$ to alleviate notations). As a special case, a strategy for Agt is called a *strategy profile*.

A path π is *compatible* with a strategy σ of coalition C if, for any $i < |\pi|$, there exists a move $(m_A)_{A\in\mathsf{Agt}}$ such that $\mathsf{Tab}(\rho_{i-1}, (m_A)_{A\in\mathsf{Agt}}) = \rho_i$ and $m_A = \sigma_A(\rho_{<i})$ for all $A \in C$. The set of *outcomes* of σ from a state s, denoted $\mathsf{Out}(s,\sigma)$, is the set of plays from s that are compatible with σ.

Nash equilibria. Let $\mathscr{G}$ be a game. A *winning condition* for player A is a set Ω_A of plays of $\mathscr{G}$. We say that a play $\rho \in \Omega_A$ yields payoff 1 to A, and a play $\rho \notin \Omega_A$ yields payoff 0 to A. Winning conditions are usually infinite, but will most often be given symbolically as the set of plays satisfying a given property. For instance, given ϕ a formula of some logic (like LTL, see e.g. [2]) using States as atomic propositions, we write $\Omega(\phi)$ for the set of plays satisfying ϕ. A strategy σ of a coalition C is *winning* for A from a state s if $\mathsf{Out}(s,\sigma) \subseteq \Omega_A$. A strategy profile σ is a *Nash equilibrium* if, for any $A \in \mathsf{Agt}$ and any strategy σ'_A, if σ is losing for A, then so is $\sigma[A \mapsto \sigma'_A]$. In other terms, no player can individually improve his payoff.

Remark 2 *In this paper, we restrict to* pure *Nash equilibria, which correspond to deterministic programs for the players. Considering randomized strategies would clearly be of interest in presence of symmetry, and is part of our future works.*

Remark 3 *In this paper, we only use purely Boolean winning conditions, but our algorithms could easily be extended to the* semi-quantitative *setting of [3], where each player has several (pre)ordered Boolean objectives. We omit such extensions in this paper, and keep focus on symmetry issues.*

2.2 Symmetric Concurrent Games

As mentioned in the introduction, our aim is to propose a convenient way of modelling situations where all the interacting systems are identical, and to develop algorithms for synthesizing symmetric strategy profiles in that setting (which would be in fact a single strategy that will be played by all the players).

The model we propose is made of a one-player arena, together with an observation relation. Intuitively, each player plays in his own copy of the one-player arena; the global system is the product of all the local copies, but each player observes the state of the global system only through the observation relation. This is in particular needed for representing large networks of systems, in which each player may only observe some of his neighbours.

Example 4 *Consider for instance a set of identical devices (e.g. cell phones) connected on a local area network. Each device can modulate its emitting power. In order to increase its bandwidth, a device tends to increase its emitting power; but besides consuming more energy, this also adds* noise *over the network, which decreases the other players' bandwidth and encourages them to in turn increase their power. We can model a device as an n-state arena (state i corresponding to some power p_i, with $p_0 = 0$ representing the device being off). Any device would not know the exact state of the other devices, but would be able to evaluate the surrounding noise; this can be modelled using our observation relation. Based on this information, the device can decide whether it should increase or decrease its emitting power (knowing that the other devices plays the same strategy as it is playing), resulting in a good balance between bandwidth and energy consumption.*

Despite the global arena being described as a product of identical arenas, not all games described this way will be symmetric: the observation relation also has to be symmetric. We will impose extra conditions on that relation in order to capture our expected notion of symmetry. Moreover, the observation relation relates global states of the system, and an explicit description of it will most often not be practical. We will thus consider compact representations of this relation.

2.2.1 Formalization of the model

Game networks. We first define our notion of an n-player game network, which describes a complex game as a product of n identical one-player games.

Definition 5 *An* n-player game network *is a tuple* $\mathscr{G} = \langle G, (\equiv_i)_{i\in[n]}, (\Omega_i)_{i\in[n]}\rangle$ *s.t.*

- $G = \langle \mathit{States}, \{A\}, \mathit{Act}, \mathit{Mov}, \mathit{Tab}\rangle$ *is a one-player arena;*
- *for each* $i \in [n]$, $\equiv_i$ *is an equivalence relation on* States^n *(extended in a natural way to sequences of states of* States^n*). Two* $\equiv_i$*-equivalent configurations are indistinguishable to player i. This models* imperfect information *for player i;*
- *for each* $i \in [n]$, $\Omega_i \subseteq (\mathit{States}^n)^\omega$ *is the objective of player i. We require that for all* $\rho, \rho' \in (\mathit{States}^n)^\omega$*, if* $\rho \equiv_i \rho'$ *then* ρ *and* ρ' *are equivalently in* Ω_i*.*

The semantics of this game is defined as the "product game" $\mathscr{G}' = \langle \mathsf{States}', [n], \mathsf{Act}, \mathsf{Mov}', \mathsf{Tab}', (\Omega_i)_{i\in[n]}\rangle$ where $\mathsf{States}' = \mathsf{States}^n$, $\mathsf{Mov}'((s_0,\dots,s_{n-1}),i) = \mathsf{Mov}(s_i)$, and the transition table is defined as

$$\mathsf{Tab}'((s_0,\dots,s_{n-1}),(m_i)_{i\in[n]}) = (\mathsf{Tab}(s_0,m_0),\dots,\mathsf{Tab}(s_{n-1},m_{n-1})).$$

Notice that we do not fix an initial state for the one-player arena, as we want to consider cases where the players start in different states, thus modelling the setting where not all players start playing at the same time. This also makes the problem more interesting, as playing the same strategy from a *symmetric* state (*i.e.*, made of n copies of the same state) would only visit symmetric states.

Example 6 *Consider the cell-phone game again. It can be modelled as a game network where each player observes everything (i.e., the equivalence relations $\equiv_i$ are the identity). A more realistic model for the system can be obtained by assuming that each player only gets precise information about his close neighbours, and less precise information (only an estimation of the global noise in the network), or no information at all, about the devices that are far away.*

We now give some further useful definitions. An element of States^n is called a *configuration* of $\mathscr{G}$. Equivalence relation $\equiv_i$ induces equivalence classes of configurations that player i cannot distinguish. We call these equivalence classes *information sets* and denote $\mathscr{I}_i$ the set of information sets for player i. Strategies should respect these information sets: a strategy σ_i for player i is $\equiv_i$-*realisable* whenever $\rho \equiv_i \rho'$ implies $\sigma_i(\rho) = \sigma_i(\rho')$. A strategy profile $\sigma = (\sigma_i)_{1\leq i\leq n}$ is said *realisable* whenever σ_i is $\equiv_i$-realisable for every $i \in [n]$.

Symmetric game networks. If we impose no restriction on the observation relation, n-player game networks do not fully capture symmetries in a system. Besides playing on similar arenas, we will add the extra requirement that all the players are in similar situations w.r.t. the other players.

Given a permutation π of $[n]$, for a configuration $t = (s_i)_{i\in[n]}$ we define $t(\pi) = (s_{\pi(i)})_{i\in[n]}$; similarly, for a path $\rho = (t_j)_{j\in\mathbb{N}}$, we define $\rho(\pi) = (t_j(\pi))_{j\in\mathbb{N}}$.

We now refine the previous definition for a game network to capture symmetries in the system.

Definition 7 *A game network $\mathscr{G} = \langle G, (\equiv_i)_{i\in[n]}, (\Omega_i)_{i\in[n]}\rangle$ is* symmetric *whenever for any two players $i, j \in [n]$, there is a permutation $\pi_{i,j}$ of $[n]$ such that $\pi_{i,j}(i) = j$ and satisfying the following conditions: for every $i, j, k \in [n]$,*

1. *$\pi_{i,i}$ is the identity, and $\pi_{k,j} \circ \pi_{i,k} = \pi_{i,j}$; hence $\pi_{i,j}^{-1} = \pi_{j,i}$.*
2. *the observation made by the players is compatible with the symmetry of the game: for any two configurations t and t', $t \equiv_i t'$ iff $t(\pi_{i,j}^{-1}) \equiv_j t'(\pi_{i,j}^{-1})$;*
3. *objectives are compatible with the symmetry of the game: for every play ρ, $\rho \in \Omega_i$ iff $\rho(\pi_{i,j}^{-1}) \in \Omega_j$.*

In that case, $\pi = (\pi_{i,j})_{i,j\in[n]}$ is called a symmetric representation *of $\mathscr{G}$.*

The mappings $\pi_{i,j}$ define the symmetry of the game: $\pi_{i,j}(k) = l$ means that player l plays vis-à-vis player j the role that player k plays vis-à-vis player i. We give the intuition why we apply $\pi_{i,j}^{-1}$ in the definition above, and not $\pi_{i,j}$. Assume configuration $t = (s_0, \ldots, s_{n-1})$ is observed by player i. The corresponding configuration for player j is $t' = (s'_0, \ldots, s'_{n-1})$ where player-$\pi_{i,j}(k)$ state should be that of player k in t. That is, $s'_{\pi_{i,j}(k)} = s_k$, so that $t' = t(\pi_{i,j}^{-1})$. As mentioned in the introduction, we discuss such subtleties in the context of normal-form games in Appendix A.

These mappings define how symmetry must be used in strategies: let $\mathscr{G}$ be a symmetric n-player game network with symmetric representation π. We say that a strategy profile $\sigma = (\sigma_i)_{i\in[n]}$ is *symmetric* for the representation π if it is realisable (*i.e.*, each player only plays according to what he can observe) and if for all $i, j \in [n]$ and every history ρ, it holds $\sigma_i(\rho) = \sigma_j(\rho(\pi_{i,j}^{-1}))$.

Example 8 *Consider a card game tournament with six players, three on each table. Here each player has a left neighbour, a right neighbour, and three opponents at a different table. To model this, one could assume player* 0 *knows everything about himself, and has some informations about his right neighbour (player* 1*) and his left neighbour (player* 2*). But he knows nothing about players* 3*,* 4 *and* 5*.*

Now, the role of player 2 *vis-à-vis player* 1 *is that of player* 1 *vis-à-vis player* 0 *(he is his right neighbour). Hence, we can define the symmetry as $\pi_{0,1}(0) = 1$, $\pi_{0,1}(1) = 2$, $\pi_{0,1}(2) = 0$, and $\pi_{0,1}(\{3,4,5\}) = \{3,4,5\}$ (any choice is fine here). As an example, the observation relation in this setting could be that*

player 0 *has perfect knowledge of his set of cards, but only knows the number of cards of players* 1 *and* 2*, and has no information about the other three players. Notice that other observation relations would have been possible (for instance, giving more information about the right player).*

Example 9 *Consider again the cell-phone example. In this model, the noise depends on the relative positions of the devices, and in that sense this game is not symmetric. The model of the cell-phone could include informations about the relative positions of the other devices, by including several disjoints copies of the model, in which the neighbour devices have different influences over the noise. The initial state for each player would then depend on the topology of the network.*

Example 10 *Finally, let us mention that even though it is not fully symmetric, it is possible to model a client-server architecture in our framework. Let S be a model for the server and C be a model for the client. The game arena G will then be the disjoint union of S and C, and the equivalence $\equiv_i$ will look like: "if player i is in part S, then he has perfect information on all players, and if player i is in part C, then he sees his own states and the state of player* 0*", having in mind that player* 0 *will be the server and all other players will be clients. Such a game is not symmetric since the server observes more players than the clients do. In order to make the game fully symmetric, we would add extra players, trapped in a sink state, and observable by the clients. See Appendix B for a possible modeling.*

Discussion on the model. Note that symmetric representations are not unique in general. We now analyse how this interferes with the existence of Nash equilibria.

We first define for each player a partitioning of the set of all the players, that will define what the players cannot distinguish. For every player $i \in [n]$, we let $\cong_i$ be the following equivalence relations on the set of players $[n]$: $j \cong_i k$ iff for every configuration t, $t \equiv_i t(\pi_{j\leftrightarrow k})$, where $\pi_{j\leftrightarrow k}$ is the permutation of j and k. It means that player i cannot distinguish between the players j and k. We then define for every $i \in [n]$, the partition $\mathscr{P}_i$ of $[n]$ which is induced by $\cong_i$. We call $\mathscr{P} = (\mathscr{P}_i)_{i\in[n]}$ the canonical partitioning for $\mathscr{G}$.

