Reasoning about social choice and games in monadic fixed-point logic

Ramit Das
IMSc (HBNI), Chennai, India
ramitd@imsc.res.in

R. Ramanujam
IMSc (HBNI), Chennai, India
jam@imsc.res.in

Sunil Simon
Department of CSE
IIT Kanpur, Kanpur, India
simon@cse.iitk.ac.in

The notion of a finite set of players producing a choice profile and a particular player having an incentive to deviate between two almost similar choice profiles (known as improvement for the particular player) are some common concepts when we approach game theory from a discrete lens to address computational issues of finding a pure strategy nash equilibrium. Based on these fundamental notions one can create a directed graph structure amongst the strategy profile space, and connect them up via improvement edges, we call this structure the *improvement graph*. We wish to be able to come up with a logical framework to reason about the improvement dynamics on such graphs. We suggest the usage of least fixed logics as a natural specification language for the concepts arising in the improvement dynamics. Even though this is the core essence of our paper, we play around with the representation on the model side to get as close to the short description of the game from an algorithmic or computational point of view and accordingly modify our logic as well. We try to extend the applicability from social network games to the social choice theory setting of fair allocations. We root our findings to the concrete setting, of a variant of polymatrix games, called *priority separable games*, a graphical game which has a succinct representation. We present hardness proofs of finding a nash equilibria in such a setting and also present expressibility results.

1 Introduction

A logical study of game theory aims at exposing the assumptions and reasoning that underlies the basic concepts of game theory. This involves the study of individual, rational, strategic decision making between presented alternatives (in the non-cooperative setting). One potential form of reasoning in such a situation is to envisage all possible strategic choices by others, consider one's own response to each, then others' response to it in their turn, and so on *ad infinitum*, with Nash equilibrium representing fixed-points of such an iteration..

Such reasoning, which we might call *improvement dynamics*, is similar to but distinct from rational decision making under uncertainty; it is also similar to but distinct from epistemic reasoning. The former is about optimization, selecting the 'best' option in light of one's information; the latter is about 'higher order information' involving information about others' information etc. Improvement dynamics intends to yield the same end results as these, but operates at a more operational, computational level, and reasoning about it can be seen as reasoning at the level of computations searching for equilibria. In this sense, logic is seen as a succinct language for describing computational structure, rather than as a deductive system of reasoning by agents. In spirit, the role of such logics is similar to that of logics in descriptive complexity theory. If we were to talk of the descriptive complexity of game theoretic equilibrium notions, it would need to account for the implicit improvement dynamics embedded in the solution concept.

Interestingly, several contexts in social choice theory embed such improvement dynamics as well. When we aggregate individual choices or preferences into social choices / preferences, or decide on social action (like resource allocation) based in individual preferences, once again we see implicit improvement dynamics. If a particular profile of voter preferences yields a specific electoral outcome, one can consider a voter announcing a revised (and altered) preference to force a different outcome. Two agents might exchange their allocated goods to move to a new allocation, if they perceive advantage in doing so. Again, these can be seen as offers and counter-offers, perhaps leading to an equilibrium, or not. Some of these situations involve individual improvements, some (like pairs of agents swapping goods) involve coalitions, but they have the same underlying computational structure.

In this paper, we suggest that a suitable language for reasoning about this computational structure underlying games and social choice contexts can be found with variants of logics based on *monadic fixed-point logic*. This is an extension of first order logic with monadic least fixed-point operators. In this, we follow the spirit of descriptive complexity, where extensions of first order logics describe complexity classes. Formulas offer concise descriptions of reasoning embedded in improvement dynamics.

Why bother? When we have a common language across contexts, we can employ a form of reasoning common in one (say normal form games) in another (say fair resource allocations) and thus transfer results and techniques. We show that the idea of improvement under swaps corresponds to certain form of strong equilibria and coalitional improvement in games. Dynamics in iterated voting again correpond to improvement dynamics in games. In such cases when the structures studied possess interesting properties such as the *finite improvement property* or *weak acyclicity* we get certificates of existence of equilibria. Interesting subclasses of games (such as *potential games*) possess such properties and by “transfer” we can look for similar subclasses in social choice contexts, and *vice versa*.

The choice of monadic fixed-point logic is also motivated by the fact that it admits an efficient model checking algorithm. Monadic least fixed point operator, iterating over subsets of strategy profiles, suffices for improvement dynamics. Counting can help us constrain paths succinctly: though counting is first order expressible, such expression would be prohibitively long.

One natural obstacle in reasoning about improvement dynamics is that the strategy space is exponential in the size of the game description. A natural question arises whether we can reason directly with game descriptions, and similarly in the other contexts, with the resource allocation system description, and so on. In this context, we study a subclass of *separable games with priorities*, where we can show some restricted validities in the logic: for instance, we can assert that acyclicity of the game presentation ensures existence of Nash equilibrium. The *separable games with priorities* let’s us restrict our attention from arbitrary general normal form games to a setting where the utilities are of a certain form (*pairwise separable*). This allows for a succinct representation of games.

We further progress, to strengthen our logic to account for arbitrary players in the model side. Each such changes made to the setting raise different complexity issues and we mention them in the passing.

Thus the contribution of this paper is modest and simple. The reasoning discussed is familiar, that of improvement dynamics in normal form games, and expressing this in monadic fixed-point logic with counting. We discuss improvement dynamics in different contexts, and present a model checking algorithm for the logic.

1.1 Related work

Various logical formalisms have been used in the literature to reason about games and strategies. Action indexed modal logics have often been used to analyse finite extensive form games where the game representation is interpreted as models of the logical language [8, 5, 6]. A dynamic logic framework can then be used to describe games and strategies in a compositional manner [30, 16, 31] and encode existence of equilibrium strategies [18].

Alternating temporal logic (ATL) [1] and its variants [19, 37, 12] constitute a popular framework to reason about strategic ability in games, especially infinite game structure defined by unfoldings of finite graphs. These formalisms are useful to analyse strategic ability in terms of existence of strategies satisfying certain properties (for example, winning strategies and equilibrium strategies). Some of these logical formalisms are also able to make assertions about partial specifications that strategies have to conform to in order to constitute a stable outcome.

In this work we suggest a framework to reason about the dynamics involved in iteratively updating strategies and to analyse the resulting convergence properties. [7] consider dynamics in reasoning about games in the same spirit as ours and describe it in fixed-point logic. But crucially, the dynamics is on iterated announcements of players' rationality, and belief revision in response to it. Moreover, they discuss extensive form games rather than normal form games. However, they do advocate the use of the fixed-point extension of first order logic for reasoning about games.

Monadic least fixed point logic (MLFP) is an extension of first-order logic which is well studied in finite model theory [35]. It is a restriction of first order logic with least fixed point in which only unary relation variables are allowed. MLFP is an expressive logic for which, on finite relational structures, model checking can be solved efficiently [13]. It is also known that MLFP is expressive enough to describe various interesting properties of games on finite graphs. MLFP can also naturally describe transitive closure of a binary relation which makes it an ideal logical framework to analyse the dynamics involved in updating strategies and its convergence properties. When α is a formula with one first order free variable, $C_x\alpha \leq k$ asserts that the number of elements in the domain satisfying α is at most k. Clearly, this is expressible in first order logic with equality, but at the expense of succinctness. In the literature on first order logic with arithmetical predicates [34], it is customary to consider two sorted structures to distinguish between domain elements and the countsWhen we deal with the *Improvement Graph* whose domain elements are always profiles, there is no need for such caution. In subsequent sections when we start focussing on *pairwise separable game with priorities* we take into consideration the different sorts in our disposal.

It is well known that a variety of contexts in the mathematical social sciences can be formulated in terms of improvement dynamics leading to equilibria (of some kind). Our observation here is that the deployment of the MLFP logic variants can help to unify algorithmic techniques as well as seek interesting validities (as we do in the context of separable games in this paper).

Admittedly when we present contexts as diverse as normal form games, allocations in social choice theory or voting rules, all in one uniform framework, we only get a broad-strokes description of the models, and the literature on these contexts vary widely in details. We hope to convince the reader that *a priori*, the MLFP logic variants have sufficient expressiveness to capture interesting variations. Our hope is to delineate the logical resources needed to express the variations, but that will require further work ahead.

1.2 Organisation of the Paper

We first present our results in the general setting of *Improvement Graphs*. We introduce the variant of MLFP, MLFPC. We present a theorem concerning the model checking algorithm for the logic. It turns out to be polynomial in the size of the model given. Although it sounds good, the problem lies in the fact that the *Improvement Graph* structure is not a succinct representation in terms of the number of players and strategies.

So the natural progression was in trying to capture properties where our logical formulas would be directly over the succinct game descriptions, thus we wanted to get rid of the explicit construction of the *Improvement Graph*. We introduce the concept of *pairwise separable games with priorities* inspired from the *Polymatrix Games*. We are able to show that deciding the existence of nash equilibrium for such games is *NP-complete*. We use MLFP as our logic here and fix the number of players. Since we formalise over the graphical game description, we need to have explicit access to the different sorts of variables denoting players, strategies and outcomes. We are able to show certain restricted validities in this setting.

In the above attempts we have the logical description fixed to a finite number of players. But, in the game theory literature we hardly find payoff functions dependent on the number of players. Hence it makes sense that we try to emulate our results over games with arbitrary number of players. We introduce second order set quantifiers over the player sets, a minimal extension from the previous logical extension of mlfp we were working with and go about doing the same exercise as before.

We finally show how the improvement dynamics as a core concept can also be lifted to the social choice theory setting.

2 The improvement graph structure

Improvement dynamics is a natural notion to study in the context of any situation involving strategic interaction of agents. In this section we formalise this dynamics in terms of the data structure called improvement graphs. We consider three specific application domains: strategic form games, voting theory and allocation of indivisible items. We show how improvement graphs can be interpreted in these applications and argue that the analysis of the structure acts as the basis for reasoning about strategic interaction.

Let $[n] = \{1, \dots, n\}$ denote the set of n agents. Each agent is associated with a finite set of choices S_i. A profile of choices (one for each agent) written $s = (s_1, \dots, s_n)$ induces an outcome of the strategic interaction. Let S denote the set of all choice profiles, O denote the set of all outcomes and $s(O)$ denote the outcome associated with the profile $s \in S$. Each agent $i \in [n]$ is associated with a preference ordering over the outcome set: $\preceq_i \subseteq (O \times O)$. This ordering induces a preference ordering over profiles as follows: for $s, s' \in S$ and $i \in [n]$, $s \preceq_i s'$ if $s(O) \preceq_i s'(O)$. For a choice profile $s = (s_1, \dots, s_n)$, we use the standard notation s_{-i} to denote the $n-1$ tuple arising from s in which the choice of agent i is removed.

The associated *improvement graph* is the directed graph $G = (V, E)$ where $V = S$ and $E \subseteq (V \times [n] \times V)$. We will denote the triple $(s, i, s') \in E$ by $s \rightarrow_i s'$. The edge relation E satisfies the condition: for $i \in [n]$ and $s, s' \in S$, we have $s \rightarrow_i s'$ if $s \prec_i s'$, $s_i \neq s'_i$ and $s_{-i} = s'_{-i}$, corresponding to unilateral deviation. An *improvement path* in G is a maximal sequence of profiles $s^1 s^2 \cdots$ such that for every $j > 0$ there is a player k_j such that $s^j \rightarrow_{k_j} s^{j+1}$. Note that here we use deviation by a single player to define the improvement graph. We could easily extend the definition to deviation by a subset of players, this interpretation might be more relevant in certain domains.

2.1 Strategic form games

A strategic form game is given by the tuple $T = ([n], \{S_i\}_{i \in [n]}, O, \lambda, \{\preceq_i\}_{i \in [n]})$ where the set of strategies S_i for agent $i \in [n]$ can be viewed as its set of choices. For $S = S_1 \times \cdots \times S_n$, the function $\lambda : S \rightarrow O$ associates an outcome to every strategy profile. In this paper, we consider only *finite* strategic form games. The notion of *best response* and *Nash equilibrium* are standard: s_i is best response to s_{-i} if for all $s'_i \in S_i$, $\lambda(s) \succeq_i \lambda(s'_i, s_{-i})$; s is a Nash equilibrium if for all $i \in [n]$, s_i is best response to s_{-i}. Existence of Nash equilibrium and computation of an equilibrium profile (when it exists) are important questions in the context of strategic form games.

Given a strategic form game T, let G_T denote the improvement graph associated with T (as defined above). Improvement paths in G_T correspond to maximal sequences of strategy profiles that arise by allowing players to make unilateral profitable deviations that result in improving their choice according to their preference ordering. We say that a game has the *finite improvement property* (FIP) if every improvement path in G_T is finite [27]. An improvement path, in which each k_j edge in the sequence is the best response of agent k_j to $s^{j-1}_{-k_j}$, is called a best response improvement path. We can analogously define the finite best response property (FBRP) if every best response improvement path is finite. FIP not only guarantees the existence of Nash equilibrium, but also ensures the stronger property that a decentralised local search mechanism convergences to a equilibrium outcome. Various natural classes of resource allocation games like congestion games [33], fair cost sharing games and restrictions of polymatrix games [2] are known to have the FIP.