Lemma 11 *For every symmetric representation π of $\mathscr{G}$, for every $i, j \in [n]$, for every $P_i \in \mathscr{P}_i$, it holds $\pi_{i,j}(P_i) \in \mathscr{P}_j$.*

Proof. Assume that it is not the case. There are two cases:

- either $\pi_{i,j}(P_i) \subsetneq P_j$ for some $P_j \in \mathscr{P}_j$. Take $k_j \in P_j \subseteq \pi_{i,j}(P_i)$ and $p_j \in \pi_{i,j}(P_i)$. For every configuration t, we have that $t \equiv_j t(\pi_{k_j\leftrightarrow p_j})$. We define $p_i = \pi_{i,j}^{-1}(p_j)$ (which is then in P_i) and $k_i = \pi_{i,j}^{-1}(k_j)$ (which is then not in P_i). As π is a symmetric representation of $\mathscr{G}$, we have that $t(\pi_{i,j}^{-1}) \equiv_i t(\pi_{i,j}^{-1} \circ \pi_{k_j\leftrightarrow p_j})$. We can now notice that $t(\pi_{i,j}^{-1} \circ \pi_{k_j\leftrightarrow p_j}) = t(\pi_{k_i\leftrightarrow p_i} \circ \pi_{i,j}^{-1})$, which then implies $t(\pi_{i,j}^{-1}) \equiv_i t(\pi_{k_i\leftrightarrow p_i} \circ \pi_{i,j}^{-1})$. For every t', we therefore get $t' \equiv_i t'(\pi_{k_i\leftrightarrow p_i})$. This contradicts the fact that $p_i \in P_i$ and $k_i \notin P_i$. This case is not possible.
- or there exists $P_j \neq P'_j \in \mathscr{P}_j$ such that $\pi_{i,j}(P_i) \cap P_j \neq \emptyset$ and $\pi_{i,j}(P_i) \cap P'_j \neq \emptyset$. The reasoning is similar as above. □

Example 12 *For a configuration $t = (s_i)_{i\in[n]}$ and a subset P of players, we define t_P as the subsequence $(s_i)_{i\in P}$, and its Parikh image* $\mathsf{Parikh}(t_P)$ *as the function mapping each state s to its number of occurrences in t_P. Now, we define the observation relation* $\mathsf{Parikh}(P)$ *as follows:*

$$(t,t') \in \mathsf{Parikh}(P) \quad \textit{iff} \quad \mathsf{Parikh}(t_P) = \mathsf{Parikh}(t'_P).$$

Similarly, we define the observation relation $\mathsf{Id}(P)$ *as*

$$(t,t') \in \mathsf{Id}(P) \quad \textit{iff} \quad t[i] = t'[i] \textit{ for all } i \in P.$$

Using these relations and the fact that the intersection of two equivalence relations is an equivalence relation, we can define various observation relations, for instance the relation $\equiv_i$ defined by

$$\mathsf{Id}(\{i\}) \wedge \mathsf{Parikh}(\{i+1,i+2,i+3\}) \wedge \mathsf{Parikh}(\{i+3,i+4,i+5\})$$

(where all indices are taken modulo n). With such an observation, player i has perfect information about his own state, and knows the Parikh images for players $i+1$, $i+2$ and $i+3$ and for players $i+3$, $i+4$ and $i+5$. One can check that the partition $\mathscr{P}_i$ is then $(\{i\},\{i+1,i+2\},\{i+3\},\{i+4,i+5\})$ (where player $i+3$ plays a special role as he appears in the two Parikh conditions).

There are two reasons why a symmetric game network may admit several symmetric representations: First, symmetric representations may swap players that behave the same or that are observed the same. For instance, in a three-player game where each player only observes the Parikh image of the other two players, mappings π can either be defined as $\pi_{0,1}(1)=2$ and $\pi_{0,1}(2)=0$, or $\pi_{0,1}(1)=0$ and $\pi_{0,1}(2)=2$. Such distinctions are harmless in general, and those will generate the same symmetric behaviours. More precisely:

Lemma 13 *Let $\mathscr{G}$ be a symmetric n-player game network, and assume $\mathscr{P}$ is the canonical partitioning of $\mathscr{G}$. Take two symmetric representations π and $\widetilde{\pi}$ for $\mathscr{G}$. Assume that for every $i \in [n]$, for every piece $P \in \mathscr{P}_i$, $\pi_{i,j}(P) = \widetilde{\pi}_{i,j}(P)$. Then, a strategy profile σ is symmetric for π iff it is symmetric for $\widetilde{\pi}$.*

Proof. It is sufficient to show that for every configuration t, $t(\pi_{i,j}^{-1}) \equiv_j t(\widetilde{\pi}_{i,j}^{-1})$.

Let π be a permutation of $[n]$ that preserves partition $\mathscr{P}_i$ such that $\widetilde{\pi}_{i,j} = \pi_{i,j} \circ \pi$. Let t be a configuration. As π preserves $\mathscr{P}_i$, $t \equiv_i t(\pi^{-1})$. This implies, if we apply the symmetry condition for $\pi_{i,j}$: $t(\pi_{i,j}^{-1}) \equiv_j t(\pi^{-1} \circ \pi_{i,j}^{-1})$, that is, $t(\pi_{i,j}^{-1}) \equiv_j t(\widetilde{\pi}_{i,j}^{-1})$. Therefore the symmetry condition for the strategy profile does not depend on the choice of the symmetry mappings. □

Symmetric representations might however differ more 'dramatically'. Assume for instance that $n=6$, and that $\equiv_i$ is defined for every $i \in [6]$ as '$\mathsf{Id}(\{i\}) \wedge \mathsf{Parikh}(\{i+1,i+2\}) \wedge \mathsf{Parikh}(\{i+3,i+4\})$' (taken modulo 6). Then the canonical partition $\mathscr{P}_i$ is equal to $(\{i\},\{i+1,i+2\},\{i+3,i+4\},\{i+5\})$, and the mappings $\pi_{i,j}(i+k) = j+k \mod 6$ properly define the symmetry. But there are other mappings that define the symmetry, for instance:

$$\pi'_{2i,2i+1}: \begin{cases} 2i & \mapsto 2i+1 \\ 2i+1 & \mapsto 2i+4 \\ 2i+2 & \mapsto 2i+5 \\ 2i+3 & \mapsto 2i+2 \\ 2i+4 & \mapsto 2i+3 \\ 2i+5 & \mapsto 2i \end{cases} \qquad \pi'_{2i+1,2i+2}: \begin{cases} 2i & \mapsto 2i+1 \\ 2i+1 & \mapsto 2i+2 \\ 2i+2 & \mapsto 2i+5 \\ 2i+3 & \mapsto 2i \\ 2i+4 & \mapsto 2i+3 \\ 2i+5 & \mapsto 2i+4 \end{cases}$$

The other mappings are obtained by composition. This also properly represents the symmetry, but generates different symmetric strategy profiles. Under additional technical conditions, we can prove that Nash equilibria coincide for two symmetric representations of a given symmetric game network. First we realise that a symmetric strategy profile is fully determined by an $\equiv_0$-realisable strategy for player 0.

Lemma 14 *Fix a symmetric representation π for $\mathscr{G}$. If σ_0 is an $\equiv_0$-realisable strategy for player 0, then the strategy profile σ defined by $\sigma_i(\rho) = \sigma_0(\rho(\pi_{i,0}^{-1}))$ defines a realisable and symmetric strategy profile.*

Proof. Symmetry is straightforward: $\sigma_j(\rho(\pi_{i,j}^{-1})) = \sigma_0(\rho(\pi_{j,0}^{-1} \circ \pi_{i,j}^{-1})) = \sigma_0(\rho(\pi_{i,0}^{-1})) = \sigma_i(\rho)$.

Assume that σ_i is not $\equiv_i$-realisable: this means that there are two runs $\rho \equiv_i \rho'$ such that $\sigma_i(\rho) \neq \sigma_i(\rho')$. By the symmetry of the game, it holds that $\rho(\pi_{i,0}^{-1}) \equiv_0 \rho'(\pi_{i,0}^{-1})$, which implies that $\sigma_0(\rho(\pi_{i,0}^{-1})) = \sigma_0(\rho'(\pi_{i,0}^{-1}))$. However this precisely means $\sigma_i(\rho) = \sigma_i(\rho')$. Strategy σ_i is $\equiv_i$-realisable. □

We can now show the following result.

Lemma 15 *Assume that a symmetric representation π of game network $\mathscr{G} = (G, (\equiv_i)_{i\in[n]}, (\Omega_i)_{i\in[n]})$ has to satisfy the following additional property: if there exist permutations $(\kappa_i)_{i\in[n]}$ of $[n]$ such that:*

(i) $\kappa_i(i) = i$ for every i

(ii) for every two configurations t and t',

$$t \equiv_i t' \;\Leftrightarrow\; t(\kappa_i \circ \pi_{j,i} \circ \kappa_j^{-1} \circ \pi_{i,j}) \equiv_i t'(\kappa_i \circ \pi_{j,i} \circ \kappa_j^{-1} \circ \pi_{i,j})$$

(iii) for every run ρ, ρ and $\rho(\kappa_i \circ \pi_{j,i} \circ \kappa_j^{-1} \circ \pi_{i,j})$ are equivalently in Ω_i

then for every configuration t, $t \equiv_i t(\kappa_i \circ \pi_{j,i} \circ \kappa_j^{-1} \circ \pi_{i,j})$. Under that additional constraint, the choice of the representations does not affect Nash equilibria. More precisely: if π and $\widetilde{\pi}$ are two symmetric representations of game $\mathscr{G}$ which satisfy the above hypothesis, then a realisable strategy profile σ is a symmetric Nash equilibrium from t in $\mathscr{G}$ for representation π iff it is a symmetric Nash equilibrium from t in $\mathscr{G}$ for representation $\widetilde{\pi}$.

Proof. Let $(\pi_{i,j})_{i,j}$ and $(\widetilde{\pi}_{i,j})_{i,j}$ be two different representations (for the pieces of the canonical partitioning). Those mappings are uniquely characterized by $(\pi_{0,i})_i$ and $(\widetilde{\pi}_{0,i})_i$. Assume κ_i is the permutation of $[n]$ such that $\widetilde{\pi}_{0,i} = \kappa_i \circ \pi_{0,i}$ (in particular w.l.o.g. κ_i swaps pieces of $\mathscr{P}_i$). We notice that $\widetilde{\pi}_{i,j} = \kappa_j \circ \pi_{i,j} \circ \kappa_i^{-1}$.

It is not difficult to prove all the conditions for the κ_i's. We therefore get that $t \equiv_i t(\kappa_i \circ \pi_{j,i} \circ \kappa_j^{-1} \circ \pi_{i,j})$ for every i and j, and in particular, taking $j = 0$, we get that $t \equiv_i t(\kappa_i)$ (since κ_0 is the identity).

Fix a strategy σ_0 for player 0 (which is $\equiv_0$-realisable). It defines two strategy profiles σ and $\widetilde{\sigma}$. For every i, we compute:

$$\begin{aligned}
\widetilde{\sigma}_i(\rho) &= \sigma_0(\rho(\widetilde{\pi}_{i,0}^{-1})) \\
&= \sigma_0(\rho(\kappa_i \circ \pi_{i,0}^{-1})) \\
&= \sigma_i(\rho(\kappa_i)) \\
&= \sigma(\rho) \text{ since } \rho \equiv_i \rho(\kappa_i)
\end{aligned}$$

In particular, $\widetilde{\sigma}$ and σ have the same outcome, yielding the same payoff to all players.

Now if one othe players can improve, say player i can use strategy σ_i' to improve his strategy σ_i, then he can also improve by playing the same strategy. □

In the sequel, we always assume that the symmetric representation is given. Note however that a symmetric representation can be computed in space polynomial in the number of players, by just enumerating the permutations and checking that they satisfy the constraints.

Discussion on the encoding of symmetric game networks. One motivation for the definition of this model is to represent large networks of identical systems in a rather compact way. To this aim, we need a succinct representation of game networks, in particular for the relations $\equiv_i$. Notice that representing those equivalence relations explicitely (*i.e.*, as $|\mathsf{States}|^n \times |\mathsf{States}|^n$ tables indicating whether the configurations are equivalent or not) is not practicable. We therefore assume that each equivalence relation $\equiv_i$ is given *compactly* as a function $\varphi_i \colon \mathsf{States}^n \times \mathsf{States}^n \to \{0,1\}$ that can be encoded in space polynomial in $|\mathsf{States}|$ and independent of n, whose value can be computed in polynomial time for any two configurations, with $t \equiv_i t'$ iff $\varphi_i(t,t') = 1$. Examples of such functions are $\mathsf{Parikh}(P)$ and $\mathsf{Id}(P)$ defined previously.