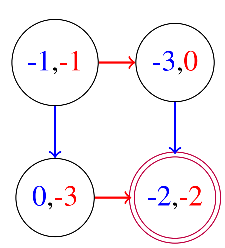
A weakening of FIP was proposed by Young [38] that asks only for the existence of *some* finite improvement path from any given strategy profile. Classes of strategic form games that satisfy this property are called weakly acyclic games. Note that weak acyclicity ensures that a randomised local search procedure almost surely converges to an equilibrium outcome [24]. Examples of classes of games which have this property include congestion games with player specific payoff functions [26], certain internet routing games [14] and network creation games [21].

As we can see, the improvement graph presents a data structure for analysing normal form games. It captures the epistemic reasoning underlying player choices: if I were to consider a particular profile of choices by all of us, I would rather choose another strategy to improve my payoff; in that case, agent j would revise her choice; and so on, unless we reach a profile from where none of us has any reason to deviate. Such reasoning is closely related to *pre-play negotiations* studied by game theorists.

2.2 Examples

We take the prisoner's dilemma game and show the improvement graph associated with it. We have two players 1 and 2 denoted by blue and red colours respectively. The improvement arrows of each player are also color coded by their respective colors. Strategy A is basically to maintain silence and not reveal the details, whereas strategy B is to betray the other guy involved.

		Player 2	
		Strategy A	Strategy B
Player 1	Strategy A	$(-1,-1)$	$(-3,0)$
	Strategy B	$(0,-3)$	$(-2,-2)$



Next we take the game of Rock-Paper-Scissors and show the improvement graph associated with it. This game doesn't have a pure strategy nash equilibrium. We preserve the same colors for each players. The strategies here are Rock (R), Paper (P) and Scissors (S).

		Player 2		
		R	P	S
Player 1	R	$(0,0)$	$(0,1)$	$(1,0)$
	P	$(1,0)$	$(0,0)$	$(0,1)$
	S	$(0,1)$	$(1,0)$	$(0,0)$

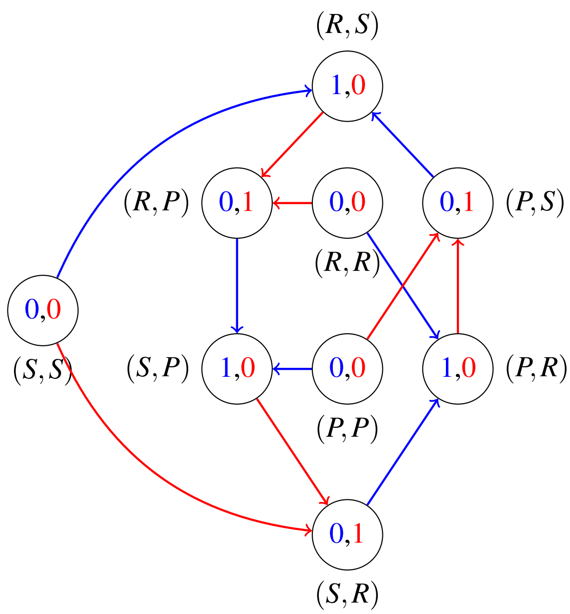


Figure 1: Improvement graph for the game of Rock-Paper-Scissors

3 Monadic fixed-point logic with counting

In this section we present the monadic fixed-point and counting extension of first order logic interpreted over improvement graphs. The use of the fixed-point extension is motivated by the fact that we wish to express properties like acyclicity of the graph, which is not first-order expressible. As we will see below, we need the fixed-point quantifier to range only over collections of nodes, and hence monadic fixed-point quantifiers suffice for our purpose. The counting extension helps us to count nodes in a subgraph, or along a path; this helps us express notions like fairness of schedules, which is of relevance in specifying improvement dynamics.

An alternative formalism would be the transitive closure extension of first order logic. But as Grohe has shown [17], monadic lfp logic is strictly less expressive than transitive closure logic, and hence we prefer a minimal extension of first order logic that serves our purposes. Note that the counting extension does not add expressivess but only succinctness. This is of use when we discuss concurrent deviation by a subset of players.

3.1 MLFPC Syntax

Let σ be a first order relational vocabulary. Let $(S_i)_{i\in\mathbb{N}}$ be a sequence of monadic relation symbols, such that for each i, $S_i \notin \sigma$. These are the second order fixed-point variables of the logic.

The set of all MLFPC formulas, Φ_{MLFPC}, is defined inductively as follows, where $fv^1(\alpha) =$ the set of first order free variables in α; $fv^2(\alpha) =$ the set of all relation symbols S_i occurring in α; $fv(\alpha) = fv^1(\alpha) \cup fv^2(\alpha)$.

- An MLFPC atomic formula if of the form $R(x_1,\dots,x_k)$, or $x = y$ or $S_i(x)$ where R is a k-ary relation symbol in σ, S_i is a second order fixed-point variable.
- If α,β are MLFPC formulas then so are, $\sim\alpha$, $\alpha\wedge\beta$ and $\alpha\vee\beta$.
  $fv(\sim\alpha) = fv(\alpha)$ and $fv(\alpha\wedge\beta) = fv(\alpha\vee\beta) = fv(\alpha)\cup fv(\beta)$.
- If α is a MLFPC formula, $x \in fv^1(\alpha)$ and $k\in\mathbb{N}$, then so are $\exists x\alpha$, $\forall x\alpha$ and $C_x\alpha \le k$.
  $fv(\exists x\alpha) = fv(\forall x\alpha) = fv(C_x\alpha \le k) = fv(\alpha)\setminus\{x\}$.
- If α is an MLFPC formula, $S_i \in fv^2(\alpha)$, $x\in fv^1(\alpha)$, and $u \notin fv^1(\alpha)$ and S_i occurs positively in α, then $[\mathbf{lfp}_{S_i,x}\ \alpha](u)$ is an MLFPC formula.
  $fv([\mathbf{lfp}_{S_i,x}\ \alpha](u)) = fv(\alpha)\setminus\{S_i,x\}\cup\{u\}$.

The restriction to positive second order variables in the lfp operator is essential to provide an effective semantics to the logic. It is a standard way of ensuring monotonicity, given that we do not have an effective procedure to test whether a given first-order formula is monotone on the class of finite σ-structures [23]. It should be noted that the use of positive second order variables in no way restricts us to contexts where equilibria are guaranteed to exist. Equilibria are given by graph properties, and these variables allow us to collect sets of vertices monotonically. Also note that in a formula $[\mathbf{lfp}_{S_i,x}\ \alpha](u)$ there can be free variables other than x (these are often called parameters). It is well known that it is possible to remove the parameters in the formula by increasing the arity of the fixed-point variables (see for instance, [29][Lemma 1.28]).

3.2 MLFPC Semantics

To interpret formulas, we extend σ-structures with interpretations for the free first order and second order variables (the latter from the given sequence $(S_i)_{i\in\mathbb{N}}$). Let $\mathfrak{A}$ be a σ-structure, which has domain A. The notion

$$\mathfrak{A}, u_1\mapsto a_1,\dots u_m\mapsto a_m, S_1\mapsto A_1,\dots S_n\mapsto A_n, \models \alpha(u_1,\dots,u_m,S_1,\dots,S_n)$$

is defined in the standard fashion, where $a_i\in A$ and $A_j\subseteq A$, $u_i\in fv^1(\alpha)$ and $S_j\in fv^2(\alpha)$. We abbreviate this by: $\mathfrak{A},\vec{a},\vec{A}\models\alpha(\vec{u},\vec{S})$. To simplify notation, we also use ρ to denote the interpretation, where, for every first order variable u, $\rho(u)\in A$ and for every second order variable S_i, $\rho(S_i)\subseteq A$. The semantics is then given as follows.

- $\mathfrak{A},\vec{a},\vec{A}\models R(x_1,\dots,x_k)$, iff the tuple $\vec{a}\in R^{\mathfrak{A}}$.
- $\mathfrak{A},\vec{a},\vec{A}\models S_j(x_i)$ iff $a_i\in A_j$.
- $\mathfrak{A},\vec{a},\vec{A}\models x_i = x_j$ iff $a_i = a_j$.

- $\mathfrak{A},\vec{a},\vec{A} \models \sim \alpha$ iff $\mathfrak{A},\vec{a},\vec{A} \not\models \alpha$.
- $\mathfrak{A},\vec{a},\vec{A} \models \alpha \vee \beta$ iff $\mathfrak{A},\vec{a},\vec{A} \models \alpha$ or $\mathfrak{A},\vec{a},\vec{A} \models \beta$.
- $\mathfrak{A},\vec{a},\vec{A} \models \exists x.\alpha$ iff for some $a \in A$, $\mathfrak{A},\vec{a},x \mapsto a,\vec{A} \models \alpha$.
- $\mathfrak{A},\vec{a},\vec{A} \models C_x\alpha \leq k$ iff $|\{a \in A \mid \mathfrak{A},\vec{a},x \mapsto a,\vec{A} \models \phi\}| \leq k$.
- $\mathfrak{A},\vec{a},u \mapsto a,\vec{A} \models [\mathbf{lfp}_{S_i,x}\alpha](u)$ iff $a \in \text{lfp}(f_\alpha)$ where for any formula β with $x \in fv^1(\beta)$, $f_\beta : \wp(A) \mapsto \wp(A)$ is defined by: $f_\beta(B) = \{a \in A \mid \mathfrak{A}, S_i \mapsto B, x \mapsto a \models \beta(x)\}$.

The lfp quantifier induces an operator on the powerset of elements on the structure ordered by inclusion. The positivity restriction ensures that the operator is monotone and hence least fixed-points exist.

3.3 Illustration

We illustrate a working example of how the lfp operator works. We present a formula on a particular directed graph in the example below and iterate over the fixed point computation highlighting the effects on the graph as well. The yellow color on the vertices are supposed to highlight the vertices chosen in a particular round of the fixed point computation. The green edges reaching out to further vertices denote the vertices that satisfy the formula and the red ones are to show when the formula isn't satisfied.

Given the graph below, Figure 2, we want to be computing whether $[\mathbf{lfp}_{\{S,x\}}\ \psi](c)$ and $[\mathbf{lfp}_{\{S,x\}}\ \psi](g)$ is true or false, where $\psi(S,x) = (x=a) \vee \exists y(S(y) \wedge (C_zE(y,z) \leq 2) \wedge E(y,x))$, given below as the sample formula! The fixed point operator f operates on a sequence of sets beginning from the $\emptyset$ and iterating over $f(\emptyset) \to f(f(\emptyset)) \to \ldots$ till it reaches a fixed point where $f^n(\emptyset) = f^{n+1}(\emptyset)$, which happens to be our fixed point that the induction operator S stores.

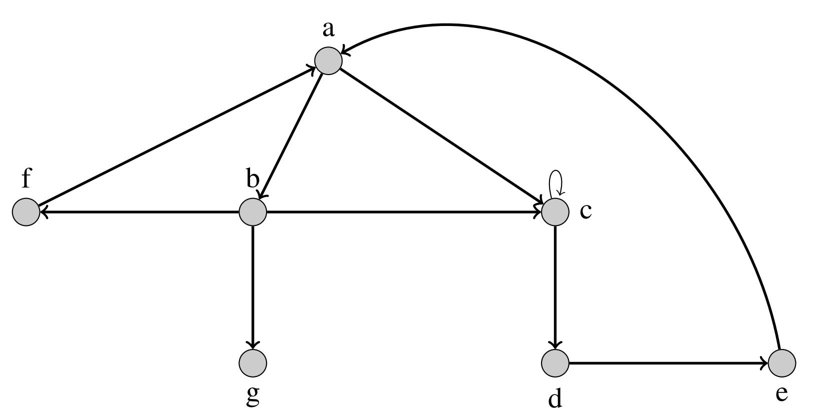
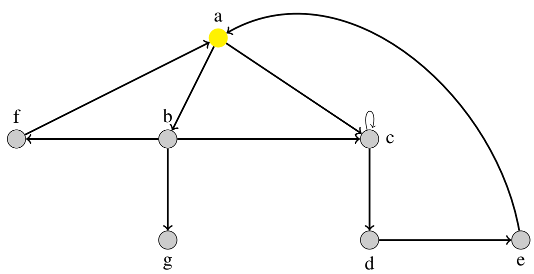


Figure 2: A Graph over which the LFP formula computation will be done

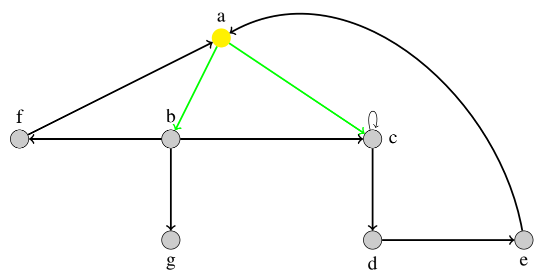
Sample formula over the directed graph

$$\psi(S,x) \ = \ (x=a) \vee \exists y(S(y) \wedge (C_zE(y,z) \leq 2) \wedge E(y,x))$$

The intended meaning of this formula is either $x=a$ or x is some vertex connected to a vertex already present in the set S which has the property that it is connected to at max 2 other vertices.



(a) The first iteration



(b) The second iteration

The first iteration

In the first iteration when S is mapped to the $\emptyset$, only $x = a$ is a solution to f_ψ. It is coloured yellow to mark it. Therefore, $f_\psi(\phi) = \{a\}$

(c)

Second iteration

In the second iteration we can see that apart from $x = a$, two other vertices connected to vertex a, ie, b, c are the solutions as well. So we have $f_\psi(\{a\}) = \{a, b, c\}$

(d)

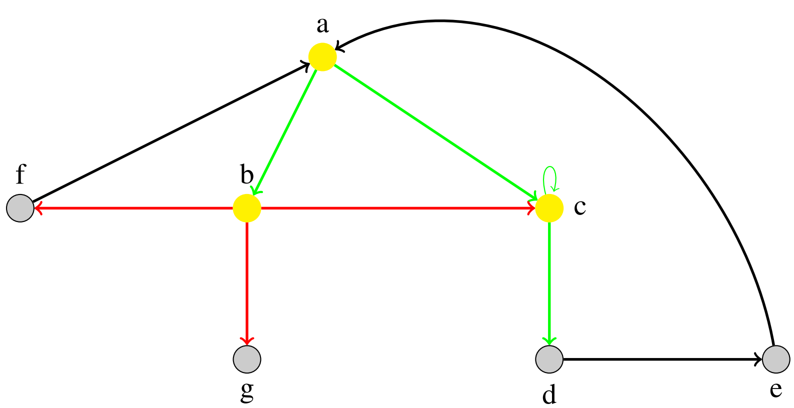
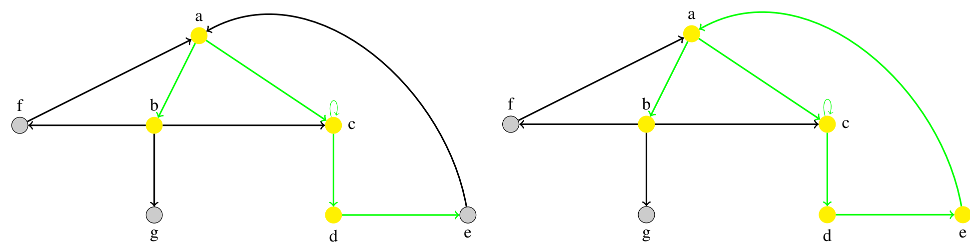


Figure 3: The third iteration

Computing the fixed point for the third iteration

In the third iteration we can see that apart from a and the solutions due to a, which are b and c, we have a new solution d due to the vertex c belonging to S. d is a vertex that is connected to c which belongs to S, and c has the property that it is connected to at most two other vertices, namely c and d. We also show that even though b is in S, it doesn't give us any new solution because it is connected to three vertices, highlighted with red edges. Therefore, $f_\psi(\{a, b, c\}) = \{a, b, c, d\}$



(a) The fourth iteration

(b) Fifth iteration - The fixed point is reached!

4th iteration

Similarly, by now we can see that in the fourth iteration we get a new vertex e due to d and thus, $f_{\psi}(\{a,b,c,d\}) = \{a,b,c,d,e\}$

(c)

Final fixed point

And, finally we reach the fixed point of the computation. $f_{\psi}(\{a,b,c,d,e\}) = \{a,b,c,d,e\}$

(d)

Sample formula

Therefore we can now answer the following lfp formulas.

- $[\mathbf{lfp}_{\{S,x\}}\ \psi](c)$ is true.
- $[\mathbf{lfp}_{\{S,x\}}\ \psi](g)$ is false.

3.4 Properties

Since the models of interest are improvement graphs, first order variables range over nodes in the graph, monadic second order variables range over subsets of nodes and the vocabulary consists of binary relations E_u, where $u \subseteq [n]$. When $|u| = 1$ and $u = \{i\}$, we will simply write the relation as E_i. We write formulas of the form $E(x,y)$ to denote $\bigvee_{i \in [n]} E_i(x,y)$.

We now write special formulas that will be of interest in the sequel.

- $sink(x) = \forall y. \sim E(x,y)$
- $trap(S,x) = \forall y.(E(y,x) \implies S(y))$
- $acyclic = \forall u.[\mathbf{lfp}_{S,x}\ trap](u)$
- $reach(S,x) = sink(x) \vee \exists y(E(y,x) \wedge S(y))$
- $weakly-acyclic = \forall u.[\mathbf{lfp}_{S,x}\ reach](u)$

Now consider the formulas interpreted over improvement graphs of normal form games. *sink* refers to the set of sink nodes, and these are exactly the Nash equilibria of the associated game. The sentence *acyclic* is true exactly when the improvement graph is acyclic and hence such games have the finite improvement property (as every improvement path is finite). To see that the sentence captures acyclicity, note the action of the lfp operator: at the zeroth iteration, we get all nodes with in-degree 0; we then get all nodes which have incoming edges from nodes whose in-degree is 0; and so on. Eventually it collects all nodes through which no path leads to a cycle. Since the sentence applies to every node, we infer that the graph does not contain any cycle.

For weak acyclicity, we require that there exists a finite improvement path starting from every node. Again the lfp operator picks up sink nodes at the zeroth iteration, then all nodes that have a sink node as successor, and so on. Eventually it collects all nodes that start finite improvement paths. The sentence asserts that every node has this property.

We can easily generalize these ideas to improvement graphs where the edge relation denotes deviation by a subset of players, rather than a single player. This leads us naturally to a *concurrent setting*, and we get what are called k-equilibria in *coordination games*.

Let s and s' be strategy profiles in a normal form game, $u \subseteq [n]$. Define $s \to_u s'$ when $s[j] = s'[j]$ for $j \notin u$, and for all $i \in u$, $s'[i] \succ_i s[i]$. That is, with the choices of the other agents fixed, the coalition of agents in u can coordinate their choices and deviate to get a better outcome. Thus k-equilibria are nodes from which no coalition of at most k agents can profitably deviate. We can then define a coalitional k improvement path, where at each step a coalition of at most k agents deviate, which leads us further to a k-FIP. [3] shows that a class of *uniform* coordination games has this property.

- $sink_u(x) = \forall y. \sim E_u(x,y)$
- $sink_k(x) = \bigvee_{u \subseteq [n], |u| \leq k} sink_u(x)$
- $k - edge(x,y) = \bigvee_{u \subseteq [n], |u| \leq k} E_u(x,y)$
- $k - trap(S,x) = \forall y.(k - edge(y,x) \implies S(y))$
- $k - FIP = \forall u.[\mathbf{lfp}_{S,x}\, k - trap](u)$

Note that the disjunctions are large, exponential in k. Since we have a counting operator, we could add further structure to nodes, prising out the individual strategies of players and then use the counting quantifier over these to get a succinct formula, linear in k.

In general, we might want to specify reachability of a set of distinguished nodes satisfying some property. For instance, we might want to assert that a particular node is reachable from any node. Note that the lfp operator is sufficient to specify reachability of such nodes.

- $reach_\phi(S,x) = \phi(x) \vee \exists y(E(y,x) \wedge S(y))$
- $\phi - reachable = \forall u.[\mathbf{lfp}_{S,x}\, reach](u)$

Further, the counting quantifier can give us interesting relaxations of improvement dynamics. For instance consider the following specifications:

- $reach(S,x) = sink(x) \vee \exists y(E(y,x) \wedge S(y))$
- $path - count = C_u([\mathbf{lfp}_{S,x}\, reach](u)) < 5$

This specifies that at most 5 nodes have finite improvement paths originating from them. Here the lfp operator is in the scope of the counting quantifier. In the following specification, we have them the other way about.

- $count - trap(S,x) = C_y E(y,x) < k \implies (\forall z.(E(z,x) \implies S(z)))$
- $special = \forall u.[\mathbf{lfp}_{S,x}\, count - trap](u)$

4 Model Checking Algorithm

The model checking problem for MLFPC is as follows: given a finite structure $\mathfrak{A}$, a MLFPC formula ϕ along with an interpretation ρ decide whether $\mathfrak{A}, \rho \models \phi$. We show that the recursive procedure given below solves the model checking problem for MLFPC efficiently (in the size of the structure).

MC($\mathfrak{A}, \rho, \phi$)

Input. An MLFPC formula ϕ, a σ structure $\mathfrak{A}$ and an interpretation ρ.
Output. If $\mathfrak{A}, \rho \models \phi$ then 1 and 0 otherwise.
switch *type of* ϕ

1. **case** ϕ is an atomic formula
   if $\mathfrak{A} \models \phi[\rho]$ **return** 1 **else return** 0
2. **case** $\phi = \sim \beta$
   if MC($\mathfrak{A}, \rho, \beta$) **return** 0 **else return** 1
3. **case** $\phi = \beta_1 \vee \beta_2$
   Let $\rho_1 := \rho|_{fv(\beta_1)}$, $\rho_2 := \rho|_{fv(\beta_2)}$
   if [MC($\mathfrak{A}, \rho_1, \beta_1$) or MC($\mathfrak{A}, \rho_2, \beta_2$)] **return** 1 **else return** 0
4. **case** $\phi = \exists y \beta$
   for $a \in A$
     if MC($\mathfrak{A}, \rho[x \mapsto a], \beta$) **return** 1
   return 0
5. **case** $\phi = C_x \beta \leq k$
   count $= 0$
   for $a \in A$
     if MC($\mathfrak{A}, \rho[x \mapsto a], \beta$) count ++
   if count $\leq k$ **return** 1 **else return** 0
6. **case** $\phi = [\mathbf{lfp}_{S_i,x}\ \beta](u)$
     iter = $\emptyset$, $f_\beta = \emptyset$
     do
       iter = f_β
       $f_\beta = \{b \in A \mid \text{MC}(\mathfrak{A}, \rho[S_i \mapsto f_\beta, x \mapsto b], \beta)\}$
     while $f_\beta \neq$ iter
     if $\rho(u) \in f_\beta$ **return** 1 **else return** 0

For a formula $\phi \in \Phi_{MLFPC}$, we denote by $cl(\phi)$, the set of all subformulae of ϕ and define the width of ϕ as follows: $width(\phi) = max\{|fv(\psi)| : \psi \in cl(\phi)\}$. The quantifier rank of ϕ (denoted $qr(\phi)$) is the maximal nesting depth of quantifiers in ϕ. Formally $qr(\phi)$ is defined as follows:

- If ϕ is atomic, then $qr(\phi) = 0$.
- $qr(\sim \phi) = qr(\phi)$.
- $qr(\phi_1 \vee \phi_2) = qr(\phi_1 \wedge \phi_2) = \max(qr(\phi_1), qr(\phi_2))$.
- $qr(\exists x \phi) = qr(\forall x \phi) = qr(C_x \phi) = qr([\mathbf{lfp}_{S_i,x}\phi](u)) = qr(\phi) + 1$.

Theorem 1. *Given $\phi \in \Phi_{MLFPC}$, a σ-structure $\mathfrak{A}$ and an interpretation ρ: $\mathfrak{A},\rho \models \phi$ iff algorithm MC returns 1 in time $|A|^{O(qr(\phi))} + O(|\phi|)$ and space $O(log|\phi| + width(\phi) log|A|)$.*

Proof. We first argue the correctness of the procedure by induction on the structure of the formula ϕ and then analyse the complexity.
Base case. When ϕ is an atomic formula, the procedure does a direct lookup given the structure and the interpretation. Correctness follows from the semantics.
Induction step.

- For boolean connectives, the correctness follows from induction hypothesis and the semantics.
- $\alpha \equiv \exists x\beta$: $\mathfrak{A},\sigma \models \exists x\beta$ iff by semantics, $\exists a \in A$ such that $\mathfrak{A},\rho[x \mapsto a] \models \alpha$. By induction hypothesis, for each $a \in A$, we have $\mathfrak{A},\rho[x \mapsto a] \models \beta$ iff MC($\mathfrak{A},\rho[x \mapsto a],\beta$) returns 1. By case 4, in MC it follows that the algorithm returns 1 iff $\exists a \in A$ such that $\mathfrak{A},\vec{a},x \mapsto a,\vec{A} \models \alpha$.
- $\phi \equiv C_x\beta \leq k$: $\mathfrak{A},\sigma \models C_x\beta \leq k$ iff by semantics, $|\{a \in A \mid \mathfrak{A},\rho[x \mapsto a] \models \beta\}| \leq k$. By induction hypothesis, for each $a \in A$, we have $\mathfrak{A},\rho[x \mapsto a] \models \beta$ iff MC($\mathfrak{A},\rho[x \mapsto a],\beta$) returns 1. In the algorithm MC, by definition of case 5, the algorithm maintains a local variable *count* which registers the number of different interpretations of variable x for which $\mathfrak{A},\rho[x \mapsto a] \models \beta$. Thus from the definition of the algorithm it follows that MC returns 1 iff $|\{a \in A \mid \mathfrak{A},\rho[x \mapsto a] \models \beta\}| \leq k$.
- $\phi \equiv [\mathbf{lfp}_{S_i,x}\ \beta](u)$: By semantics, $\mathfrak{A},\rho \models [\mathbf{lfp}_{S_i,x}\beta](u)$ iff $\rho(u) \in \text{lfp}(f_\beta)$ where $f_\beta(B) = \{a \in A \mid \mathfrak{A}, S_i \mapsto B, x \mapsto a \models\ \beta(x)\}$. By induction hypothesis, $\mathfrak{A},\rho[S_i \mapsto B, x \mapsto a] \models\ \beta(x)$ iff MC($\mathfrak{A},\rho[S_i \mapsto f_\beta, x \mapsto a],\beta$) returns 1. Now consider case 6, in the algorithm MC. The *do-while* loop computes the least fixed point $f_\beta\ : \wp(A) \mapsto \wp(A)$. The *if* conditional statement checks if $\rho(u) \in f_\beta$. Thus from the definition of the algorithm it follows that MC returns 1 iff $\mathfrak{A},\rho \models [\mathbf{lfp}_{S_i,x}\beta](u)$.