Figure 1: Matching penny as a symmetric game network

Problems we are interested in. In this paper we are interested in the computation of (symmetric) Nash equilibria in symmetric game networks. More precisely, we are interested in the following two problems:

Problem 1 (Existence of (symmetric) NE) *The* existence problem *asks, given a symmetric game network $\mathscr{G}$, a symmetric representation π, and a configuration t, whether there is a (symmetric) Nash equilibrium in $\mathscr{G}$ from t for the representation π.*

Remark 16 *There might not exist a (pure) Nash equilibrium in a symmetric game network. Figure 1 shows how one can simulate the matching-penny game. We assume there are two players, and they both have perfect information. Player* 0 *starts from p_0 whereas player* 1 *starts from q_0. The objective is the same for player* 0 *and for player* 1 *and is written:*

$$\Big(p_0 \Rightarrow (\mathbf{F}\,((p_+ \wedge q_+) \vee (p_- \wedge q_-)))\Big) \wedge \Big(q_0 \Rightarrow (\mathbf{F}\,((p_+ \wedge q_-) \vee (p_- \wedge q_+)))\Big).$$

This reads as follows: "if you are in p_0, then you have to eventually visit both p_+ and q_+, or both p_- and q_-, and if you are in q_0, you have to eventually visit both p_+ and q_-, or both p_- and q_+". It is not hard to be convinced that it is symmetric, and that there is no pure Nash equilibrium from (p_0, q_0) in that game network.

Problem 2 (Constrained existence of (symmetric) NE) *The* constrained existence problem *asks, given a symmetric game network $\mathscr{G}$, a symmetric representation π, a configuration t, a set $L \subseteq [n]$ of losing players, and a set $W \subseteq [n]$ of winning players, whether there is a (symmetric) Nash equilibrium σ in $\mathscr{G}$ from t for the representation π, such that all players in L lose and all players in W win. If $W = [n]$, the problem is called the* positive existence problem*.*

We first realise that even though symmetric Nash equilibria are Nash equilibria with special properties, they are in some sense at least as hard to find as Nash equilibria.

Proposition 17 *From a symmetric game network $\mathscr{G}$ we can construct in polynomial time a symmetric game network $\mathscr{H}$ such that there exists a symmetric Nash equilibrium in $\mathscr{H}$ if and only if there exists a Nash equilibrium in $\mathscr{G}$. Furthermore the construction only changes the arena, but does not change the number of players nor the objectives or the resulting payoffs.*

This means that we cannot in general hope to have an algorithm with better complexity for the symmetric problem by using properties of symmetry. This proposition allows furthermore to infer hardness results from the framework with standard Nash equilibria to the framework with symmetric Nash equilibria.

Proof. Let $\mathscr{G} = (G, (\equiv_i)_{i\in[n]}, (\Omega_i)_{i\in[n]})$ be a symmetric game network and $(s_{0,0}, ..., s_{n-1,0})$ be a configuration as an input to the existence problem. We build a symmetric game network $\mathscr{H}$ which has a symmetric Nash equilibrium from some particular configuration if, and only if, $\mathscr{G}$ has a Nash equilibrium from $(s_{0,0}, ..., s_{n-1,0})$. We generate $\mathscr{H}$ as follows.

Let $\mathscr{H} = (H, (\sim_i)_{i\in[n]}, (\Theta_i)_{i\in[n]})$ be a symmetric game network with n players as $\mathscr{G}$. We design H as n disconnected copies of G as shown in Figure 2.

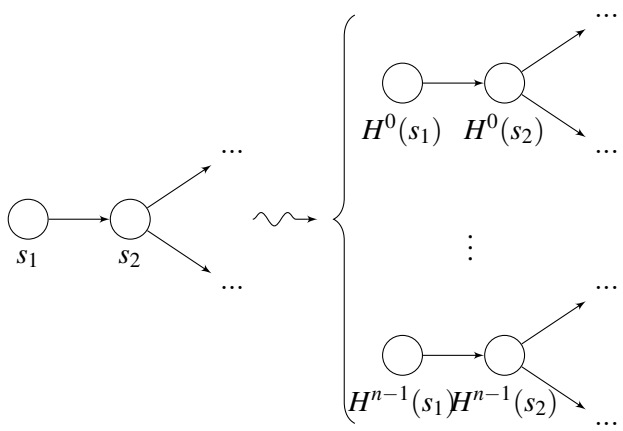

Figure 2: Arena G on the left and H on the right

These copies will be denoted $H^0,...,H^{n-1}$. To denote the state in some H^j corresponding to some state s in G we write $H^j(s)$. We introduce the mapping λ_1 such that $\lambda_1(H^j(s)) = s$ for all states s and all players j. Then we let the initial configuration in $\mathscr{H}$ be $(H^0(s_{0,0}),H^1(s_{1,0}),...,H^{n-1}(s_{n-1,0}))$. In other words, in H each player starts in the same states as in $\mathscr{G}$, but in different copies of G. We define $\sim_i$ such that for all i, for all states $s_0,...,s_{n-1},s'_0,...,s'_{n-1}$ in G and for all $m_0,...,m_{n-1},j_0,...,j_{n-1} \in [n]$, it holds:

$$(H^{m_0}(t_0),H^{m_1}(t_1),...,H^{m_{n-1}}(t_{n-1})) \sim_i (H^{j_0}(v_0),H^{j_1}(v_1),...,H^{j_{n-1}}(v_{n-1}))$$

if, and only if,

$$(t_0,...,t_{n-1}) \equiv_i (v_0,...,v_{n-1}) \wedge m_i = j_i$$

Finally, for any player i and any infinite play $\rho = (s^1_0,...,s^1_{n-1})(s^2_0,...,s^2_{n-1})...$ in H, we define the objectives of the players in $\mathscr{H}$ such that for all $j_0,...,j_{n-1} \in [n]$, $\rho \in \Omega_i$ if, and only if,

$$(H^{j_0}(s^1_0),...,H^{j_{n-1}}(s^1_{n-1}))(H^{j_0}(s^2_0),...,H^{j_{n-1}}(s^2_{n-1}))... \in \Theta_i$$

We start by showing that $\mathscr{H}$ as defined here is really a symmetric game network. First we show that $\sim_i$ is an equivalence relation for all i. Note that every state in H can be written as $H^j(s)$ for a state s in G and an index $j \in [n]$ in a unique way. Now $\sim_i$ is reflexive for all i since for all states $s_0,...,s_{n-1}$ in G and all $j_0,...,j_{n-1} \in [n]$, we have

$$(s_0,...,s_{n-1}) \equiv_i (s_0,...,s_{n-1}) \wedge j_i = j_i \Rightarrow (H^{j_0}(s_0),...,H^{j_{n-1}}(s_{n-1})) \sim_i (H^{j_0}(s_0),...,H^{j_{n-1}}(s_{n-1}))$$

It is symmetric for all i since for all states $s_0,...,s_{n-1},s'_0,...,s'_{n-1}$ in G and all $j_0,...,j_{n-1},m_0,...,m_{n-1} \in [n]$, it holds

$$\begin{aligned}(H^{j_0}(s_0),...,H^{j_{n-1}}(s_{n-1})) &\sim_i (H^{m_0}(s'_0),...,H^{m_{n-1}}(s'_{n-1}))\\ &\Rightarrow (s_0,...,s_{n-1}) \equiv_i (s'_0,...,s'_{n-1}) \wedge j_i = m_i\\ &\Rightarrow (s'_0,...,s'_{n-1}) \equiv_i (s_0,...,s_{n-1}) \wedge j_i = m_i\\ &\Rightarrow (H^{m_0}(s'_0),...,H^{m_{n-1}}(s'_{n-1})) \sim_i (H^{j_0}(s_0),...,H^{j_{n-1}}(s_{n-1})).\end{aligned}$$

It is transitive for all i since for all states $s_0,...,s_{n-1},s'_0,...,s'_{n-1},s''_0,...,s''_{n-1}$ in G and $j_0,...,j_{n-1}$, $k_0,...,k_{n-1},m_0,...,m_{n-1} \in [n]$

$$\begin{aligned}
&(H^{j_0}(s_0),...,H^{j_{n-1}}(s_{n-1})) \sim_i (H^{k_0}(s'_0),...,H^{k_{n-1}}(s'_{n-1})) \wedge \\
&(H^{k_0}(s'_0),...,H^{k_{n-1}}(s'_{n-1})) \sim_i (H^{m_0}(s''_0),...,H^{m_{n-1}}(s''_{n-1})) \\
&\quad \Rightarrow (s_0,...,s_{n-1}) \equiv_i (s'_0,...,s'_{n-1}) \wedge (s'_0,...,s'_{n-1}) \equiv_i (s''_0,...,s''_{n-1}) \wedge j_i = k_i \wedge k_i = m_i \\
&\quad \Rightarrow (s_0,...,s_{n-1}) \equiv_i (s''_0,...,s''_{n-1}) \wedge j_i = m_i \\
&\quad \Rightarrow (H^{j_0}(s_0),...,H^{j_{n-1}}(s_{n-1})) \sim_i (H^{m_0}(s''_0),...,H^{m_{n-1}}(s''_{n-1}))
\end{aligned}$$

This means that $\sim_i$ is an equivalence relation for all i.

Since $\mathscr{G}$ is symmetric there exists a symmetric representation $(\pi_{i,j})_{i,j\in[n]}$ of $\mathscr{G}$. We will show that this is also a symmetric representation of $\mathscr{H}$. It satisfies the first point in the definition of symmetry since $(\pi_{i,j})_{i,j\in[n]}$ is a symmetric representation of $\mathscr{G}$. Secondly, for any two configurations $(H^{k_0}(s_0),...,H^{k_{n-1}}(s_{n-1}))$ and $(H^{k'_0}(s'_0),...,H^{k'_{n-1}}(s'_{n-1}))$ of $\mathscr{H}$ and any $i,j \in [n]$, we have

$$\begin{aligned}
&(H^{k_0}(s_0),...,H^{k_{n-1}}(s_{n-1})) \sim_i (H^{k'_0}(s'_0),...,H^{k'_{n-1}}(s'_{n-1})) \\
&\qquad \Leftrightarrow (s_0,...,s_{n-1}) \equiv_i (s'_0,...,s'_{n-1}) \wedge s_i = s'_i \\
&\qquad \Leftrightarrow (s_0,...,s_{n-1})(\pi_{i,j}^{-1}) \equiv_j (s'_0,...,s'_{n-1})(\pi_{i,j}^{-1}) \wedge s_{\pi_{i,j}^{-1}(j)} = s'_{\pi_{i,j}^{-1}(j)} \\
&\qquad \Leftrightarrow (H^{k_0}(s_0),...,H^{k_{n-1}}(s_{n-1}))(\pi_{i,j}^{-1}) \sim_j (H^{k'_0}(s'_0),...,H^{k'_{n-1}}(s'_{n-1}))(\pi_{i,j}^{-1})
\end{aligned}$$

which means it satisfies the second requirement. For the third point, for every play $\rho = (H^{i_0}(s^0_0),...,H^{i_{n-1}}(s^0_{n-1}))$ $(H^{i_0}(s^1_0),...,H^{i_{n-1}}(s^1_{n-1}))...$ and every $i,j \in [n]$, we have

$$\begin{aligned}
&(H^{i_0}(s^0_0),...,H^{i_{n-1}}(s^0_{n-1}))(H^{i_0}(s^1_0),...,H^{i_{n-1}}(s^1_{n-1}))... \in \Theta_i \\
&\qquad \Leftrightarrow (s^0_0,...,s^0_{n-1})(s^1_0,...,s^1_{n-1})... \in \Omega_i \\
&\qquad \Leftrightarrow (s^0_{\pi_{i,j}^{-1}(0)},...,s^0_{\pi_{i,j}^{-1}(n-1)})(s^1_{\pi_{i,j}^{-1}(0)},...,s^1_{\pi_{i,j}^{-1}(n-1)})... \in \Omega_j \\
&\qquad \Leftrightarrow (H^{i_{\pi_{i,j}^{-1}(0)}}(s^0_{\pi_{i,j}^{-1}(0)}),...,H^{i_{\pi_{i,j}^{-1}(n-1)}}(s^0_{\pi_{i,j}^{-1}(n-1)}))(H^{i_{\pi_{i,j}^{-1}(0)}}(s^1_{\pi_{i,j}^{-1}(0)}),...,H^{i_{\pi_{i,j}^{-1}(n-1)}}(s^1_{\pi_{i,j}^{-1}(n-1)}))... \in \Theta_j
\end{aligned}$$

which is the final step showing that $(\pi_{i,j})_{i,j\in[n]}$ is also a symmetric representation for $\mathscr{H}$ which is therefore a symmetric game network.