Complexity analysis. We analyse the running time by induction on the structure of the formula ϕ. For atomic formulas, we perform a direct lookup. Thus the time taken is $O(|A|^k)$ where k is the maximum arity of a predicate symbol in σ. The boolean connectives are straightforward. For $\phi \equiv \exists\psi$, $\phi \equiv \forall\psi$ and $\phi \equiv C_x\psi$, if $\mathfrak{A},\rho \models \psi$ can be decided in time $O(|A|^p)$ then $\mathfrak{A},\rho \models \phi$ can be decided in time $O(|A|^{p+1})$. And for $\phi \equiv \mathbf{lfp}_{S_i,x}\ \psi$, whether $\mathfrak{A},\rho \models \phi$ can be decided in time $O(|A|^{2p})$. Thus the total time taken is $|A|^{O(qr(\phi))} + O(|\phi|)$.

To bound the space required, note that algorithm needs to maintain a pointer to the current subformula of ϕ and to store the current interpretation, which needs $fv(\phi) \times \log|A|$ bits. Hence the space needed by the algorithm is $O(\log|\phi| + width(\phi)\log|A|)$. ☐

The above result implies that for formulas with bounded quantifier depth, the model checking procedure runs in time polynomial in both the size of the structure and the formula. All the formulas expressing interesting properties in games and social choice theory that we have presented in this paper have quantifier depth 1. Therefore, all these properties can be verified in polynomial time.

In the context of improvement graphs, if there are n agents and at most m choices for each agent, the size of the associated improvement graph is $O(m^n)$. Since it is possible to have a compact representation for certain subclasses of strategic form games, for instance, polymatrix games [20], the size of the improvement graph structure can be exponential in the representation of the game. Thus the model checking procedure, while polynomial on the size of the underlying improvement graph, can in principle, be exponential in the size of the game representation. This observation may not be very surprising since even for restricted classes of games like 0/1 polymatrix games, checking for the existence of Nash equilibrium is known to be NP-complete [4].

5 Pairwise separable games

While the improvement graph is a natural structure to reason about the dynamic properties of strategic interaction, explicitly representing the improvement graph has the disadvantage that the structure is exponential in the number of strategies in the underlying game. Reasoning about properties of games without explicitly constructing the improvement graph is therefore of obvious interest. Given that strategic form games constitute a rich model allowing arbitrary utility functions over strategy profiles, in general, the representation of a game can be exponential in the number of strategies. Thus a natural question is whether we can use the logical framework to analyse game which have compact representations directly using the game description rather than the improvement graph structure. In this section we identify a subclass of strategic form games where the utility functions are restricted to be pairwise separable. We show that the logical framework introduced in section 3 can be effectively used to analyse such games by interpreting formulas on the (compact) game description rather than the associated improvement graph.

Priority separable games. Let N be the set of players and $(S_i)_{i\in N}$ be the set of strategies for each player. Let O be a finite set of outcomes and for all $i \in N$, let $\ll_i \subseteq O \times O$ be a strict total ordering over the outcome set. Let $G = (N,E)$ be a directed graph (without self loops) and for each $i \in N$, let $R(i) = \{j \mid (j,i) \in E\}$ be the neighbourhood of i in G. We also associate a priority ordering within the neighbourhood for each node $i \in N$ and denote this by the relation $\triangleright_i \subseteq R(i) \times R(i)$. For $i, j \in N$, let $p_{i,j} : S_i \times S_j \to O$ be a partial payoff function. A priority separable game is specified by the tuple $\mathscr{G} = (G,(S_i)_{i\in N},O,(\ll_i)_{i\in N},(\triangleright_i)_{i\in N},(p_{i,j})_{i,j\in N})$. Observe that $\mathscr{G}$ has a compact representation that is polynomial in $|N|$, $\max_{i\in N}|S_i|$.

Given a strategy profile s, the payoff for player $i \in N$ is then defined as $p_i(s) = \times_{j\in R(i)} p_{i,j}$. Let $p^*_i(s)$ denote the reordering of the tuple $p_i(s)$ in decreasing order of the priority of neighbours of i. That is, if $R(i) = \{i_1,\dots,i_k\}$ and $i_1 \triangleright_i i_2 \triangleright_i \cdots \triangleright_i i_k$, then for $j \in \{1,\dots,k\}$, $(P^*_i(s))_j = p_{i,i_j}(s_i,s_{i_j})$. In order to analyse the strategic aspect of the game, we need to define how players compare between strategy profiles. For $i \in N$, we define the relation $\preceq_i \subseteq S \times S$ as follows: $s \preceq_i s'$ if $p^*_i(s) \preceq^{lex} p^*_i(s')$ where $\preceq^{lex}$ denotes the lexicographic ordering. All the basic notions in games that we introduced earlier, like better response, best response, improvement graph and Nash equilibrium remain the same and is based on the ordering of strategy profiles $(\preceq_i)_{i\in N}$.

Priority separable games form a qualitative subclass of *polymatrix games* [20], a class of strategic form games where payoffs are quantitative and pairwise separable. A strategic form game $(N,(S_i)_{i\in N},(u_i)_{i\in N})$ with $u_i : S \to \mathbb{R}$ is a polymatrix game if for every $i, j \in N$ there exist a partial utility function $u_{i,j}$ over the domain $S_i \times S_j$ such that for all $s \in S$, $u_i(s) = \Sigma_{j\in R(i)} u_{i,j}(s_i,s_j)$. Priority separable games are qualitative versions of strategic games with pairwise separable payoffs where players' preferences are defined using a lexicographic ordering over their local neighbourhoods. Note that priority separable games allow $\ll_i$ to be different for each player $i \in N$.

Example 1. Consider the game where $N = \{1,\dots,6\}$ and the graph G is as given in figure 4. For $i \in N$ the set of strategies S_i is specified in figure 4 as a label next to each node in G. Let $O = \{0,1\}$ with $0 \ll_i 1$ for all $i \in N$. For $i, j \in N$, let $p_{i,j} = 1$ if $s_i = s_j$ and $p_{i,j} = 0$ if $s_i \neq s_j$. Let $3 \triangleright_1 4$, $1 \triangleright_2 5$ and $2 \triangleright_3 6$. For $j \in \{4,5,6\}$, $|S_j| = 1$ and $R(j) = \emptyset$. Consider the strategy profile $s = (G,B,B,R,B,G)$ which is denoted with an underline in figure 4. Note that in s player 1 is not playing its best response and has a profitable deviation to R.

A natural question is whether priority separable games always have a pure Nash equilibrium. Below we show that the class of priority separable games need not always have a pure Nash equilibrium using an example which is similar to the one given in [4] for polymatrix games.

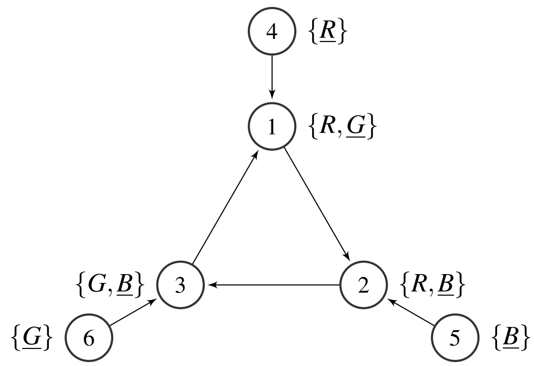


Figure 4: A priority separable game

Example 2. Consider the game given in example 1 along with the neighbourhood graph given in figure 4. For players $i \in \{4,5,6\}$, $R_i = \emptyset$ and since $|S_i| = 1$, for all $s \in S$, s_i is a best response to s_{-i}. Thus in each strategy profile s only the choices made by players $1,2$ and 3 are relevant. Below we enumerate all such strategy profiles and underline a strategy which is not a best response for each strategy profile. It then follows that this game does not have a Nash equilibrium. $(R,R,\underline{B})$, $(\underline{R},R,G)$, $(R,\underline{B},B)$, $(R,\underline{B},G)$, $(\underline{G},R,B)$, $(G,\underline{R},G)$, $(\underline{G},B,B)$, $(G,B,\underline{G})$.

Given that priority separable games need not always have a Nash equilibrium, an immediate question is whether there is an efficient procedure to check if a Nash equilibrium exists in this class of games. We show that checking for the existence of a Nash equilibrium is NP-complete. While the upper bound is straightforward, to show NP-hardness we give a reduction from 3-SAT using an argument similar to the one in [4].

Theorem 2. *Given a priority separable game $\mathcal{G}$, deciding if $\mathcal{G}$ has a Nash equilibrium is NP-complete.*

Proof. Given a priority separable game $\mathcal{G}$ and a strategy profile s, in polynomial time we can verify if s is a Nash equilibrium in $\mathcal{G}$. It follows that the problem is in NP. We show NP-hardness by giving a reduction from 3-SAT.

Suppose the instance of the 3-SAT formula is $\varphi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_k \vee b_k \vee c_k)$ with k clauses and m propositional variables $x_1, \ldots, x_m$. For $j \in \{1, \ldots, k\}$, a_j, b_j and c_j are literals of the form x_l or $\neg x_l$ for some $l \in \{1, \ldots, m\}$. We construct a priority separable game $\mathcal{G}_\varphi$ with the neighbourhood graph structure $G = (N,E)$ such that $\mathcal{G}_\varphi$ has a Nash equilibrium iff φ is satisfiable.

For every propositional variable x_l where $l \in \{1, \ldots, m\}$, we add a player X_l in $\mathcal{G}_\varphi$ with $S_{X_l} = \{\top, \bot\}$. With each clause $a_j \vee b_j \vee c_j$ for $j \in \{1, \ldots, k\}$, we associate 9 players whose neighbourhood is specified by the graph given in figure 5. The strategy set for each such node (or player) in the graph is specified as a label next to the node. We use x,y,z as variables where $x,y,z \in \{\top, \bot\}$ whose values are specified as part of the reduction. We denote this graph by $F_j(x,y,z)$ indicating that x,y and z are parameters.

For a literal d, let $\lambda(d) = \top$ if d is a positive literal and $\lambda(d) = \bot$ if d is a negative literal. For each clause with literals a_j, b_j and c_j, which is of the form x_l or $\neg x_l$, we add to $\mathcal{G}_\varphi$ the subgraph $F_j(\lambda(a_j), \lambda(b_j), \lambda(c_j))$ and an edge from X_l to the node A_j, B_j or C_j. Let $O = \{0,1\}$ with $0 \ll_i 1$ for all $i \in N$. For all $i,i' \in N$, we define $p_{i,i'} = 1$ if $s_i = s_{i'}$ and $p_{i,i'} = 0$ if $s_i \neq s_{i'}$. For each subgraph $F_j(x,y,z)$ corresponding to the clause $(a_j \vee b_j \vee c_j)$ and nodes A_j, B_j and C_j let $X_{l[A_j]}, X_{l[B_j]}, X_{l[C_j]}$ denote the nodes such that $(X_{l[A_j]}, A_j) \in E$, $(X_{l[B_j]}, B_j) \in E$ and $(X_{l[C_j]}, C_j) \in E$ respectively for $l[A_j], l[B_j], l[C_j] \in \{1, \ldots, m\}$. We specify the priority ordering for all players i with $|R(i)| > 1$ as follows. For each subgraph $F_j(x,y,z)$ we have,

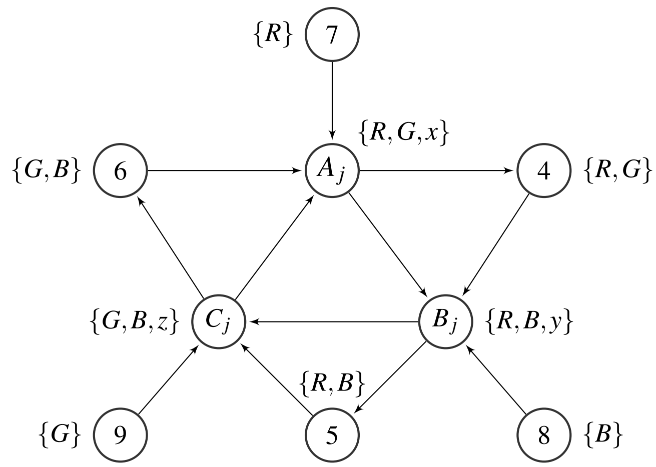


Figure 5: Gadget $F_j(x,y,z)$

- $X_{l[A_j]} \triangleright_{A_j} 6_j \triangleright_{A_j} 7_j \triangleright_{A_j} C_j$.
- $X_{l[B_j]} \triangleright_{B_j} 4_j \triangleright_{B_j} 8_j \triangleright_{B_j} A_j$.
- $X_{l[C_j]} \triangleright_{C_j} 5_j \triangleright_{C_j} 9_j \triangleright_{C_j} B_j$.

The crucial observation used in the reduction is the following. Consider the subgraph H_j induced by the nodes in $F_j(x,y,z)$ for $j \in \{1,\ldots,k\}$ along the nodes $X_{l[A_j]}, X_{l[B_j]}, X_{l[C_j]}$. Consider the priority separable game $\mathscr{G}(H_j)$ induced by nodes in H_j and the neighbourhood structure specified by H_j. Observe that a strategy profile t in $\mathscr{G}(H_j)$ is a Nash equilibrium iff at least one of the following conditions hold: $t_{A_j} = t_{X_{l[A_j]}}$ or $t_{B_j} = t_{X_{l[B_j]}}$ or $t_{B_j} = t_{X_{l[B_j]}}$. Using this observation, we can argue that $\mathscr{G}_\varphi$ has a Nash equilibrium iff φ is satisfiable.

Suppose s is a Nash equilibrium in $\mathscr{G}_\varphi$. Consider the valuation $v_s : \{x_1,\ldots,x_m\} \to \{\top,\bot\}$ defined as follows: $x_l = s_{X_l}$. From the observation above, it follows that for every $F_j(x,y,z)$ for $j \in \{1,\ldots,k\}$ at least one of the following conditions hold: $s_{A_j} = s_{X_{l[A_j]}}$ or $s_{B_j} = s_{X_{l[B_j]}}$ or $s_{B_j} = s_{X_{l[B_j]}}$. Assume without loss of generality that $s_{A_j} = s_{X_{l[A_j]}}$. By the definition of $\mathscr{G}_\varphi$, we have $S_{A_j} \cap S_{X_{l[A_j]}} = \{\lambda(a_j)\}$. By the definition of v_s we have $v_s(x_{l[A_j]}) = \lambda(a_j)$. This implies that $v_s \models a_j$ and therefore $v_s \models a_j \vee b_j \vee c_j$. Since this holds for all clauses, it follows that $v_s \models \varphi$.

Conversely, suppose φ is satisfiable and let $v \models \varphi$ for some valuation $v : \{x_1,\ldots,x_m\} \to \{\top,\bot\}$. Consider the partially defined strategy profile s^v where $s^v_{X_l} = v(x_l)$ for all $l \in \{1,\ldots,m\}$. Since $v \models \varphi$, for all clauses $a_j \vee b_j \vee c_j$, for $j \in \{1,\ldots,k\}$ we have $v \models a_j$ or $v \models b_j$ or $v \models c_j$. Without loss of generality suppose $v \models a_j$. By definition of $\mathscr{G}_\varphi$ we have $S_{A_j} \cap S_{X_{l[A_j]}} = \{\lambda(a_j)\}$. Therefore, the unique best response for node A_j in the game $\mathscr{G}_\varphi$ is the strategy $\lambda(a_j)$. This holds for all clauses and therefore, it is possible to extend s^v to a strategy profile which is a Nash equilibrium in $\mathscr{G}_\varphi$. $\square$

MLFPC and priority separable games. Instead of considering improvement graphs, we take the description of the priority separable game as the model along with a relational vocabulary for the logic. The underlying domain consists of three components: the player set N, the strategy set T and the outcome set O. In addition, we fix the following relations to interpret the game model:

- Monadic relation P which identifies the players.
- Monadic relation O which identifies the outcomes.

- Monadic relations $S_i \subset T$ indicating the strategy belonging to the particular player.
- Relation $\mathscr{S} \subseteq N \times T$ that specifies which strategy belongs to the strategy set of a player.
- Binary relation $E \subseteq N \times N$ that specifies the neighbourhood for each player.
- Relation $u \subseteq N \times T \times T \times O$ that specifies the pairwise payoffs.
- Binary relations $\ll_i$ specifying the ordering in the outcome set for each of the player.
- Binary relation $\triangleright_i$ that specifies the priority ordering for each player $i \in N$.

Let σ denote the language consisting the above vocabulary. We call a σ-structure *priority separable* if the interpretation on relational symbols confirm to the above definition. Thus there is a correspondence between priority separable games and priority separable σ-structures.

We first show that we can write a formula to characterise one step improvement of a player. We use $\vec{x}$ and $\vec{y}$ to denote tuple of strategies, $\vec{a}$ and $\vec{b}$ denotes the corresponding tuple of outcomes and p, v, w as player variables. We make use of the auxiliary formulas given below. Given two outcome vectors $\vec{a}$ and $\vec{b}$ we can compare the outcome for player i using the formulas $\psi_1(p,v,\vec{x},\vec{a},\vec{y},\vec{b})$ and $\psi_2(p,\vec{x},\vec{a},\vec{y},\vec{b})$.

- $\mathrm{chkStrat}(\vec{x}) = \bigwedge\limits_{j} S_j(x_j)$ - states that $\vec{x}$ is a valid strategy profile.
- $\mathrm{chkOut}(p,\vec{x},\vec{a}) = \bigvee\limits_{i\in N} (\mathscr{S}(p,x_i) \implies (\bigwedge\limits_{j\neq i} u(p,x_i,x_j,a_j)))$ - states that the outcome vector $\vec{a}$ is consistent with the pairwise utility relation u and the tuple of strategies $\vec{x}$.
- $\mathrm{first}(p,v,\vec{x}) = E(v,p) \wedge \bigvee\limits_{i\in N} (\mathscr{S}(p,x_i) \implies \forall w(E(w,p) \implies w \triangleright_i v))$ - states that $z \in R(p)$ is the player in the neighbourhood of p with the highest priority.
- $\mathrm{priority}_<(p,\vec{x},w,v) = \bigvee\limits_{i\in N} (\mathscr{S}(p,x_i) \implies w \triangleright_i v)$ - states that u, v are players in the neighbourhood of i and u has higher priority than v.
- $\mathrm{nxt}(p,w,v,\vec{x}) = \sim \exists v'(\mathrm{priority}_<(p,\vec{x},w,v') \wedge \mathrm{priority}_<(p,\vec{x},v',v))$ states that in the priority preference for player p the player w comes before v.
- $1\text{-step}(p,\vec{x},\vec{y}) = \mathrm{chkStrat}(\vec{x}) \wedge \mathrm{chkStrat}(\vec{y}) \wedge \bigvee\limits_{i\in[n]} (\mathscr{S}(p,x_i) \implies x_i \neq y_i \wedge \bigwedge\limits_{j\neq i} x_j = y_j)$ - states that $\vec{x}, \vec{y}$ are tuple of strategies and the only difference between them is in the strategy of player p.
- $\psi_1(p,u,\vec{x},\vec{a},\vec{y},\vec{b}) = \bigvee\limits_{i\in N} [(\mathscr{S}(p,x_i) \wedge \mathscr{S}(u,x_j)) \implies a_j \ll_i b_j]$.
- $\psi_2(p,\vec{x},\vec{a},\vec{y},\vec{b}) = \bigvee\limits_{i\in N} [\mathscr{S}(p,x_i) \implies a_i = b_i]$.

We can then write the formula $\mathrm{Dev}(p,\vec{x},\vec{a},\vec{y},\vec{b})$ that states that $\vec{y}$ is an improvement for player p from $\vec{x}$ by a 1-step deviation.

$$\begin{aligned}
\mathrm{Dev}(p,\vec{x},\vec{y}) = \exists \vec{a},\vec{b}\ (&\bigwedge_i (O(a_i) \wedge O(b_i)) \\
&\wedge\ 1\text{-step}(p,\vec{x},\vec{y}) \\
&\wedge \mathrm{chkOut}(p,\vec{x},\vec{a}) \wedge \mathrm{chkOut}(p,\vec{y},\vec{b}) \\
&\wedge \exists v([\mathbf{lfp}_{M,w}\alpha](v,p,\vec{x},\vec{a},\vec{y},\vec{b}) \\
&\quad \wedge \psi_1(p,v,\vec{x},\vec{a},\vec{y},\vec{b}))) \\
\alpha(M,w,p,\vec{x},\vec{a},\vec{y},\vec{b}) = {}& P(p) \wedge P(w) \wedge \mathrm{first}(p,w,\vec{x}) \\
&\vee \exists v[M(v) \wedge \mathrm{nxt}(p,w,v,\vec{x}) \\
&\quad \wedge \psi_2(v,\vec{x},\vec{a},\vec{y},\vec{b})]
\end{aligned} \tag{1}$$

Now, existence of Nash equilibrium can be characterised using the following formula:

- $\mathrm{Imp}(\vec{x},\vec{y}) = \exists p(P(p) \wedge (\ \exists\vec{b},\ \exists\vec{a}(\bigwedge_i (O(a_i) \wedge O(b_i))\ \wedge\ \mathrm{Dev}(p,\vec{x},\vec{a},\vec{y},\vec{b})))).$
- $\mathrm{NE}(\vec{x}) = \forall\vec{y} \sim \mathrm{Imp}(\vec{x},\vec{y}).$
- $\mathbb{G} \models \exists\vec{x}\,\mathrm{NE}(\vec{x})$

Restricted validities. It is easy to see that priority separable games need not always have a (pure) Nash equilibrium. Thus, a natural question is to identify subclasses where Nash equilibrium is guaranteed to exists. The main aspect behind priority separable games is the fact that the final utility for each player constitutes a combination of the outcomes in each pairwise interaction. Thus to identify interesting subclasses, we can restrict both the graphical structure that represents the neighbourhood relation as well as the pairwise utility relation. In this section, we show that the logical framework can be effectively used to describe such restrictions. We show that both properties describing the neighbourhood relation as well as restrictions on the pairwise utility relation can be characterised over the class of priority separable σ-structures. Below we provide a few formulas characterising properties of the neighbourhood relation in the underlying game.

- $\mathrm{trap}(M,w) = \forall v(P(v) \wedge P(w) \wedge E(v,w) \implies M(v)).$
- $\mathrm{acyclic} = \forall w(P(w) \wedge [\mathbf{lfp}_{M,u}\,\mathrm{trap}](w)).$ As we saw in section 3.4, this formula captures the property of acyclicity. By using the monadic relation P, we restrict the domain to the set of players. Thus this formula characterises the absence of a cycle in the player graph (N,E).
- $\mathrm{uniq}(p,v) = E(p,v) \wedge \forall w(P(w) \wedge E(p,w) \implies w = v)$ - characterises all nodes u which has a unique outgoing edge (to node v).
- $\mathrm{scycle} = \forall p \exists v \exists w(P(p) \wedge P(v) \wedge P(w) \wedge \mathrm{uniq}(p,v) \wedge \mathrm{uniq}(w,p))$ - states that the underlying player graph (N,E) is a simple cycle.

Some properties of the utility function that can be characterised in the logic are given below:

- $\mathrm{top}_i(o) = \forall a(O(a) \implies a \ll_i o)$ - states that outcome o is the most preferred outcome for player i.
- $\mathrm{bot}_i(o) = \forall a(O(a) \implies o \ll_i a)$ - states that outcome o is the least preferred outcome for player i.
- $\mathrm{coord}_i(\vec{x}) = \exists a \exists b(O(a) \wedge O(b) \wedge \mathrm{top}_i(a) \wedge \mathrm{bot}_i(b) \wedge S_i(x_i) \wedge$ $(\bigwedge_{j \neq i} ((E(j,i) \wedge x_i = x_j \implies u(i,x_i,x_j,a)) \wedge (\sim E(j,i) \vee x_i \neq x_j \implies u(i,x_i,x_j,b)))))$ - states that for each pairwise interaction, if player i coordinates its strategy with that of j (where $j \in R(i)$), then player i's utility is its most preferred outcome.
- $\mathrm{anticoord}_i(\vec{x}) = \exists a \exists b(O(a) \wedge O(b) \wedge \mathrm{top}_i(a) \wedge \mathrm{bot}_i(b) \wedge S_i(x_i) \wedge$ $(\bigwedge_{j \neq i} ((E(j,i) \wedge x_i \neq x_j \implies u(i,x_i,x_j,a)) \wedge (\sim E(j,i) \vee x_i = x_j \implies u(i,x_i,x_j,b)))))$ - captures anti-coordination. It states that if player i chooses a strategy different from its neighbour j then player i's utility in the pairwise interaction is its most preferred outcome.
- $\mathrm{coordall}_i = \forall\vec{x}(\mathrm{chkStrat}(\vec{x}) \implies \mathrm{coord}_i(\vec{x})).$
- $\mathrm{anticoordall}_i = \forall\vec{x}(\mathrm{chkStrat}(\vec{x}) \implies \mathrm{anticoord}_i(\vec{x})).$
- $\mathrm{coord}_G = \bigwedge_{j \in N} \mathrm{coordall}_j$ - states that the game is a coordination game.
- $\mathrm{anticoord}_G = \bigwedge_{j \in N} \mathrm{anticoordall}_j$ - states that the game is an anti-coordination game.

- $\text{dummy}_i = \bigwedge_{j\neq i} (\forall x_j \exists a \forall x_i (S_i(x_i) \wedge S_j(x_j) \wedge O(a) \implies u(i,x_i,x_j,a)))$ - states that the utility of player i is independent of his choice of strategy. That is, player i does not have any influence on his utility function in the pairwise interaction.