We wish to show that there is a Nash equilibrium in $\mathscr{G}$ from configuration $(s_{0,0},...,s_{n-1,0})$ if, and only if, there is a symmetric Nash equilibrium in $\mathscr{H}$ from configuration $(H^0(s_{0,0}),...,H^{n-1}(s_{n-1,0}))$. But first we will introduce a bit of notation. The way we define the equivalence relations $(\sim_i)_{i\in[n]}$ the information sets of a player in $\mathscr{H}$ depends on which copy of G he is in as well as which information set in $\mathscr{G}$ the current configuration corresponds to. We denote the information sets for every player i, every copy j of G and every information set I of player i in $\mathscr{G}$ as follows

$$H_i^j(I) = \{(H^{m_0}(s_0),...,H^{m_{n-1}}(s_{n-1})) \mid (s_0,...,s_{n-1}) \in I \wedge m_i = j\}.$$

We now start with the first direction and assume there is a Nash equilibrium σ in $\mathscr{G}$ from $(s_{0,0},...,s_{n-1,0})$. Then we create the strategy profile σ' in $\mathscr{H}$ such that for all players i,j and all sequences of information sets $I_0,...,I_{k-1}$ of player i in $\mathscr{G}$,

$$\sigma'_i(H_i^j(I_0)...H_i^j(I_{k-1})) = \sigma_j(\pi_{i,j}^{-1}(I_0)...\pi_{i,j}^{-1}(I_{k-1}))$$

which we will prove is a symmetric Nash equilibrium. Note again that in all legal sequences of information sets, a player will stay in the same copy of G and therefore this is a full definition of a strategy for each player. To prove that it is symmetric we need the following result, stating that for all i, j and all information sets I of player i in $\mathscr{G}$, it holds

$$\begin{aligned}
\pi_{i,j}^{-1}(H_i^j(I)) &= \{(H^{m_{\pi_{i,j}^{-1}(0)}}(s_{\pi_{i,j}^{-1}(0)}),...,H^{m_{\pi_{i,j}^{-1}(n-1)}}(s_{\pi_{i,j}^{-1}(n-1)}))|(H^{m_0}(s_0),...,H^{m_{n-1}}(s_{n-1})) \in H_i^j(I)\} \\
&= \{(H^{m_{\pi_{i,j}^{-1}(0)}}(s_{\pi_{i,j}^{-1}(0)}),...H^{m_{\pi_{i,j}^{-1}(n-1)}}(s_{\pi_{i,j}^{-1}(n-1)}))|(s_0,...,s_{n-1}) \in I \wedge m_i = j\} \\
&= \{(H^{m_{\pi_{i,j}^{-1}(0)}}(s_{\pi_{i,j}^{-1}(0)}),...H^{m_{\pi_{i,j}^{-1}(n-1)}}(s_{\pi_{i,j}^{-1}(n-1)}))|(s_{\pi_{i,j}^{-1}(0)},...,s_{\pi_{i,j}^{-1}(n-1)}) \in \pi_{i,j}^{-1}(I) \wedge m_{\pi_{i,j}^{-1}(j)} = j\} \\
&= H_j^j(\pi_{i,j}^{-1}(I))
\end{aligned}$$

The requirement for σ' to be symmetric can now be reformulated as follows. For all players i, j and all information sets $I_0,...I_{k-1}$ of player i in $\mathscr{G}$ we have

$$\begin{aligned}
\sigma_i'(H_i^j(I_0)...H_i^j(I_{k-1})) &= \sigma_j'(\pi_{i,j}^{-1}(H_i^j(I_0))...\pi_{i,j}^{-1}(H_i^j(I_{k-1}))) \\
&\Leftrightarrow \sigma_j(\pi_{i,j}^{-1}(I_0)...\pi_{i,j}^{-1}(I_{k-1})) = \sigma_j'(H_j^j(\pi_{i,j}^{-1}(I_0))...H_j^j(\pi_{i,j}^{-1}(I_{k-1}))) \\
&\Leftrightarrow \sigma_j(\pi_{i,j}^{-1}(I_0)...\pi_{i,j}^{-1}(I_{k-1})) = \sigma_j(\pi_{j,j}^{-1}(\pi_{i,j}^{-1}(I_0))...\pi_{j,j}^{-1}(\pi_{i,j}^{-1}(I_{k-1})))
\end{aligned}$$

Since $\pi_{j,j} = \pi_{j,j}^{-1}$ and $\pi_{j,j}$ is the identity, it follows that the bottom equality is true and therefore σ' is symmetric for representation $(\pi_{i,j})_{i,j\in[n]}$.

To see that σ' is also a Nash equilibrium from $(H^0(s_{0,0}),...,H^{n-1}(s_{n-1,0}))$ consider the deviation of a player p from σ_p' to $\sigma_{p,\text{dev}}'$. We then look at a corresponding deviation of p from σ_p to $\sigma_{p,\text{dev}}$ in $\mathscr{G}$ where for all sequences of information set tuples

$$\sigma_{p,\text{dev}}(I_0,...,I_{k-1}) = \sigma_{p,\text{dev}}'(H_p^p(I_0),...,H_p^p(I_{k-1})).$$

We consider the outcomes of the two profiles in the two games, denoted $\rho_{\mathscr{G}}$ and $\rho_{\mathscr{H}}$ respectively. We wish to show that $\rho_{\mathscr{G}} = \lambda_1(\rho_{\mathscr{H}})$ by induction. For the base case we have

$$\rho_{\mathscr{G},=0} = (s_{0,0},...,s_{n-1,0}) = \lambda_1(H^0(s_{0,0}),...,H^{n-1}(s_{n-1,0})) = \lambda_1(\rho_{\mathscr{H},=0}).$$

As induction hypothesis suppose it holds for prefixes of outcomes with length at most v. Further, let $\rho_{\mathscr{G},\leq v+1} = (s_0^0,...,s_{n-1}^0)...(s_0^{v+1},...,s_{n+1}^{v+1})$ and let I_i^j be the information set for player i in $\mathscr{G}$ which contains s_i^j. We will need that for a move m of all the players we have for all states $s_0,...,s_{n-1}$ in $\mathscr{G}$ that

$$\text{Tab}((s_0,...,s_{n-1}),m) = \lambda_1(\text{Tab}((H^0(s_0),...,H^{n-1}(s_{n-1})),m)).$$

Then we get

$$\begin{aligned}
\rho_{\mathscr{G},\leq v+1} &= \rho_{\mathscr{G},\leq v} \cdot \text{Tab}(\rho_{\mathscr{G},=v}, \sigma[\sigma_p \mapsto \sigma_{p,\text{dev}}](I(\rho_{\mathscr{G},\leq v}))) \\
&= \lambda_1(\rho_{\mathscr{H},\leq v}) \cdot \text{Tab}((s_0^v,...,s_{n-1}^v), \sigma[\sigma_p \mapsto \sigma_{p,\text{dev}}](I(\rho_{\mathscr{G},\leq v}))) \\
&= \lambda_1(\rho_{\mathscr{H},\leq v}) \cdot \lambda_1(\text{Tab}((H^0(s_0^v),...,H^{n-1}(s_{n-1}^v)), \sigma[\sigma_p \mapsto \sigma_{p,\text{dev}}](I(\rho_{\mathscr{G},\leq v})))) \\
&= \lambda_1(\rho_{\mathscr{H},\leq v}) \cdot \lambda_1(\text{Tab}(\rho_{\mathscr{H},=v}, \sigma[\sigma_p \mapsto \sigma_{p,\text{dev}}](I(\rho_{\mathscr{G},\leq v}))))
\end{aligned}$$

Since for all players i and all information sets $I_0,...,I_{k-1}$ of i in $\mathscr{G}$ it holds that $\sigma[\sigma_p \mapsto \sigma_{p,\text{dev}}]_i(I_0,...,I_{k-1}) = \sigma'[\sigma'_p \mapsto \sigma'_{p,\text{dev}}]_i(H^i_i(I_0),...,H^i_i(I_{k-1}))$ we get

$$\begin{aligned}
\rho_{\mathscr{G},\leq v+1} &= \lambda_1(\rho_{\mathscr{H},\leq v}) \cdot \lambda_1(\text{Tab}(\rho_{\mathscr{H},=v}, \sigma[\sigma_p \mapsto \sigma_{p,\text{dev}}](I(\rho_{\mathscr{G},\leq v}))))\\
&= \lambda_1(\rho_{\mathscr{H},\leq v}) \cdot \lambda_1(\text{Tab}(\rho_{\mathscr{H},=v},\\
&\qquad \sigma'[\sigma'_p \mapsto \sigma'_{p,\text{dev}}]((H^0_0(I^0_0),...,H^{n-1}_{n-1}(I^0_{n-1}))...(H^0_0(I^v_0),...,H^{n-1}_{n-1}(I^v_{n-1})))))\\
&= \lambda_1(\rho_{\mathscr{H},\leq v}) \cdot \lambda_1(\text{Tab}(\rho_{\mathscr{H},=v}, \sigma'[\sigma'_p \mapsto \sigma'_{p,\text{dev}}](\rho_{\mathscr{H},\leq v})))\\
&= \lambda_1(\rho_{\mathscr{H},\leq v}) \cdot \lambda_1(\rho_{H,=v+1})\\
&= \lambda_1(\rho_{\mathscr{H},\leq v+1})
\end{aligned}$$

This means that when a player p can deviate from σ' in $\mathscr{H}$ to obtain outcome $\rho_{\mathscr{H}}$ then he can deviate from σ in $\mathscr{G}$ to obtain an outcome $\rho_{\mathscr{G}}$ with $\lambda_1(\rho_{\mathscr{H}}) = \rho_{\mathscr{G}}$. The way we have defined objectives in $\mathscr{H}$ every player will get the same payoff from a play ρ in $\mathscr{H}$ as in $\lambda_1(\rho)$ in $\mathscr{G}$ for every play ρ. And since σ is a Nash equilibrium in $\mathscr{G}$ from $(s_{0,0},...,s_{n-1,0})$ where no player can deviate to improve his payoff then no player can deviate to improve his payoff from σ' in $\mathscr{H}$ from $(H^0(s_{0,0}),...,H^{n-1}(s_{n-1,0}))$ because otherwise that player would be able to deviate from σ in $\mathscr{G}$ to improve his payoff. Thus, σ' is a symmetric Nash equilibrium from $(H^0(s_{0,0}),...,H^{n-1}(s_{n-1,0}))$.

For the other direction we assume there is a symmetric Nash equilibrium σ' in $\mathscr{H}$ from $(H^0(s_{0,0}),...,H^{n-1}(s_{n-1,0}))$. We now define σ in $\mathscr{G}$ for all information sets $I_0,...,I_{k-1}$ of player i by letting

$$\sigma_i(I_0,...,I_{k-1}) = \sigma'_i(H^i_i(I_0),...,H^i_i(I_{k-1})).$$

We wish to show that this is a Nash equilibrium in $\mathscr{G}$ from $(s_{0,0},...,s_{n-1,0})$. Contrary to the previous case we consider a deviation from σ in $\mathscr{G}$ by player p from σ_p to $\sigma_{p,\text{dev}}$ and consider a corresponding deviation in $\mathscr{H}$ from σ' defined by

$$\sigma'_{p,\text{dev}}(H^i_i(I_0),...,H^i_i(I_{k-1})) = \sigma_{p,\text{dev}}(I_0,...,I_{k-1})$$

for all information sets $I_0,...,I_{k-1}$ of player p in $\mathscr{G}$. Let the outcomes of the profiles with the deviations be $\rho_{\mathscr{G}}$ and $\rho_{\mathscr{H}}$. As in the previous case we can show that $\lambda_1(\rho_{\mathscr{H}}) = \rho_{\mathscr{G}}$ which means that when a player p deviates in $\mathscr{G}$ from σ he can do a deviation in $\mathscr{H}$ from σ' which gives him the same payoff. Since σ' is a Nash equilibrium from $(H^0(s_{0,0}),...,H^{n-1}(s_{n-1,0}))$ no player can deviate to improve his payoff from σ'. This means that no player can deviate to improve his payoff from σ in $\mathscr{G}$ from $(s_{0,0},...,s_{n-1,0})$ and therefore it is a Nash equilibrium from this configuration.