- $\text{dummy}_G = \bigwedge_{j\in N} \text{dummy}_j$ - states that the game is a dummy game for every player.

Theorem 3. *The following are validities over the class of priority separable σ-structures.*

1. $\varphi_1 = \mathit{acyclic} \implies \exists\vec{x}\ (\mathit{chkStrat}(\vec{x}) \wedge \mathit{NE}(\vec{x}))$.
2. $\varphi_2 = \mathit{scycle} \wedge \mathit{coord}_G \implies \exists\vec{x}(\mathit{chkStrat}(\vec{x}) \wedge \mathit{NE}(\vec{x}))$.

Proof. (1): Consider any structure $\mathfrak{A}$ such that $\mathfrak{A} \models \varphi_1$ (note that φ_1 is a sentence - it has no free variables) and suppose $\mathfrak{A} \models$ acyclic. By the argument given earlier, this implies that the player neighbourhood graph $G = (N,E)$ is acyclic. We need to show that there exists a tuple of strategies $\vec{x}$ such that $\forall\vec{y} \sim \text{Imp}(\vec{x},\vec{y})$. In other words, $\vec{x}$ is a tuple of strategies such that no player has an improvement from $\vec{x}$. We argue this by showing a stronger property: for priority separable games whose underlying player neighbourhood graph is acyclic, starting at any arbitrary tuple of strategies $\vec{z}$ there exists a finite improvement path terminating in the tuple of strategies $\vec{x}$ in the induced improvement graph (which implies that no player has an improvement from $\vec{x}$). Suppose the neighbourhood graph (N,E) is acyclic. Let λ be the ordering of players corresponding to the topological sorting of (N,E). Consider an arbitrary strategy profile $\vec{z}$, we construct a finite improvement path σ starting at $\vec{z}$ as follows: we use the ordering λ and update the strategy of players to their best response according to this ordering. Note that, by definition of λ, when a player j is chosen in the above procedure, it is guaranteed that the strategy s_k for all $k \in R(j)$ is defined. Clearly σ is finite by definition. Suppose the last strategy profile in σ is $\vec{x}$ and there is a player i and strategy profile $\vec{x}'$ such that $\mathfrak{A} \not\models \phi(p,\vec{x},\vec{a},\vec{x}',\vec{b})$. But this contradicts the fact that the choice of x_i was the best response when the strategy of player i was updated. Note that for all $j \in R(i)$, j occurs before i in the ordering λ.

(2): Consider any structure $\mathfrak{A}$ such that $\mathfrak{A} \models \varphi_2$ and suppose $\mathfrak{A} \models \text{scycle} \wedge \text{coord}_G$. This implies that $\mathfrak{A} \models \forall u \exists v \exists w (P(u) \wedge P(v) \wedge P(w) \wedge \text{uniq}(u,v) \wedge \text{uniq}(w,u))$. Therefore, for every player i in the player graph there is exactly one incoming and one outgoing edge. We also have that $\mathfrak{A} \models \text{coord}_G$ and therefore, for all players i, there is an outcome o such that $\text{top}_i(o)$ holds and for all $j \in R(i)$ utility of i in the pairwise interaction with j is o when $x_i = x_j$. Again, we show that in this case, starting at any arbitrary tuple of strategies $\vec{z}$ there exists a finite improvement path terminating in the tuple of strategies $\vec{x}$ in the induced improvement graph (which implies that no player has an improvement from $\vec{x}$). The argument is very similar to the one in [4][Lemma 6]. Since the underlying player graph forms a simple cycle, assume without loss of generality that it is of the form $1 \to 2 \cdots \to n-1 \to n \to 1$ (otherwise, we can simply relabel the players). Consider an arbitrary strategy profile $\vec{z}$, we construct a finite improvement path σ starting at $\vec{z}$ as follows: we perform two rounds of updates in the cyclic order starting at player 1. In the first round, we update players 1 to n in each step let them switch to their best response. Let x' be the strategy at the end of round 1, every player is playing its best response in x' except possible player 1 (since player n might have changed its strategy in round 1). In round two we let the player update again in the order $1,\ldots,n$. Note that in this round if a player updates then it update to the same strategy as its unique predecessor. Therefore player n does not update its strategy and we terminate in a strategy profile x such that $\mathfrak{A} \models \forall\vec{y} \sim \text{Imp}(\vec{x},\vec{y})$. $\square$

6 Generalising to Arbitrary Number of Players

We try to generalise our attempt at being able to logicise pure nash equilibria game theory. We now allow arbitrary number of players as part of our model and hence get closer to describing the algorithmic problem statements on these pure equilibria games. In this setting the number of players n becomes part of the input and isn't fixed like in the previous formalisations. Hence, now our model checking algorithm can now be shown to be NP-hard by Theorem 2 .

The rest of this section will work on the same priority separable games and elicit the same kind of formulas for Nash equilibria. And finally we will also remark on our attempt at finding a tight complexity bound for the model checking procedure.

6.1 Priority Separable Games Revamped

The definition of the priority separation games stays the same as in Section 5. What we need to do is instead change the vocabulary we worked with. The previous vocabulary basically needed the number of players fixed because depending on the number of players we had a bunch of relations indexed by each player. In the current treatment we will go away from this need to index to a situation where we can relate a player variable with the (appropriately modified) similar relations as present before. We also need to realise the strategy profiles are second order structures in such a setting. They are functions from the player set N to the union of all possible strategies.

Strategy Profile A sequence $(p, s_p)_{p\in N}$ where each player p chooses a strategy $s_p \in S_p$. We will represent strategy profiles as binary second order variables **X, Y, Z**.

Modifications in the Vocabulary from the previous one We present a minimally modified vocabulary. The underlying domain continues to contain these three components: the player set N, the strategy set T and the outcome set O. We have the following relations to interpret the game model.

- $\ll \subseteq N \times O \times O$ which basically is, $\ll (i, o_1, o_2) = \ll_i (o_1, o_2)$ when compared to the previous section 5 formulas. But while writing our formulas we will be using the syntactic sugar : $\ll_p$ as an infix operator. Example : $o_1 \ll_p o_2$ should be read as $\ll (p, o_1, o_2)$
- $\triangleright \subseteq N \times N \times N$ that specifies the priority ordering for each player $i \in N$. Basically it's $\triangleright(i, p_1, p_2) = \triangleright_i(p_1, p_2)$. Similar to the above scenario here also we will add the syntactic sugar $\triangleright_p$ as an infix operator such that $p_1 \triangleright_p p_2$ means $\triangleright(p, p_1, p_2)$.

Now we are set to write down the formulas in our current setting that would essentially be able to capture the nash equilibrium for a priority separable structure. Before that we just need to write down the syntax of the *extended* **MLFP** which varies slightly from the previous definitions.

6.2 Syntax of *extended* MLFP Logic

Let σ be a first order relational vocabulary. Let $(S_i)_{i\in\mathbb{N}}$ be a sequence of monadic relation symbols, such that for each i, $S_i \notin \sigma$. These are the second order fixed-point variables of the logic. Let $(\mathbf{X}_i)_{i\in\mathbb{N}}$ be a set of binary second order variables. We will usually want to express the other binary second order variables as **X,Y,Z**. These are supposed to represent the strategy profiles. Semantically, they are meant to capture $\mathbf{X} \subseteq N \times T$. Since we have three different types of domains, we usually represent them by three different types of variables - $p, u.v.w$ for player variables, o, a, b - for outcome variables and s, t - for strategy variables. The set of all formulas, $\Phi_{extended\ MLFP}$, is defined inductively as follows, where $fv^1(\alpha)$ = the set of first order free variables in α; $fv^2(\alpha)$ = the set of all relation symbols S_i occurring in α; $fv(\alpha) = fv^1(\alpha) \cup fv^2(\alpha)$.

- A **MLFP** atomic formula is of the form $R(x_1,\ldots,x_k)$, or $x=y$ or $S_i(x)$ or X(p,s) where R is a k-ary relation symbol in σ, S_i is a monadic second order fixed-point variable and where X_i, Y_i are second order variables denoting strategy profiles.
- If α,β are **MLFP** formulas then so are, $\sim\alpha$, $\alpha\wedge\beta$ and $\alpha\vee\beta$. $fv(\sim\alpha)=fv(\alpha)$ $fv(\alpha\wedge\beta)=fv(\alpha\vee\beta)=fv(\alpha)\cup fv(\beta)$.
- If α is a **MLFP** formula, $x\in fv^1(\alpha)$, then so are $\exists x\alpha$, $\forall x\alpha$ $fv(\exists x\alpha)=fv(\forall x\alpha)=fv(\alpha)\setminus\{x\}$.
- If α is a **MLFP** formula, $S_i\in fv^2(\alpha)$, $x\in fv^1(\alpha)$, and $u\notin fv^1(\alpha)$ and S_i occurs positively in α, then $[\mathbf{lfp}_{S_i,x}\ \alpha](u)$ is an **MLFP** formula. $fv([\mathbf{lfp}_{S_i,x}\ \alpha](u))=fv(\alpha)\setminus\{S_i,x\}\ \cup\{u\}$.
- Now that we have defined all the first order and the lfp operator quantification for the **MLFP** formulas.
  If α is a *extended* **MLFP** formula , then $\exists\vec{X}\alpha(\vec{X})$ is an *extended* **MLFP** formula as well.

6.3 Formulas to get to Nash Equilibrium

Let σ denote the *extended* **MLFP** language consisting of the above vocabulary. *Priority Separable structures* are the σ-structures where the interpretation on the relational symbols conform to the above definition.

- func(**X**) $=\forall p(P(p)\implies\sim\exists s_1,s_2(S(s_1)\wedge S(s_2)\wedge\mathbf{X}(p,s_1)\wedge\mathbf{X}(p,s_2)))$ - checks if **X** is a function from the player to strategy set.
- chkStr(**X**) $=\forall p\forall s(P(p)\wedge S(s)\wedge\mathbf{X}(p,s)\rightarrow\mathscr{S}(p,s))\wedge\text{func}(\mathbf{X})$ - checks if **X** is a strategy profile.
- first(p,z) $=E(z,p)\wedge\forall p_1(P(p_1)\wedge E(p_1,p)\wedge(z\triangleright_p p_1))$ - states that z is the first player is the priority ordering of player p.
- nxt(p,u,v) $=E(u,p)\wedge E(v,p)\wedge(u\triangleright_p v)\wedge(\sim\exists p_1(P(p_1)\wedge E(p_1,p)\wedge(u\triangleright_p p_1)\wedge(p_1\triangleright_p v)))$ - states that v is the immediate next player to player u among the priority ordering of player p.
- P-Pay$(p,u,\mathbf{X},o)=\exists s_1,s_2(S(s_1)\wedge S(s_2)\wedge\mathbf{X}(p,s_1)\wedge\mathbf{X}(u,s_2)\wedge u(p,s_1,s_2,o))$ - this captures the partial payoff o, of player p against player u in the strategy profile **X**.
- notEq(p,**X**,**Y**) $=\exists x,y(S(x)\wedge S(y)\wedge x\neq y\wedge X(p,x)\wedge Y(p,y))$ - states that the two strategy profiles **X** and **Y** differ at the coordinate index for player p.
- 1-step(p,**X**,**Y**) = chkStr(**X**) $\wedge$ chkStr(**Y**) $\wedge\,\text{notEq}(p,\mathbf{X},\mathbf{Y})\wedge\forall u(P(u)\wedge u\neq p\implies\exists z(S(z)\wedge\mathbf{X}(u,z)\wedge\mathbf{Y}(u,z))$ - states that player p has a 1-step improvement from **X** to **Y**.
- $\psi_1(p,u,\mathbf{X},\mathbf{Y})=\exists a,b(O(a)\wedge O(b)\wedge a\neq b\wedge\text{P-Pay}(p,u,\mathbf{X},a)\wedge\text{P-Pay}(p,u,\mathbf{Y},b)\wedge(a\ll_p b))$
- $\psi_1'(p,u,\mathbf{X},\mathbf{Y})=\exists a,b(O(a)\wedge O(b)\wedge a\neq b\wedge\text{P-Pay}(p,u,\mathbf{X},a)\wedge\text{P-Pay}(p,u,\mathbf{Y},b)\wedge(b\ll_p a))$
- $\psi_2(p,u,\mathbf{X},\mathbf{Y})=\exists a(O(a)\wedge\text{P-Pay}(p,u,\mathbf{X},a)\wedge\text{P-Pay}(p,u,\mathbf{Y},a))$

We can then write the formula Dev$(p,\mathbf{X},\mathbf{Y})$ that states that the strategy profile **Y** is an improvement for player p from the strategy profile **X** by a 1-step deviation.

$$\begin{aligned}\text{Dev}(p,\mathbf{X},\mathbf{Y}) &= \text{1-step}(p,\mathbf{X},\mathbf{Y})\\ &\quad\wedge\exists v\left(P(v)\wedge[\mathbf{lfp}_{M,w}\alpha](v,p,\mathbf{X},\mathbf{Y})\wedge\psi_1(p,v,\mathbf{X},\mathbf{Y})\right)\\ \alpha(M,w,p,\mathbf{X},\mathbf{Y}) &= P(p)\wedge P(w)\wedge\min(p,w)\\ &\quad\vee\exists v[M(v)\wedge\text{nxt}(p,v,w)\wedge\psi_2(p,v,\mathbf{X},\mathbf{Y})]\end{aligned}$$

Now, existence of Nash equilibrium can be characterised using the following formula:

- Imp(**X**,**Y**) $= \exists p(P(p) \wedge$ Dev(p,**X**,**Y**))
- NE(**X**) = $\forall$ **Y** $\sim$ Imp(**X**,**Y**)
- NE(**X**) = $\forall p \forall s\ \exists \mathbf{Y}\ (\ Y(p,s) \wedge$ 1-step(p,**X**,**Y**) $\wedge (\exists v \left(P(v) \wedge [\mathbf{lfp}_{M,w}\alpha](v,p,\mathbf{X},\mathbf{Y}) \wedge \psi_1'(p,v,\mathbf{X},\mathbf{Y})\right) \vee \forall v \left(P(v) \wedge E(v,p) \wedge [\mathbf{lfp}_{M,w}\alpha](v,p,\mathbf{X},\mathbf{Y})\right)))$
- $\mathbb{G} \models \exists$ **X** NE(**X**)

6.4 Remark on the Complexity

The model checking of the *extended* **MLFP** logic is in PSPACE. For the upper bound we can modify our model checking algorithm to tackle the case for $\phi = \exists Y \beta$. And, from a space complexity perspective, we would have to solve the following recurrence to show the algorithm is in PSPACE.