Hence there is a symmetric Nash equilibrium from $(H^0(s_{0,0}),...,H^{n-1}(s_{n-1,0}))$ in $\mathscr{H}$ if and only if there is a Nash equilibrium from $(s_{0,0},...,s_{n-1,0})$ in $\mathscr{G}$.

There are n times as many states in the arena H as in the arena G. In addition, there are n times as many equivalence classes, which implies that the size of $\mathscr{H}$ is polynomial in the size of $\mathscr{G}$. □

2.2.2 What makes the computation of symmetric Nash equilibria difficult?

Recent works have considered the computation of Nash equilibria in standard concurrent or turn-based games. In particular, the abstraction of suspect games described in [3] has allowed the development of efficient algorithms for computing Nash equilibria in concurrent games, for various classes of objectives. However those algorithms cannot be applied to our framework for the following reasons:

- each player has only *imperfect information* on the state-space of the game;
- the *symmetry requirement* induces non-local constraints in the game.

Notice that even in the case of symmetric games with perfect information, an approach using *Strategy Logic* [12], which can express Nash equilibria and impose several players to play the same strategy, would not work out-of-the-box, as in our setting strategies are equal *up to a permutation of the states*.

2.3 Deciding the existence of symmetric Nash equilibria

We now list the results we have obtained about computing Nash equilibria in symmetric concurrent games. We first present undecidability results, which provides us with borders beyond which exact computations won't succeed. We then propose algorithms for restricted cases, which we prove are optimal in the worst case.

2.3.1 Undecidability with non-regular objectives

Our games allow for arbitrary Boolean objectives, defined for each player as a set of winning plays. We prove that it is too general to get decidability of our problems even with perfect information.

Theorem 18 *The (constrained) existence of a symmetric Nash equilibrium for non-regular objectives in (two-player) perfect information symmetric game networks is undecidable.*

Proof. We do a reduction of the halting problem for a deterministic two-counter machine which is undecidable. A two-counter machine M is a 3-tuple $M = \langle Q, \Delta, q_F \rangle$ where

- Q is a finite set of control states
- $\Delta\colon Q \setminus \{q_F\} \to \{\text{inc}\} \times \{c,d\} \times Q \cup \{\text{dec}\} \times \{c,d\} \times Q^2$ is an instruction function which assigns an instruction to each state.
- $q_F \in Q$ is a halting state.

A configuration of M is a 3-tuple in $Q \times \mathbb{N} \times \mathbb{N}$. A run of M is a sequence of configurations $\rho = (q_0,c_0,d_0)(q_1,c_1,d_1)...$ where (q_0,c_0,d_0) is the initial configuration (usually assuming $c_0 = d_0 = 0$), and for two consecutive configurations we are in one of the following situations:

- $\Delta(q_i) = (\text{inc}, c, q_{i+1})$, $c_{i+1} = c_i + 1$ and $d_{i+1} = d_i$;
- $\Delta(q_i) = (\text{inc}, d, q_{i+1})$, $d_{i+1} = d_i + 1$ and $c_{i+1} = c_i$;
- $\Delta(q_i) = (\text{dec}, c, q_{i+1}, q)$ for some q, $c_{i+1} = c_i - 1 \geq 0$ and $d_{i+1} = d_i$;
- $\Delta(q_i) = (\text{dec}, d, q_{i+1}, q)$ for some q, $d_{i+1} = d_i - 1 \geq 0$ and $c_{i+1} = c_i$;
- $\Delta(q_i) = (\text{dec}, c, q, q_{i+1})$ for some q, $c_{i+1} = c_i = 0$ and $d_{i+1} = d_i$;
- $\Delta(q_i) = (\text{dec}, d, q, q_{i+1})$ for some q, $d_{i+1} = d_i = 0$ and $c_{i+1} = c_i$.

The run is infinite if there is no i so $q_i = q_F$ and otherwise it is finite with q_F being the halting state in the final configuration of ρ. The problem of deciding if the run of a two-counter machine has a halting run from a configuration (q_0,c_0,c_d) is undecidable and we wish to reduce an instance of this problem to the existence problem in symmetric game networks.

Let M be a deterministic two-counter machine and let (q_0,c_0,d_0) be an initial configuration. From this we create a symmetric game network with 2 players $\mathscr{G} = \langle G, (\equiv_i)_{i\in[2]}, (\Omega_i)_{i\in[2]} \rangle$ where $(s_1,s_2) \equiv_i (s'_1,s'_2)$ iff $s_1 = s'_1$ and $s_2 = s'_2$ for $i = 1,2$. The arena G consists of two disconnected parts. It is shown in Figure 3, but without G'.

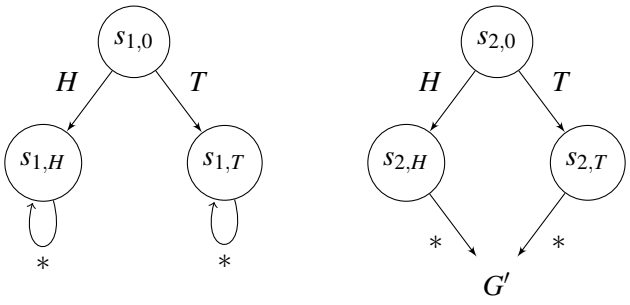

Figure 3: Illustration of G

The idea is that player 1 will start in $s_{1,0}$ and player 2 will start in $s_{2,0}$. They start by playing a matching penny game and afterwards player 2 will play in G' which simulates the counter machine M. Then we will design the objectives so player 2 wins if he acts according to the rules of the counter machine and reaches a halting state. If he does not reach a halting state he wins if the two players choose different coins initially and otherwise player 1 wins. In this way, if there is a legal, halting run of the counter machine then there is a Nash equilibrium where player 2 wins and player 1 loses. If there is no legal halting run then the game is essentially reduced to a matching penny game which has no Nash equilibrium.

Formally, the way we do this is let G' consist of the control states of M with the state connected to $s_{2,H}$ and $s_{2,T}$ being q_0. Then for all i, j there will be an action C^+ taking the play from q_i through an intermediate state C^+_{ij} to q_j if $\Delta(q_i) = (inc, C, q_j)$ for some counter $C \in \{c, d\}$ as illustrated in Figure 4.

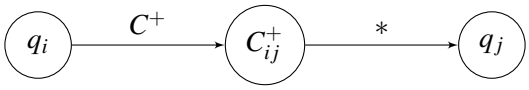

Figure 4: Construction of incrementation module

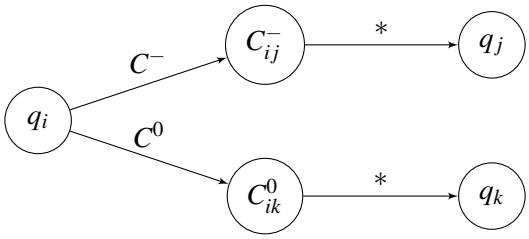

Figure 5: Construction of decrementation module

In addition for all i, j, k there will be an action C^- and an action C^0 respectively taking the play from q_i through intermediate states C^-_{ij} to q_j and C^0_{ik} to q_k if $\Delta(q_i) = (dec, C, q_j, q_k)$ for some counter $C \in \{c, d\}$ as illustrated in Figure 5.

Additionally, we add a self-loop to the halting state q_F. For a finite path ρ we now define

$$C_\rho = \left|\{k \mid \rho_k = C^+_{ij} \text{ for some } i, j\}\right| - \left|\{k \mid \rho_k = C^-_{ij} \text{ for some } i, j\}\right|$$

When given an initial value c_0 and d_0 of the counters we then define the objectives such that player 2 loses in all plays ρ that contains a prefix $\rho_{\leq k}$ such that state $\rho_k = C^-_{ij}$ and $C_0 - C_{\rho_{\leq k}} < 0$ for some C, i

and j to make sure player 2 plays according to the rules of the counter machine and does not subtract from a counter with value zero. In addition, he loses in all plays ρ that contains a prefix $\rho_{\leq k}$ such that $\rho_k = C^0_{ij}$ and $C_0 - C_{\rho_{\leq k}} \neq 0$ to make sure player 2 does not follow the true branch of a zero test when the value of the counter being tested is not zero. Finally, player 2 wins if he does not violate any of these restrictions and reaches q_F. He also wins if he wins the matching penny game (no matter if he violates the restrictions). Player 1 simply wins whenever player 2 doesn't win.

In total this means that there is a Nash equilibrium where player 2 wins and player 1 loses if M halts with initial counter values c_0 and d_0. If M doesn't halt with initial values c_0 and d_0 the game is reduced to a matching penny game which has no Nash equilibrium. Thus, there is a Nash equilibrium in $\mathscr{G}$ if and only if M halts implying that the (constrained) existence problem is undecidable.

The partially defined strategies specified for the two players in the reduction can trivially be extended to symmetric strategies which makes the symmetric (constrained) existence problem undecidable as well. □

Because of this undecidability result, in the sequel we only consider regular objectives. For practical reasons we restrict to objectives defined as LTL formulas, but as our algorithms use automata we could handle regular languages as well.

2.3.2 Undecidability with a parameterized number of players

Parameterized synthesis of Nash equilibria (that is, a single program that each player will apply, and which yields a Nash equilibrium for any number of players) was one of our target applications in this work. We first show that computing such equilibria is unfortunately not possible.

Theorem 19 *The (positive) existence of a parameterized symmetric Nash equilibrium for LTL objectives in symmetric game networks is undecidable (even for memoryless strategies).*

We give a proof for positive existence and with no symmetry constraints in the strategy, and then apply Propositions 17 and 24 to get the expected result.

Proof. To prove this result we simulate a deterministic Turing machine.

Let $\mathscr{M} = (Q, q_0, \Sigma, \delta, \mathsf{Halt})$ be a deterministic Turing machine ($\delta \colon Q \times \Sigma \to Q \times \Sigma \times \{\leftarrow, \rightarrow\}$). We assume Halt is a sink state. We build a symmetric game network which will capture the behaviour of $\mathscr{M}$. We will first show the result when the number of players is supposed to be infinite, and will then explain how we extend the result to a parameterized number of players.

We first define the one-player arena $G = \langle \mathsf{States}, \{A\}, \mathsf{Act}, \mathsf{Mov}, \mathsf{Tab} \rangle$, which is depicted on Figure 6, as follows:

- $\mathsf{States} = (Q \times \Sigma) \cup \Sigma \cup \{\mathsf{Halt}\}$
- $\mathsf{Act} = \Sigma \cup Q$
- $\mathsf{Mov}((q,a),A) = \Sigma$ if $q \neq \mathsf{Halt}$
  $\mathsf{Mov}(a,A) = Q \cup \{a\}$
  $\mathsf{Mov}((\mathsf{Halt},a),A) = \mathsf{Mov}(\mathsf{Halt}) = \{\mathsf{Halt}\}$
- $\mathsf{Tab}((q,a),b) = b$ if $q \neq \mathsf{Halt}$
  $\mathsf{Tab}(a,q) = (q,a)$, $\mathsf{Tab}(a,a) = a$
  $\mathsf{Tab}((\mathsf{Halt},a),\mathsf{Halt}) = \mathsf{Tab}(\mathsf{Halt},\mathsf{Halt}) = \mathsf{Halt}$

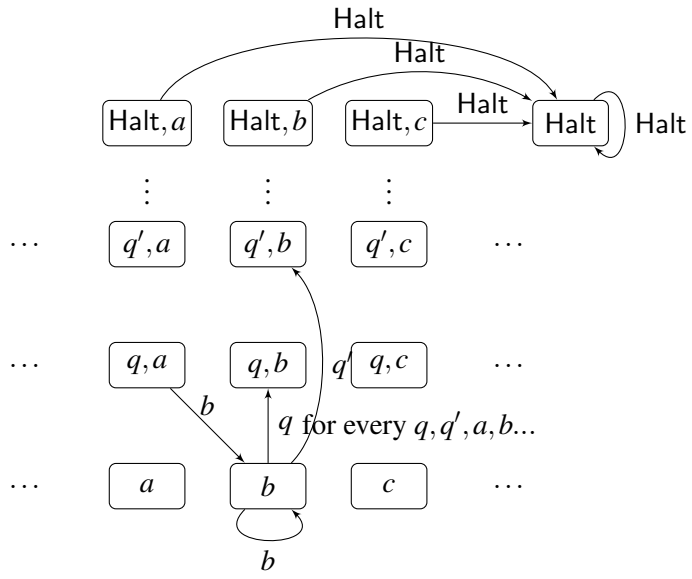

Figure 6: The one-player arena

We then define for every i the equivalence $\equiv_i$ as follows:

$$(\dots, s_{i-1}, s_i, s_{i+1}, \dots) \equiv_i (\dots, s'_{i-1}, s'_i, s'_{i+1}, \dots) \Leftrightarrow \begin{cases} s_{i-1} = s'_{i-1} \\ s_i = s'_i \\ s_{i+1} = s'_{i+1} \end{cases}$$

That is, player i can only observe his right and left neighbours. We now define the objective of player i by an LTL formula over states of players $i-1$, i and $i+1$. In such formulas, p_i holds whenever the state of player i satisfies atomic proposition p:

$$\begin{aligned} \varphi_i \;=\; & \bigwedge_{\substack{(q,a)\in Q\times\Sigma \\ \delta(q,a)=(q',b,\sim)}} \bigwedge_{c,d\in\Sigma} \mathbf{G}\left[\Big((q,a)_i \wedge c_{\sim i} \wedge d_{\not\sim i}\Big) \Rightarrow \mathbf{X}\Big(b_i \wedge (q',c)_{\sim i} \wedge d_{\not\sim i}\Big)\right] \\ & \wedge \bigwedge_{\substack{(q,a)\in Q\times\Sigma \\ \delta(q,a)=(q',b,\sim)}} \bigwedge_{c,d\in\Sigma} \mathbf{G}\left[\Big(c_i \wedge d_{\sim i} \wedge (q,a)_{\not\sim i}\Big) \Rightarrow \mathbf{X}\Big((q',c)_i \wedge d_{\sim i} \wedge b_{\not\sim i}\Big)\right] \\ & \wedge \bigwedge_{a,b,c\in\Sigma} \mathbf{G}\left[\Big(a_{i-1} \wedge b_i \wedge c_{i+1}\Big) \Rightarrow \mathbf{X} b_i\right] \\ & \wedge\; \mathbf{G}\neg \mathsf{Halt}_i \;\wedge\; \mathbf{G}\mathbf{X}\,\mathsf{true} \end{aligned}$$

where we define $\rightarrow i$ is defined as $i+1$, $\leftarrow i$ is $i-1$, and $\not\sim i \in \{i-1, i+1\} \setminus \{\sim i\}$.

Let $\mathscr{G}$ be the above game network with infinitely many players (indexed by $\mathbb{N}$). It is obviously symmetric. We assume there is a special character, say $\flat$, in $\mathscr{M}$ and therefore in Σ, that represents an empty cell. Let w be an input word to $\mathscr{M}$. We set the initial configuration of the game $\mathscr{G}$ (with infinitely many players) by setting the initial state of player 1 to be $(q_0, w[1])$ where q_0 is the initial state of $\mathscr{M}$ and

$w[1]$ is the first letter of w; the initial state of player i ($2 \leq i \leq |w|$) to $w[i]$ and the initial state of all other players to $\flat$. We write γ_w for this initial configuration

Lemma 20 *$\mathscr{M}$ does not halt on input w if, and only if, there is a positive memoryless (or general) Nash equilibrium in $\mathscr{G}$ from γ_w.*

Proof. Assume that $\mathscr{M}$ does not halt. This means that the unique execution of the Turing machine is infinite. In that case, the players should play according to the Turing machine rules. As the computation is infinite, no player reaches its Halt-state and has an infinite local execution (hence satisfies $\mathbf{GX}\,\mathsf{true}$). This is a memoryless strategy profile over information sets, which yields an outcome that is winning for every player.

Assume that there is a Nash equilibrium along which all players win. Then it must be the case that the rules of the Turing machine are properly simulated, and that no player reaches its Halt-state. The computation of the Turing machine is therefore infinite and does not halt. □

To bound the number of players, we now construct another one-player arena $G' = \langle \mathsf{States}', \{A\}, \mathsf{Act}, \mathsf{Mov}', \mathsf{Tab}' \rangle$ where:

- $\mathsf{States}' = \mathsf{States} \times \{L, \#, R\}$
- $\mathsf{Mov}'((s,\bullet),A) = \mathsf{Mov}(s,A)$ if $\bullet \in \{L,\#,R\}$
- $\mathsf{Tab}'((s,\bullet),a) = (\mathsf{Tab}(s,a),\bullet)$

States paired with L (resp. #, R) correspond to the left-most cell of the tape (resp. all states in the middle of the type, the right-most cell of the tape).

We fix an integer n, and define the new game network $\mathscr{G}' = \langle G', (\equiv'_i)_{i\in[n]}, (\Omega'_i)_{i\in[n]} \rangle$ as follows:

$$(s_0,\ldots,s_{n-1}) \equiv'_i (s'_0,\ldots,s'_n) \quad \Leftrightarrow \text{for every } j \in \{i-1 \mod n, i, i+1 \mod n\},\ s_j = s'_j.$$

It remains to define the objective for player i as follows. We write $\widetilde{\varphi}_i$ as the same formula as φ, except that we add copies for every symbol in $\{L,\#,R\}$. The formula is given on Fig. 7. This formula is same as φ_i,

$$\widetilde{\varphi}_i = \bigwedge_{\substack{(q,a)\in Q\times\Sigma \mid \delta(q,a)=(q',b,\sim) \\ c,d\in\Sigma \\ \alpha_1,\alpha_2,\alpha_3\in\{L,\#,R\}}} \left[\begin{array}{l} \mathbf{G}\left[\left(((q,a),\alpha_1)_i \wedge (c,\alpha_2)_j \wedge (d,\alpha_3)_k\right) \Rightarrow \mathbf{X}\left((b,\alpha_1)_i \wedge ((q',c),\alpha_2)_j \wedge (d,\alpha_3)_k\right) \right] \\ \wedge \\ \mathbf{G}\left[\left((c,\alpha_1)_i \wedge (d,\alpha_2)_j \wedge ((q,a),\alpha_3)_k\right) \Rightarrow \mathbf{X}\left(((q',c),\alpha_1)_i \wedge (d,\alpha_2)_j \wedge (b,\alpha_3)_k\right) \right] \end{array} \right]$$

$$\wedge \bigwedge_{\substack{a,b,c\in\Sigma \\ \alpha_1,\alpha_2,\alpha_3\in\{L,\#,R\}}} \mathbf{G}\left[\left((a,\alpha_1)_{i-1} \wedge (b,\alpha_2)_i \wedge (c,\alpha_3)_{i+1}\right) \Rightarrow \mathbf{X}(b,\alpha_2)_i\right]$$

$$\wedge\ \mathbf{F} \bigvee_{s\in\mathsf{States}'} \mathbf{G}\, s_i \ \wedge \bigwedge_{\alpha\in\{L,R\}} \left((\flat,\alpha) \Rightarrow \mathbf{G}(\flat,\alpha)\right)$$

Figure 7: Formula $\widetilde{\varphi}_i$

except that the left-most and the right-most cells should not be changed! That means that the computation of the Turing machine never exceeds n cells.

The initial state of player 0 is set to $(\flat, L)$, the initial state of player 1 is set to $(q_0, (w[1], \#))$, the initial cell of player i ($2 \leq i \leq |w|$) is set to $(w[i], \#)$, the initial state of player $n-1$ is set to $(\flat, R)$, and all other initial states are set to $(\flat, \#)$. We write $\mathcal{G}'(n)$ for the corresponding n-player game network and γ'^n_w for this initial configuration when there are n players. We also turn to the halting problem instead of the non-halting problem.

It is now not hard to prove the following result (realising that $\mathbf{F} \bigvee_{s \in \mathsf{States}'} \mathbf{G}\, s_i$ is satisfied for every i iff some of the player reaches Halt):

Lemma 21 *$\mathscr{M}$ halts on input w if and only if there exists n such that there is a positive memoryless (or positive general) Nash equilibrium in $\mathcal{G}'(n)$ from γ'^n_w. Moreover, n is the space bound required by the Turing machine to halt.*

Remark 22 *In the above reduction one could allow each player to see the Parikh image of the whole configuration of the system, which would allow him to easily detect when the halting state is reached.*

This proves undecidability for the parameterized synthesis problem, but also a EXPSPACE-hardness result for the memoryless case (the number n of players can be exponential, since the equivalence relation can be very compactly represented).

Remark 23 *Our proof applies to LTL objectives, and could therefore be adapted to only involve Büchi objectives. The case of reachability objectives remains open.*

2.3.3 From positive existence to existence

Before turning to our decidability results, we begin with showing that positive existence of Nash equilibria is not harder than existence. Notice that this makes a difference with the setting of turn-based games, where Nash equilibria always exist.

Proposition 24 *Deciding the (symmetric) existence problem in (symmetric) game networks is always at least as hard as deciding the positive (symmetric) existence problem. The reduction doubles the number of players and uses LTL objectives, but does not change the nature of the strategies (memoryless, bounded-memory, or general).*

This result is a consequence of the following lemma, which we prove below.

Lemma 25 *Let $\mathcal{G}$ be an n-player symmetric game network and t_0 be an initial configuration in $\mathcal{G}$. We can construct in polynomial time a $(2n)$-player symmetric game network $\mathcal{G}'$ and a configuration t'_0 in $\mathcal{G}'$ such that there is a Nash equilibrium from t_0 along which all players win in $\mathcal{G}$ if, and only if, there is a Nash equilibrium from t'_0 in $\mathcal{G}'$. Moreover, this equivalence also holds for memoryless and bounded-memory equilibria.*

Proof. Let $\mathcal{G} = (G, (\equiv_i)_{i\in[n]}, (\Omega_i)_{i\in[n]})$ be symmetric game network with n players. Assume the symmetric representation $(\pi_{i,j})_{i,j\in[n]}$. The game $\mathcal{G}'$ will consist playing a matching-penny game between new players before entering the arena G. The new arena for the game is depicted in Figure 8. While one player will play in the left part (containing G), a new player will play in the right part. The former player will aim at matching the pennies or reaching his objective in G, while the second player will try to not match the pennies. If there is a Nash equilibrium in the resulting game, both players must win, since otherwise they can change their strategy and improve their payoff. In that equilibrium, it must be the case that the former player satisfies his objective in G. This is formalised below.

We define the new game network $\mathcal{G}'$ with $2n$ players as $\mathcal{G}' = \langle G', (\equiv'_i)_{i\in[2n]}, (\Omega'_i)_{i\in[2n]} \rangle$ defined below:

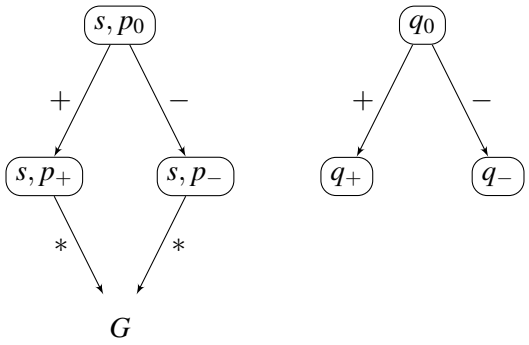

Figure 8: The player arena G'

- $G' = (\mathsf{States}', \{A\}, \mathsf{Act}', \mathsf{Mov}', \mathsf{Tab}')$ where:
  - $\mathsf{States}' = \mathsf{States} \cup (\mathsf{States} \times \{p_0, p_+, p_-\}) \cup \{q_0, q_+, q_-\}$. The last three states constitute the *isolated part* of the arena, while the other states are in the *main part*;
  - $\mathsf{Act}' = \mathsf{Act} \cup \{+, -, *\}$;
  - for every $s \in \mathsf{States}$, $\mathsf{Mov}'(s) = \mathsf{Mov}(s)$
    $\mathsf{Mov}'(s, p_0) = \{+, -\}$
    $\mathsf{Mov}'(s, p_+) = \{*\}$
    $\mathsf{Mov}'(s, p_-) = \{*\}$
    $\mathsf{Mov}'(q_0) = \{+, -\}$ and $\mathsf{Mov}'(q_+) = \mathsf{Mov}'(q_-) = \emptyset$;
  - for every $s \in \mathsf{States}$,
    $\mathsf{Tab}'(s, a) = \mathsf{Tab}(s, a)$ for every $a \in \mathsf{Mov}'(s)$
    $\mathsf{Tab}'((s, p_0), +) = (s, p_+)$
    $\mathsf{Tab}'((s, p_0), -) = (s, p_-)$
    $\mathsf{Tab}'((s, p_+), *) = \mathsf{Tab}'((s, p_-), *) = s$
    $\mathsf{Tab}'(q_0, +) = q_+$ and $\mathsf{Tab}'(q_0, -) = q_-$.