Recurrence for $Space(MC(\mathfrak{A},\rho,\phi))$

$$\leq \max \begin{pmatrix} O(1) + Space(MC(\mathfrak{A},\rho,\beta)), & \begin{cases} \text{for base case} \\ \text{for } \phi = \neg\beta \\ \text{for } \phi = \beta_1 \vee \beta_2 \end{cases} \\ O(\log|A|) + Space(MC(\mathfrak{A},\rho,\beta)), & \text{for } \phi = \exists y \beta \\ O(A^{arity(Y)}) + Space(MC(\mathfrak{A},\rho,\beta)), & \text{for } \phi = \exists Y \beta \\ O(1) + O(\log|A|) + Space(MC(\mathfrak{A},\rho,\beta)), & \text{for } \phi = C_x\beta \leq k \\ O(1) + O(|A|) + Space(MC(\mathfrak{A},\rho,\beta)), & \text{for } \phi = [\mathbf{lfp}_{S_i,x}\ \beta](u) \end{pmatrix}$$

The model checking problem of our logic is in PSPACE.
Induction Hypothesis :

$$\text{for all } \phi : Space(MC(\mathfrak{A},\rho,\phi)) \leq O(width(\phi) \times |A|^{max\ arity(\phi)})$$

Proof. **Base Case :**

i when ϕ is an atomic formula, it can be according to the algorithm checking in space $O(1)$

Induction Case :

i when $\phi := \alpha \vee \beta$ then,

$$Space(MC(\mathfrak{A},\rho,\phi) \leq O(1) + max(Space(MC(\mathfrak{A},\rho,\alpha)), Space(MC(\mathfrak{A},\rho,\beta)))$$

(because space can be reused). By our hypothesis, we have,

$$Space(MC(\mathfrak{A},\rho,\phi)) \leq O(1) + O(width(\phi) \times |A|^{max\ arity(\alpha)})$$

which we can rewrite as,

$$Space(MC(\mathfrak{A},\rho,\phi)) \leq O(1) + O(width(\phi) \times |A|^{max\ arity(\phi)})$$

$$Space(MC(\mathfrak{A},\rho,\phi)) \leq O(width(\phi) \times |A|^{max\ arity(\phi)})$$

ii when $\phi := \exists x\beta$ then,

$$
\begin{aligned}
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O(\log|A|) + Space(MC(\mathfrak{A},\rho,\beta))\\
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O(\log|A|) + O(width(\beta)\times|A|^{max\ arity(\beta)})\ \textbf{I.H.}\\
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O((width(\beta)+1)\times|A|^{max\ arity(\beta)})\\
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O(width(\phi)\times|A|^{max\ arity(\beta)})\\
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O(width(\phi)\times|A|^{max\ arity(\phi)})
\end{aligned}
$$

iii when $\phi := \exists Y\beta$ then,

$$
\begin{aligned}
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O(A^{arity(Y)}) + Space(MC(\mathfrak{A},\rho,\beta))\\
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O(A^{arity(Y)}) + O(width(\beta)\times|A|^{max\ arity(\beta)})\ \textbf{I.H.}\\
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O(A^{arity(Y)}) + O(width(\beta)\times|A|^{max\ arity(\phi)})\ \beta\ \textbf{is a subformula of}\ \phi\\
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O((width(\beta)+1)\times|A|^{max\ arity(\phi)})\\
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O(width(\phi)\times|A|^{max\ arity(\phi)})
\end{aligned}
$$

iv when $\phi := C_x\beta \leq k$ then, we can simply follow the reduction for case ii. and modify accordingly to get the hypothesis.

v when $\phi := [\mathbf{lfp}_{S_i,x}\ \beta](u)$ then,

$$
\begin{aligned}
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O(1) + O(A) + Space(MC(\mathfrak{A},\rho,\beta))\\
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O(1) + O(A) + O(width(\beta)\times|A|^{max\ arity(\beta)})\ \textbf{I.H.}\\
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O(A) + O(width(\beta)\times|A|^{max\ arity(\phi)})\\
Space(MC(\mathfrak{A},\rho,\phi)) &\leq O(width(\phi)\times|A|^{max\ arity(\phi)})
\end{aligned}
$$

□

For the lower bound, the expression complexity of FO is in PSPACE [23]. We will modify the reduction known to show PSPACE-hardness for the particular case of our logic.

The original proof for FO proceeds as, given QBF $\phi := Q_1Q_2\dots Q_n\alpha(x_1,x_2,\dots,x_n)$ where α is a propositional formula. Then we construct a structure and assume the domain $A = \{0,1\}$ and the unary relation $U = \{1\}$ and convert α into α^U where x_i instance is converted to $U(x_i)$ and $\neg x_i$ is converted $\neg U(x_i)$ to construct the formula α^U. And, then the following holds,

$$\phi \text{ is true iff } (A,U) \models Q_1Q_2\dots Q_n\alpha^U(x_1,x_2,\dots,x_n)$$

.

We show an alternate reduction which will capture the same essence.

Given QBF $\phi := Q_1Q_2\dots Q_n\alpha(x_1,x_2,\dots,x_n)$ where α is a propositional formula. Then we construct a structure and assume the domain $A = \{0,1,2\}$ where 2 serves as a filler element in some sense for our binary relation and $E = \{(1,2)\}$ be a binary relation, my conversion of α is as follows, x_i instance is converted to $E(x_i,2)$ and $\neg x_i$ is converted $\neg E(x_i,2)$ to construct the formula α^E, and we have the same theorem as above,

$$\phi \text{ is true iff } (A,E) \models Q_1Q_2\dots Q_n\alpha^E(x_1,x_2,\dots,x_n)$$

.

After this reformulation, we then have our lower bound reduction from ϕ a QBF formula, to the satisfiability problem of *extended* MLFP logic.

$$\phi \text{ is true iff } \langle (A,E,\{\}),\{\},(\{\},\{\}),\{\}\rangle \models Q_1Q_2\ldots Q_n\alpha^E(x_1,x_2,\ldots,x_n)$$

.

7 Improvement dynamics in social choice theory

7.1 Allocation of indivisible goods

An important problem often studied in economics and computer science is the allocation of resources among rational agents. This problem is fundamental and has practical implications in various applications including college admissions, organ exchange and spectrum assignment. In this paper, we consider the setting where there are $[n]$ agents and a set $A = \{a_1,\ldots,a_m\}$ of m indivisible items. An allocation $\pi : N \to 2^A$ such that $\cup_{i\in[n]}\pi(i) = A$ and for all $i, j \in [n]$, $i \neq j$, $\pi(i)\cap\pi(j) = \emptyset$. In the most general setting, each agent i has a preference ordering $\prec_i$ over the allocations. Thus an instance of an allocation problem can be specified as a tuple $H = ([n],A,\{\prec_i\}_{i\in[n]})$. Let Π denote the set of all allocations. In this setting, each allocation π can be viewed as defining an outcome and agents have a preference ordering over such outcomes.

There are natural notions of fairness in the context of allocation systems; for instance, we are interested in nodes that are *envy free*. An agent i envies an agent j at node x if there exists a node y such that $y \succ_i x$ and the allocation for i at y is the same as the allocation for j at x. A node is envy-free if no player envies another at that node. We might then want to assert that an envy free node is reachable from any node. Note that we only need to enrich the first order vocabulary to speak of $x[i],y[j]$ etc to express envy-freeness, and the lfp operator is sufficient to specify reachability of such nodes.

- $reach_\phi(S,x) = \phi(x) \vee \exists y(E(y,x)\wedge S(y))$
- $\phi - reachable = \forall u.[\mathbf{lfp}_{S,x}\ reach](u)$

When agents are allowed to exchange items with each other, stability of an allocation is a natural solution concept to study. Core stable outcomes are defined as allocations in which no group of agents have an incentive to exchange their items as part of an internal redistribution within the coalition. The improvement graph structure can capture the dynamics involved in such a sequence of item exchange in a natural manner. The associated improvement graph can be defined over the set of vertices Π. Since the deviation involves exchange of goods among a subset of players (rather than a unilateral deviation by a single player), the edge relation is indexed with a subset $u \subseteq [n]$. That is, for $\pi,\pi' \in \Pi$ and $u \subseteq [n]$, we have $\pi \to_u \pi'$ if for all $i \in u$, $\pi(i) \prec_i \pi'(i)$, $\pi(i) \neq \pi'(i)$ and for all $j \notin u$, $\pi(j) = \pi'(j)$.

Note that the improvement graph here is different from the ones we discussed earlier in a crucial sense. When agents in u swap goods, the allocation for other players outside $[u]$ is unaffected. If each agent's preference ordering depends only on the valuation of the bundle of items that the agent is allocated then their satisfaction is unchanged. However, agent 1 may swap goods with 2 and then use some the goods acquired to make a swap deal with 3 thus leading to interesting causal chains. In effect the entire space of allocations may be tentatively explored by the agents. A finite path in the improvement graph corresponds to a finite sequence of exchanges that converge to a stable outcome. An important question is whether stable allocations always exist and whether a finite sequence of exchanges can converge to such an allocation.

A common assumption that is made in allocation problems, is that the preference ordering for each agent i depends only on the bundle of items assigned to agent i. A special case of the above setting is when $n = m$ (i.e. the number of agents and the items are the same) and π is required to be a bijection. An instance of such an allocation problem A along with an initial allocation π_0 defines the well known Shapley-Scarf housing market [36].

When we allow exchange of goods in the housing market, it is known that a simple and efficient procedure termed as Gale's Top Trading Cycle can compute an allocation that is core stable. The allocation constructed in this manner also satisfies desirable properties like strategy-proofness and Pareto optimality.

7.1.1 Housing market with priority separable externalities

There are many instances where an agent's utility depends not only on the items that are allocated but also the allocation received by other agents - possibly within a social context. This is particularly natural in the housing market where the utility depends not just on the house that an agent is allocated but also on who is allocated the neighbouring houses. Formally, this implies that the utility functions depend on agent externalities. When these externalities are separable, we can use the framework developed in priority separable games to reason about the stability of such allocations without having to explicitly construct the improvement graph.

Given an allocation π, a pair of agents (i,j) is called a *blocking pair* if there exists π' such that $\pi'(i) = \pi(j)$, $\pi'(j) = \pi(i)$, $\pi \prec_i \pi'$, $\pi \prec_j \pi'$ and for all $k \in N \setminus \{i,j\}$, $\pi'(k) = \pi(k)$. An allocation π is *stable* if π does not have a blocking pair.

We can use the MLFPC framework to characterise stable allocations in housing markets using the formulas given below. To keep the presentation simple, we retain the notation we used in the previous section. We view the strategy profile as the profile of allocations. While we don't equate the outcomes with the choice of strategies, in typical resource allocation setting, these two parameters are the same. The main difference from the previous section is that unilateral deviation is not possible. The only way a strategy profile can change is when two players agree to swap their items.

- $\text{swap}_{i,j}(\vec{x},\vec{y}) = \bigwedge_{k \in N\setminus\{i,j\}} x_k = y_k \wedge x_i = y_j \wedge y_i = x_j$ - states that the difference between $\vec{x}$ and $\vec{y}$ is that players i and j have swapped their items.
- $\text{chkvOut}(\vec{a}) = \bigwedge_j O(a_j)$.
- $\text{chkPay}(p,\vec{x},\vec{a},\vec{y},\vec{b}) = \text{chkOut}(p,\vec{x},\vec{a}) \wedge \text{chkOut}(p,\vec{y},\vec{b}) \wedge \exists v \left([\mathbf{lfp}_{M,x}\alpha](v,p,\vec{x},\vec{a},\vec{y},\vec{b}) \wedge \psi_1(p,v,\vec{x},\vec{a},\vec{y},\vec{b}) \right)$
- $\phi_{\{i,j\}}(\vec{x},\vec{a},\vec{y},\vec{b}) = \exists w,v (P(w) \wedge P(v) \wedge \mathscr{S}(u,x_i) \wedge \mathscr{S}(v,x_j) \wedge \text{swap}_{i,j}(\vec{x},\vec{y}) \wedge \text{chkPay}(w,\vec{x},\vec{a},\vec{y},\vec{b}) \wedge \text{chkPay}(v,\vec{x},\vec{a},\vec{y},\vec{b}))$ - this formula is similar to the one we saw in the previous section. It says that it is possible to move from an allocation $\vec{x}$ to $\vec{y}$ where players i and j swap their items in which the utility of both the players strictly increase. Due to the inability of players to make unilateral deviation, we have two instances of the **lfp** operator in this formula. Note that the quantifier rank of this formula remains 1.
- $\phi(\vec{x},\vec{a},\vec{y},\vec{b}) = \bigvee_{i<j} \phi_{\{i,j\}}(\vec{x},\vec{a},\vec{y},\vec{b})$.