  In G', states $\{q_0, q_+, q_-\}$ are called the isolated part, whereas the rest of the arena is called the main part.

- For every configuration $t = (s_0, \ldots, s_{2n-1})$, we define

$$\mathit{main}(t) = \{j \in [2n] \mid s_j \text{ is in the main part of the arena}\}.$$

  We extend *main* to runs in a straightforward way. For every $i \in [2n]$, we define $\eta(i) = n + i \mod 2n$. Then, for every $i \in [2n]$, $t \equiv_i' t'$ iff the following conditions are satisfied:
  - $\mathit{main}(t) = \mathit{main}(t')$
  - if the above conditions hold, there is a bijection $\pi : \mathit{main}(t) \to [n]$ such that $\pi(j) \in \{j, \eta(j)\}$ for every $j \in \mathit{main}(t)$ such that $t(\pi^{-1}) \equiv_i t'(\pi^{-1})$. Note that this only holds under the conditions that $|\mathit{main}(t)| = |\mathit{main}(t')| = n$, and either $i \in \mathit{main}(t)$ xor $\eta(i) \in \mathit{main}(t)$.

- Let $i \in [2n]$ and Γ_i^{isol} be the set of runs ρ such that:
  - player i plays in the isolated part (that is, $i \notin \mathit{main}(\rho)$)
  - player $\eta(i)$ plays in the main part (that is, $\eta(i) \in \mathit{main}(\rho)$)

– players i and $\eta(i)$ realise their matching penny (that is, they visit states (s,p_+) and q_+, or states (s,p_-) and q_-, along ρ)

Let Γ_i^{main} be the counterpart in the main component. In particular, we notice that $\Gamma_i^{\text{main}} = \Gamma_{\eta(i)}^{\text{isol}}$.

Then, a run ρ of $\mathscr{G}'$ will belong to $\widetilde{\Omega}_i$ whenever there exists a bijection $\pi : \mathit{main}(\rho) \to [n]$ such that $\pi(j) \in \{j, \eta(j)\}$ for every $j \in \mathit{main}(\rho)$, and $\rho[\pi]$ belongs to Ω_i, where $\rho[\pi]$ is the run obtained after having projected ρ on the players $\pi(\mathit{main}(\rho))$ and having removed the first two states of ρ.

Finally, for every $i \in [2n]$, we define the objective Ω'_i for player i as $\widetilde{\Omega}_i \cup \overline{\Gamma_i^{\text{main}}} \cup \Gamma_i^{\text{isol}}$

We now define for every $i,j,k \in [n]$, the permutations $\pi'_{i,j}$ as follows:

- $\pi'_{i,j}(k) = \pi_{i,j}(k)$ and $\pi'_{i,j}(\eta(k)) = \eta(\pi_{i,j}(k))$,
- $\pi'_{i,\eta(j)}(k) = \eta(\pi_{i,j}(k))$ and $\pi'_{i,\eta(j)}(\eta(k)) = \pi_{i,j}(k)$,
- $\pi'_{\eta(i),j}(k) = \eta(\pi_{i,j}(k))$ and $\pi'_{\eta(i),j}(\eta(k)) = \pi_{i,j}(k)$
- $\pi'_{\eta(i),\eta(j)}(k) = \eta(\pi_{i,j}(k))$ and $\pi'_{\eta(i),\eta(j)}(\eta(k)) = \eta(\pi_{i,j}(k))$

It is easy to verify that $(\pi'_{i,j})_{i,j\in[2n]}$ is a symmetric representation for game $\mathscr{G}'$. We should also note that if $\mathscr{G}$ has a compact representation, then so has $\mathscr{G}'$. Furthermore, if objectives $(\Omega_i)_{i\in[n]}$ are given by LTL formulas, then objectives $(\Omega'_i)_{i\in[2n]}$ can also be given by LTL formulas of same size.

Lemma 26 *There is a (symmetric) Nash equilibrium in $\mathscr{G}$ from configuration $t_0 = (s_0^0, \dots, s_0^{n-1})$ with payoff 1 for every player iff there is a (symmetric) Nash equilibrium in $\mathscr{G}'$ from configuration $t'_0 = ((s_0^0, p_0), \dots, (s_0^{n-1}, p_0), q_0, \dots, q_0)$.*

Proof. Assume there is a Nash equilibrium σ in $\mathscr{G}$ from t_0 with payoff 1 for everyone. We define the strategy profile σ' where everyone plays α from its initial state, and then for players $i \in [n]$ play according to σ (note that if σ is symmetric, then this new profile is also symmetric). For this new strategy profile, The payoff of each player is 1, this is therefore a Nash equilibrium.

Assume σ' is a Nash equilibrium in $\mathscr{G}'$. Each player $n+i$ can easily ensure $\Gamma_{n+i}^{\text{isol}}$ by swapping its action-choice (between α and β). Player $n+i$ has therefore already payoff 1, and the outcome of σ' belongs to set $\Gamma_{n+i}^{\text{isol}}$, and therefore to set Γ_i^{main}. Player i can easily ensure $\overline{\Gamma_i^{\text{main}}}$ (by swapping its first action), this means that the payoff of Player i is already 1 along the outcome of σ': the outcome of σ' belongs to $\widetilde{\Omega}_i$ since it does not belong to $\overline{\Gamma_i^{\text{main}}} \cup \Gamma_i^{\text{isol}}$. The strategy profile σ is then just the part of σ' after the matching-pennies, and restricted to the n first players. The outcome is then in Ω_i. We then easily get that σ is a Nash equilibrium in $\mathscr{G}$ with payoff 1 to everyone. Note that if σ' is symmetric, then so is σ.

□

□

2.3.4 Bounded-memory Nash equilibria

In this section we focus on bounded-memory strategies, proving that deciding the existence of bounded-memory symmetric Nash equilibria with LTL objectives is EXPSPACE-complete. We first notice that the EXPSPACE-hardness results are direct consequences of the proof of Theorem 19 (the only difference is that we restrict to a Turing machine using exponential space).

We now explain our algorithms for deciding the (constrained) existence of symmetric Nash equilibria requiring only finite memory. We begin with the case of memoryless equilibria, for which the algorithm is

as follows: it first guesses a memoryless strategy for one player, from which it deduces the strategy to be played by the other players. It then looks for the players that are losing, and checks if they alone can improve their payoff. If they cannot improve the guessed strategy yields a Nash equilibrium, otherwise it does not yield an equilibrium.

We fix a symmetric game network $\mathscr{G} = \langle G, (\equiv_{i\in[n]}), (\Omega_i)_{i\in[n]}\rangle$ with symmetric representation $(\pi_{i,j})_{i.j\in[n]}$. We assume that each objective Ω_i is given by an LTL formula ϕ_i.

The first step is to guess and store an $\equiv_0$-realisable memoryless strategy σ_0 for player 0, which we then prove witnesses the existence of a symmetric Nash equilibrium. Such a strategy is a mapping from States^n to Act. We intend player 0 to play according to σ_0, and any player i to play according to $\sigma_0(\pi_{i,0}^{-1}(s_0,...,s_{n-1}))$ in state $(s_0,...,s_{n-1})$. From Lemma 14 we know that all symmetric memoryless strategy profiles can be characterized by such an $\equiv_0$-realisable memoryless strategy for player 0.

The algorithm then guesses a set W of players (which satisfies the given constraint), and checks that under the strategy profile computed above, the players in W achieve their objectives while the players not in W do not. This is achieved by computing the non-deterministic Büchi automata for ϕ_i if $i \in W$ and for $\neg\phi_i$ if $i \notin W$, and checking that the outcome of the strategy profile above (which is a lasso-shaped path and can easily be computed from strategy σ_0) is accepted by all those automata.

It remains to check that the players not in W cannot win if they deviate from their assigned strategy. For each player i not in W, we build the one-player game where all players but player i play according to the selected strategy profile. The resulting automaton contains all the plays that can be obtained by a deviation of player i. It just remains to check that there is no path satisfying ϕ_i in that automaton. If this is true for all players not in W, then the selected strategy σ_0 gives rise to a memoryless symmetric Nash equilibrium.

Regarding (space) complexity, storing the guessed strategy requires space $O(|\mathsf{States}|^n)$. The Büchi automata have size exponential in the size of the formulas, but can be handled on-the-fly using classical constructions, so that the algorithm only requires polynomial space in the size of the formula. The lasso-shaped outcome, as well as the automata representing the deviations of the losing players, have size $O(|\mathsf{States}|^n)$, but can also be handled on-the-fly. In the end, the whole algorithm runs in exponential space in the number of players, and polynomial in the size of the game and in the size of the LTL formulas.

Theorem 27 *Deciding the (positive, constrained) existence of a memoryless symmetric Nash equilibrium for LTL objectives in symmetric game networks is* EXPSPACE*-complete.*

The above algorithm can be lifted to bounded-memory strategies: given a memory bound m, it guesses a strategy σ_0 using memory m, and does the same computations as above. Storing the strategy now requires space $O(m \cdot |\mathsf{States}|^n)$, which is still exponential, even if m is given in binary.

Theorem 28 *Deciding the (positive, constrained) existence of a bounded-memory symmetric Nash equilibrium for LTL objectives in symmetric game networks is* EXPSPACE*-complete.*

Remark 29 *Notice that the algorithms above could be adapted to handle non-symmetric equilibria in non-symmetric game networks: it would just guess all the strategies, the payoff, and check the satisfaction of the LTL objectives in the product automaton obtained by applying the strategies.*

The algorithm could also be adapted, still with the same complexity, to handle richer objectives, in particular in the semi-quantitative setting of [3], where the players have several (pre)ordered objectives. Instead of guessing the set of winners, the algorithm would guess, for each player, which objectives are satisfied, and check that no individual improvement is possible. The latter can be achieved by listing all possible improvements and checking that none of them can be reached.

2.3.5 General strategies

We already mentioned an undecidability proof in Theorem 18 for two players and perfect information when general strategies are allowed. However, the objectives used for achieving the reduction are quite complex. On the other hand, imperfect information also leads to undecidability for LTL objectives with only 3 players. To show this, we can slightly alter a proof from [15]. Here, synthesis of distributed reactive systems (corresponding to finding sure-winning strategies) with LTL objectives is shown undecidable in the presence of imperfect information. The situation used in the proof can be modelled in our framework and by adding a matching-penny module in the beginning and slightly changing the LTL objectives, we can obtain undecidability of Nash equilibria instead of sure-winning strategies.

Theorem 30 *Deciding the existence of a (symmetric) Nash equilibrium for LTL objectives in symmetric game networks is undecidable for $n \geq 3$ players.*

3 Conclusion

In this paper, we have proposed a model of games for large networks of identical devices. This model of games is composed of a single arena, which is duplicated (one copy for each player), and each player has only a partial information on the whole state-space of the system. To fully represent large networks of identical devices, we added symmetry constraints, which yields non-local constraints in the system.

For this model, we have studied several problems related to the computation of symmetric Nash equilibria in such games. We have fully characterized the complexity of the (constrained) existence problem for bounded-memory strategies, and we have proven several undecidability results when the memory of the strategies is unbounded.

This work opens many interesting directions of research. Besides solving the questions left open in this paper, these directions include the study of Nash equilibria in games of imperfect information, and the extension to randomised strategies.

References

[1] Parosh Aziz Abdulla & Bengt Jonsson (1999): *On the Existence of Network Invariants for Verifying Parameterized Systems*. In: *Correct System Design, Recent Insight and Advances*, *Lecture Notes in Computer Science* 1710, Springer-Verlag, pp. 180–197, doi:http://dx.doi.org/10.1007/3-540-48092-7_9.

[2] Christel Baier & Joost-Pieter Katoen (2008): *Principles of Model-Checking*. MIT Press.

[3] Patricia Bouyer, Romain Brenguier, Nicolas Markey & Michael Ummels (2012): *Concurrent games with ordered objectives*. In: *Proc. 15th International Conference on Foundations of Software Science and Computation Structure (FoSSaCS'12)*, *Lecture Notes in Computer Science* 7213, Springer-Verlag, pp. 301–315, doi:http://dx.doi.org/10.1007/978-3-642-28729-9_20.

[4] Patricia Bouyer, Nicolas Markey & Steen Vester (2014): *Nash Equilibria in Symmetric Games with Partial Observation*. Technical Report LSV-14-01, Lab. Spécification & Vérification, ENS Cachan, France.

[5] Krishnendu Chatterjee, Thomas A. Henzinger & Nir Piterman (2007): *Strategy Logic*. In: *Proc. 18th International Conference on Concurrency Theory (CONCUR'07)*, *Lecture Notes in Computer Science* 4703, Springer-Verlag, pp. 59–73, doi:http://dx.doi.org/10.1007/978-3-540-74407-8_5.

[6] Krishnendu Chatterjee, Rupak Majumdar & Marcin Jurdziński (2004): *On Nash equilibria in stochastic games*. In: *Proc. 18th International Workshop on Computer Science Logic (CSL'04)*, *Lecture Notes in Computer Science* 3210, Springer-Verlag, pp. 26–40, doi:http://dx.doi.org/10.1007/978-3-540-30124-0_6.

[7] Arnaud Da Costa, François Laroussinie & Nicolas Markey (2010): *ATL with strategy contexts: Expressiveness and Model Checking*. In: *Proc. 30th Conferentce on Foundations of Software Technology and Theoretical*

Computer Science (FSTTCS'10), *Leibniz International Proceedings in Informatics* 8, Leibniz-Zentrum für Informatik, pp. 120–132, doi:http://dx.doi.org/10.4230/LIPIcs.FSTTCS.2010.120.

[8] Partha Dasgupta & Eric Maskin (1986): *The Existence of Equilibrium in Discontinuous Economic Games, 1: Theory*. *The Review of Economic Studies* 53(1), pp. 1–26.

[9] E. Allen Emerson & A. Prasad Sistla (1996): *Symmetry and model checking*. *Formal Methods in System Design* 9(1-2), pp. 105–131, doi:http://dx.doi.org/10.1007/BF00625970.

[10] Steven M. German & A. Prasad Sistla (1992): *Reasoning about Systems with Many Processes*. *Journal of the ACM* 39(3), pp. 675–735, doi:http://dx.doi.org/10.1145/146637.146681.

[11] Thomas A. Henzinger (2005): *Games in system design and verification*. In: *Proc. 10th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'05)*, pp. 1–4, doi:http://dx.doi.org/10.1145/1089933.1089935.

[12] Fabio Mogavero, Aniello Murano & Moshe Y. Vardi (2010): *Reasoning about strategies*. In Kamal Lodaya & Meena Mahajan, editors: *Proceedings of the 30th Conferentce on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'10)*, *Leibniz International Proceedings in Informatics* 8, Leibniz-Zentrum für Informatik, pp. 133–144.

[13] John F. Nash, Jr. (1950): *Equilibrium Points in n-Person Games*. *Proc. National Academy of Sciences* 36(1), pp. 48–49.

[14] John F. Nash, Jr. (1951): *Non-cooperative Games*. *Annals of Mathematics* 54(2), pp. 286–295.

[15] Amir Pnueli & Roni Rosner (1990): *Distributed Reactive Systems Are Hard to Synthesize*. In: *Proc. 31st Annual Symposium on Foundations of Computer Science (FOCS'90)*, IEEE Computer Society Press, pp. 746–757, doi:http://dx.doi.org/10.1109/FSCS.1990.89597.

[16] Michael Ummels & Dominik Wojtczak (2011): *The Complexity of Nash Equilibria in Stochastic Multiplayer Games*. *Logical Methods in Computer Science* 7(3:20), doi:http://dx.doi.org/10.2168/LMCS-7(3:20)2011.

A Symmetric Normal-Form Games

In this section, we question the definition of symmetric normal-form games introduced in [DM86] which is as follows:

Definition 31 *A symmetric normal-form game is a tuple $\langle [n], S, (u_i)_{i\in[n]}\rangle$ where $[n]$ is the set of players, S is a finite set of strategies and $u_i\colon S^n \to \mathbb{R}$ are utility functions such that for all strategy vectors $(a_0,...,a_{n-1}) \in S^n$, all permutations π of $[n]$ and all i it holds that*

$$u_i(a_0,...,a_{n-1}) = u_{\pi(i)}(a_{\pi(0)},...,a_{\pi(n-1)}).$$

In [DM86] and other sources where symmetric games are defined [BFH09, CRVW04, RJLB10, OR94, Pap07, SLB09] the basic intuition of what a symmetric normal-form game should be is a game where all players have the same set of actions and the same utility functions in the sense that the utility of player should depend exactly on which action he chooses himself and on the number of other players choosing any particular action. However, the definition above does not seem to capture the scenario at hand and might even be erroneous. For two players we agree with the definition, and indeed in [OR94, SLB09] it is only defined for two players. In [BFH09, CRVW04, RJLB10, Pap07] however it is defined such that $u_i(a_i,a_{-i}) = u_j(a_j,a_{-j})$ whenever $a_i = a_j$ and $a_{-i} = a_{-j}$. Here, a_i means the action taken by player i and a_{-i} means the set of actions taken by players other than player i. This definition seems to agree with the intuition about the games which we wish to represent. One argument against Definition 31 is that it has the following consequence

Theorem 32 *Using Definition 31 for games with more than two players, for all pairs of players i, j and all strategy profiles $(a_0,...,a_{n-1})$ we have*

$$u_i(a_0,...,a_{n-1}) = u_j(a_0,...,a_{n-1}).$$

In particular, for zero-sum games, we have $u_i(a_0,...,a_{n-1}) = 0$ for any player i and any strategy profiles $(a_0,...,a_{n-1})$.

Proof. We start by looking at games with $n \geq 3$. First, using the permutation $(1,2,0,...)$ we get

$$u_0(a_0,a_1,a_2,...) = u_1(a_1,a_2,a_0,...).$$

Next, using $(0,2,1,...)$ we get

$$u_2(a_1,a_0,a_2,...) = u_1(a_1,a_2,a_0,...).$$

Finally, using $(1,0,2,...)$ gives us

$$u_1(a_1,a_2,a_0,...) = u_0(a_2,a_1,a_0,...).$$

Putting all these equations together, we get $u_0(a_0,a_1,a_2,...) = u_0(a_2,a_1,a_0,...)$. This means that Player 0 can switch action with Player 2 without changing his utility. This can be done for arbitrary opponents by symmetry and since we can switch actions of all other players without changing the utility using permutations where $\pi(0) = 0$, we have that $u_0(a_0,...,a_{n-1}) = u_0(a_{\pi(0)},...,a_{\pi(n-1)})$ for every permutation π. By symmetry, for every i and every permutation π, we get

$$u_i(a_0,...,a_{n-1}) = u_i(a_{\pi(0)},...,a_{\pi(n-1)}).$$

Now consider any two players i and j as well as a permutation π such that $\pi(i) = j$. Then we have

$$\begin{aligned} u_i(a_0,...,a_{n-1}) &= u_{\pi(i)}(a_{\pi(0)},...,a_{\pi(n-1)}) \\ &= u_j(a_{\pi(0)},...,a_{\pi(n-1)}) \\ &= u_j(a_0,...,a_{n-1}) \end{aligned}$$

using the result above. This means that for any choice of actions of the players, every player will get the same utility. □

One could define a class of games with this property, however for the definition at hand the property is not true for games with two players, which is a quite strange property of a class of games.

In the following definition we propose a fix to the above definition. The resulting definition is equivalent to the definitions in [BFH09, CRVW04, RJLB10, Pap07], but it is given in a form resembling Definition 31. Note that this formulation has similarities with our own definition of a symmetric game network (Definition 7).

Definition 33 *A symmetric normal-form game is a tuple $\mathcal{N} = \langle [n], S, (u_i)_{i\in[n]} \rangle$ where $[n]$ is the set of players, S is a finite set of strategies and $u_i \colon S^n \to \mathbb{R}$ are utility functions such that for all strategy vectors $(a_0,...,a_{n-1}) \in S^n$, all permutations π of $[n]$ and all i it holds that*

$$u_i(a_0,...,a_{n-1}) = u_{\pi^{-1}(i)}(a_{\pi(0)},...,a_{\pi(n-1)}).$$

The intuition behind this definition is as follows. Suppose we have a strategy profile $\sigma = (a_0,...,a_{n-1})$ and a strategy profile where the actions of the players have been rearranged by permutation π, $\sigma_\pi = (a_{\pi(0)},...,a_{\pi(n-1)})$. We would prefer that player j, using the same action in σ_π as player i does in σ, gets the same utility. Since j uses $a_{\pi(j)}$, this means that $\pi(j) = i \Rightarrow j = \pi^{-1}(i)$. Now, from this intuition we have that $u_i(a_0,...,a_{n-1}) = u_j(a_{\pi(0)},...,a_{\pi(n-1)}) = u_{\pi^{-1}(i)}(a_{\pi(0)},...,a_{\pi(n-1)})$. Apart from this intuition, the new definition can be shown to be equivalent to the one from [BFH09, CRVW04, RJLB10, Pap07]. The reason that we agree with Definition 31 for two-player games is that $\pi = \pi^{-1}$ for all permutations π of two elements.

B Example: modeling a server-client architecture

We show how to model the server-client architecture in a symmetric way. Assume S represents the arena for the server, and C for the client. The arena G will be composed of three disconnected components: S, C and an extra state sink with a selfloop on it. We assume there are n clients. The game $\mathscr{G}$ is defined as the $(2n+1)$-players game network $\mathscr{G} = (G, (\equiv_i)_{i\in[2n+1]}, (\Omega_i)_{i\in[2n+1]})$ where:

- we define $N_S(0) = \{0,1,\ldots,n\}$, for every $1 \le i \le n$, we define $N_S(i) = \{0,i,n+1\ldots,2n-1\}$, and for every $n+1 \le i \le 2n$, we define $N_S(i) = \{0,n+1,\ldots,2n\}$
- we define $N_C(0) = \{0,n+1\}$, and for every $1 \le i \le 2n$, we define $N_C(i) = \{0,i\}$
- $(s_0,\ldots,s_{2n-2}) \equiv_i (s'_0,\ldots,s'_{2n-2})$ iff the following conditions hold:
  - s_i is an S-state iff s'_i is an S-state
  - s_i is a C-state iff s'_i is a C-state
  - if s_i is an S-state, then for every $j \in N_S(i)$, $s_j = s'_j$
  - if s_i is a C-state, then for every $j \in N_C(i)$, $s_j = s'_j$

This is a symmetric game network. A possible symmetric representation is: $\pi_{0,i}(0) = i$, $\pi_{0,i}(1) = i$ and $\pi_{0,i}(j) = n - j + 1$ if $1 \leq i \leq n$; $\pi_{0,i}(0) = i$, $\pi_{0,i}(1) = i$ and $\pi_{0,i}(2) = n+1$... $\pi_{0,i}(n) = 2n$ if $n+1 \leq i \leq 2n$.

Assuming that player 0 will play in S, players $1, \ldots, n$ will play in C, and all other players are trapped in sink, this properly model the server-client architecture since in that case:

- $\equiv_0$ reduces to $\mathsf{Id}([n])$
- $\equiv_i$ reduces to $\mathsf{Id}(\{0, i\})$ for every $1 \leq i \leq n$

With this modeling, we will be interested in symmetric Nash equilibria starting at configuration $(s_0, c_0, \ldots, c_0, \mathsf{sink}, \ldots, \mathsf{sink})$ where s_0 is the initial state of S and c_0 the initial state of C.

References

[BFH09] Felix Brandt, Felix Fischer, and Markus Holzer. Symmetries and the complexity of pure Nash equilibrium. *Journal of Computer and System Sciences*, 75(3):163–177, May 2009.

[CRVW04] Shih-Fen Cheng, Daniel M. Reeves, Yevgeniy Vorobeychik, and Michael P. Wellman. Notes on equilibria in symmetric games. In Simon Parsons and Piotr Gmytrasiewicz, editors, *Proceedings of the 6th International Workshop On Game Theoretic And Decision Theoretic Agents (GTDT'04)*, pages 71–78, New York, NY, USA, July 2004.

[DM86] Partha Dasgupta and Eric Maskin. The existence of equilibrium in discontinuous economic games, 1: theory. *The Review of Economic Studies*, 53(1):1–26, January 1986.

[OR94] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[Pap07] Christos Papadimitriou. The complexity of finding Nash equilibria. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani, editors, *Algorithmic Game Theory*, chapter 2, pages 29–51. Cambridge University Press, 2007.

[RJLB10] Christopher Thomas Ryan, Albert Xin Jiang, and Kevin Leyton-Brown. Symmetric games with piecewise linear utilities. In *Proceedings of the Behavioral and Quantitative Game Theory: Conference on Future Directions (BQGT'10)*, Newport Beach, CA, USA, 2010. ACM Press.

[SLB09] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems – Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.