- swapstep$(\vec{x},\vec{y}) = \exists \vec{a},\vec{b}(\text{chkStrat}(\vec{x}) \wedge \text{chkStrat}(\vec{y}) \wedge \text{chkvOut}(\vec{a}) \wedge \text{chkvOut}(\vec{b}) \wedge \phi(\vec{x},\vec{a},\vec{y},\vec{b}))$.
- stablealloc$(\vec{x}) = \sim \exists \vec{y}\ \text{swapstep}(\vec{x},\vec{y})$ - states that $\vec{x}$ is a stable allocation.

The algorithmic properties of allocation problems where the reference orderings depend on externalities are studied in[10, 15]. Apart from stability, notions of fairness like envy-freeness, propotionality and maximin share guarantee are also well studied in the context of allocation of indivisible items [11, 9]. Analysis of the improvement graph is also useful in the context of fairness notions. Existence of a finite improvement path terminating in a fair allocation would indicate the possibility of convergence to a fair allocation in terms of an exchange dynamics.

7.2 Voting systems

Consider an electorate consisting of a set $[n] = \{1,\ldots,n\}$ of n voters and a set C of m candidates. Let $\mathfrak{R}$ be a voting rule that considers the preference of each voter over the candidates and chooses k winning candidates. The choice sets for all voters are the same $S = \mathfrak{L}(C) = \{\pi | \pi \text{ is a permutation of } C\}$. The outcome set is $O = \binom{C}{k}$. The voting rule $\mathfrak{R} : S^n \mapsto O$ specifies which candidates win given the complete preferences of all voters. We assume that each voter i has a preference ordering $\prec_i$ over the outcome set O. Thus the voting system can be given by the tuple $L = (n, m, \prec_1, \ldots, \prec_n, \mathfrak{R})$.

The improvement graph G_L associated with L is as before: $G_E = (V,E)$ where $V = S^n$, the set of strategy profiles of voters; $E \subseteq (V \times [n] \times V)$ is the improvement relation for voter i, given by: $s \rightarrow_i s'$ if $\mathfrak{R}(s') \succ_i \mathfrak{R}(s)$, $s_i \neq s'_i$ and for all $j \neq i$, $s_j = s'_j$.

Voting equilibria have been studied by Myerson and Weber [28]. In general, one speaks of the *bandwagon effect* in an election if voters become more inclined to vote for a given candidate as her standing in pre-election polls improve, or the *underdog effect*, if voters become less inclined to vote for a candidate as her standing improves. Myerson and Weber suggest that equilibrium arises when the voters, acting in accordance with both their preferences for the candidates and their perceptions of the relative chances of candidates in contention for victory, generate an election result that justifies their perceptions. Note that the improvement path again gives us the possibility of 'interaction' arising from voter preferences, and we can analyse this in the context of specific voting rules.

Given that agents may have incentive to strategically misreport their preferences, it is natural to study the convergence dynamics when voting is modelled as a game. Iterative voting [22, 32, 25] is a formalism that is useful to analyse the strategic dynamics when at each turn a voter is allowed to alter her vote based on the current outcome until it converges to an outcome from which no voter wants to deviate. In general, the outcome of iterative voting may depend on the order of voters' changes. Again, voters act myopically, without knowing the others' preferences. This dynamics is again reflected by the improvement path as discussed here and sink nodes correspond to Nash equilibria. Thus given a voting rule, it is natural to ask what equilibria are reachable from a given vote profile.

Once again, we can use the MLFPC framework to specify convergence dynamics in iterative voting.

7.3 Remark

We have suggested that the improvement graph is an important structure for the logical study of mathematical social sciences. A natural alternative to consider would be to translate all the models into that of games, and then induce the improvement graph over the defined model. This is certainly possible, but in general this can lead to an increase in the size of the graph. Moreover since we hope to use MLFPC not only to unify these contexts but also differentiate them (in terms of logical resources needed), such a reduction would not be helpful.

8 Discussion

We see this paper as a preliminary investigation, hopefully leading to a descriptive complexity theoretic study of fundamental notions in games and interaction. It is clear that fixed-point computations underlie the reasoning in a wide variety of such contexts, and logics with least fixed-point operators are natural vehicles of such reasoning. We expect that this is a minimal language for improvement dynamics, but with further vocabulary restrictions that need to be worked out. Proceeding further, we would like to delineate bounds on the use of logical resources for game theoretic reasoning. For instance, one natural question is the characterization of equilibrium dynamics definable with at most one second order (fixed-point) variable.

Expressiveness needs to be sharpened from the perspective of models as well. We would like to characterize the class of improvement graphs for different subclasses of games, resource allocation systems and voting rules, considering the wide variety of details in the literature. This would in general necessitate enriching the logical language and we wish to consider minimal extensions.

Another important issue is the identification of subclasses that avoid the navigation of huge improvement graphs. Potential games provide an interesting subclass and they correspond to some appropriate allocation rules and forms of voting (under specific election rules). But these are only specific exemplifying instances, studying the stucture of formulas and their models will (hopefully) lead us to many such correspondences.

An important direction is the study of infinite strategy spaces. Clearly the model checking algorithm needs a finite presentation of the input but this is possible and it is then interesting to explore convergence of fixed-point iterations.

References

[1] R. Alur, T. A. Henzinger & O. Kupferman (2002): *Alternating-Time Temporal Logic*. Journal of the ACM 49, pp. 672–713, doi:10.1145/585265.585270.

[2] K.R. Apt, B. Keijzer, M. Rahn, G. Schäfer & S. Simon (2017): *Coordination games on graphs*. International Journal of Game Theory 46(3), p. 851–877, doi:10.1007/s00182-016-0560-8.

[3] K.R. Apt & S. Simon (2015): *A classification of weakly acyclic games*. Theory and Decision 78(4), pp. 501–524, doi:10.1007/978-3-642-33996-7_1.

[4] K.R. Apt, S. Simon & D. Wojtczak (2015): *Coordination Games on Directed Graphs*. In: Proc. of 15th International Conference on Theoretical Aspects of Rationality and Knowledge, doi:10.4204/EPTCS.215.6.

[5] J. Benthem (2001): *Games in dynamic epistemic logic*. Bulletin of Economic Research 53(4), pp. 219–248, doi:10.1111/1467-8586.00133.

[6] J. Benthem (2002): *Extensive games as process models*. Journal of Logic Language and Information 11, pp. 289–313, doi:10.1023/A:1015534111901.

[7] J. Benthem & A. Gheerbrant (2010): *Game Solution, Epistemic Dynamics and Fixed-Point Logics*. Fundamenta Informaticae 100(1-4), pp. 19–41, doi:10.3233/FI-2010-261.

[8] G. Bonanno (2001): *Branching Time Logic, Perfect Information Games and Backward Induction*. Games and Economic Behavior 36(1), pp. 57–73, doi:10.1006/game.1999.0812.

[9] S. Bouveret, Y. Chevaleyren & N. Maudet (2016): *Fair Allocation of Indivisible Goods*, chapter 12. Handbook of Computational Social Choice, Cambridge University Press.

[10] S. Branzei, A.D. Procaccia & J. Zhang (2013): *Externalities in cake cutting*. In: Proceedings of the 23rd IJCAI, pp. 55–61.

[11] E. Budish (2011): *The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes*. Journal of Political Economy 119(6), pp. 1061–1103, doi:10.1086/664613.

[12] K. Chatterjee, T.A. Henzinger & N. Piterman (2010): *Strategy logic*. Information and Computation 208(6), pp. 677–693, doi:10.1016/j.ic.2009.07.004.

[13] H. Ebbinghaus & J. Flum (1999): *Finite Model Theory*. Springer, doi:10.1007/3-540-28788-4.

[14] R. Engelberg & M. Schapira (2011): *Weakly-Acyclic (Internet) Routing Games*. In: Proc. 4th International Symposium on Algorithmic Game Theory (SAGT11), Lecture Notes in Computer Science 6982, Springer, pp. 290–301, doi:10.1007/978-3-642-24829-0_26.

[15] M. Ghodsi, H. Saleh & M. Seddighin (2018): *Fair Allocation of Indivisible Items With Externalities*. CoRR abs/1805.06191. Available at `http://arxiv.org/abs/1805.06191`.

[16] V. Goranko (2003): *The Basic Algebra of Game Equivalences*. Studia Logica 75(2), pp. 221–238, doi:10.1023/A:1027311011342.

[17] M. Grohe (1994): *The Structure of Fixed Point Logics*. Albert-Ludwigs-Universitat.

[18] P. Harrenstein, W. Hoek, J.J. Meyer & C. Witteven (2003): *A Modal Characterization of Nash Equilibrium*. Fundamenta Informaticae 57(2-4), pp. 281–321.

[19] W. Hoek, W. Jamroga & M. Wooldridge (2005): *A Logic for Strategic Reasoning*. Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems, pp. 157–164, doi:10.1145/1082473.1082497.

[20] E.B. Janovskaya (1968): *Equilibrium Points in Polymatrix Games*. Litovskii Matematicheskii Sbornik 8, pp. 381–384.

[21] Bernd Kawald & Pascal Lenzner (2013): *On Dynamics in Selfish Network Creation*. In: Proceedings of the 25th ACM Symposium on Parallelism in Algorithms and Architectures, ACM, pp. 83 – 92, doi:10.1145/2486159.2486185.

[22] O. Lev & J. S. Rosenschei (2012): *Convergence of iterative voting*. In: Proceedings of AAMAS-2012, pp. 611–618.

[23] L. Libkin (2013): *Elements of finite model theory*. Springer, doi:10.1007/978-3-662-07003-1.

[24] J.R. Marden, G. Arslan & J.S. Shamma (2007): *Regret based dynamics: convergence in weakly acyclic games*. In: Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), IFAAMAS, pp. 194–201, doi:10.1145/1329125.1329175.

[25] R. Meir, M. Polukarov, J.S. Rosenschein & N. R. Jennings (2017): *Iterative voting and acyclic games*. Artificial Intelligence 252, pp. 100–122, doi:10.1016/j.artint.2017.08.002.

[26] I. Milchtaich (1996): *Congestion Games with Player-Specific Payoff Functions*. Games and Economic Behaviour 13, pp. 111–124, doi:10.1006/game.1996.0027.

[27] D. Monderer & L. S. Shapley (1996): *Potential Games*. Games and Economic Behaviour 14, pp. 124–143, doi:10.1006/game.1996.0044.

[28] R. B. Myerson & R. J. Weber (1993): *A Theory of Voting Equilibria*. The American Political Science Review 87(1), pp. 102–114, doi:10.2307/2938959.

[29] M. Otto (1997): *Bounded Variable Logics and Counting: A Study in Finite Models*. Cambridge University Press.

[30] R. Parikh (1985): *The logic of games and its applications*. Annals of Discrete Mathematics 24, pp. 111–140, doi:10.1016/S0304-0208(08)73078-0.

[31] R. Ramanujam & S. Simon (2008): *Dynamic logic on games with structured strategies*. In: Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR-08), AAAI Press, pp. 49–58.

[32] A. Reijngoud & U. Endriss (2012): *Voter response to iterated poll information*. In: Proceedings of AAMAS-2012, pp. 635–644.

[33] R. W. Rosenthal (1973): *A Class of Games Possessing Pure-Strategy Nash Equilibria*. International Journal of Game Theory 2(1), pp. 65–67, doi:10.1007/BF01737559.

[34] N. Schweikardt (2005): *Arithmetic, first-order logic, and counting quantifiers*. ACM Transactions on Computational Logic 6(3), pp. 634–671, doi:10.1145/1071596.1071602.

[35] N. Schweikardt (2006): *On the expressive power of monadic least fixed point logic*. Theoretical Computer Science 350, p. 325–344, doi:10.1016/j.tcs.2005.10.025.

[36] L. S. Shapley & H. Scarf (1974): *On cores and indivisibility*. Journal of Mathematical Economics 1(1), pp. 23–37, doi:10.1016/0304-4068(74)90033-0.

[37] D. Walther, W. Hoek & M. Wooldridge (2007): *Alternating-time Temporal Logic with Explicit Strategies*. In: Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2007), pp. 269–278, doi:10.1145/1324249.1324285.

[38] H. P. Young (1993): *The evolution of conventions*. Econometrica 61(1), pp. 57–84, doi:10.2307/2951778.