Cooperative Concurrent Games

Julian Gutierrez[a], Szymon Kowara[c], Sarit Kraus[b], Thomas Steeples[c], Michael Wooldridge[c]

[a]*Faculty of Information Technology, Monash University, Australia*
[b]*Department of Computer Science, Bar-Ilan University, Israel*
[c]*Department of Computer Science, University of Oxford, United Kingdom*

Abstract

In *rational verification*, the aim is to verify which temporal logic properties will hold in a concurrent game, under the assumption that players choose strategies that form a game theoretic equilibrium. Player preferences are typically defined by assuming that agents act in pursuit of individual goals, specified as temporal logic formulae. To date, rational verification has been studied using *non-cooperative* solution concepts – Nash equilibrium and refinements thereof. Such non-cooperative solution concepts assume that there is no possibility of agents forming binding agreements to cooperate, and as such they are restricted in their applicability. In this paper, we extend rational verification to *cooperative* solution concepts, as studied in the field of cooperative game theory. We focus on the *core*, as this is the most fundamental (and most widely studied) cooperative solution concept. We begin by presenting a variant of the core that seems well-suited to the concurrent game setting, and we show that this version of the core can be characterised using ATL*. We then study the computational complexity of key decision problems associated

[⋆]This paper is an extended and revised version of [1].

[∗]Corresponding author: Julian Gutierrez.

Email addresses: julian.gutierrez@monash.edu (Julian Gutierrez), sarit@cs.biu.ac.il (Sarit Kraus), thomas.steeples@cs.ox.ac.uk (Thomas Steeples), michael.wooldridge@cs.ox.ac.uk (Michael Wooldridge)

with the core, which range from problems in PSpace to problems in 3ExpTime. We also investigate conditions ensuring that the core is not empty and explore when it is invariant under bisimilarity. We then introduce and study a number of variants of the main definition of the core, leading to the issue of credible deviations, and to stronger notions of collective stable behaviour.

Keywords: Concurrent games; Cooperative games; the Core; Multi-agent Systems; Logic; Formal verification

---

1. Introduction

Concurrent games have become established as a key model for concurrent and multi-agent systems, in both the AI community and the verification/computer science community [2–5]. A concurrent game [2] is a finite-state environment, populated by a collection of independent, self-interested agents. A game takes place over an infinite sequence of rounds, where at each round, each agent chooses an action to perform. Preferences in concurrent games are typically modelled by assuming that each agent is associated with a temporal logic goal formula [6], which it desires to see satisfied. The infinite plays generated by a game will either satisfy or fail to satisfy each player's goal, and players act rationally in an attempt to achieve their goal. Since the satisfaction of a player's goal may be dependent on the choices made by other players, then players must make choices strategically in order to play optimally.

In all previous studies that we are aware of, concurrent games are assumed to be *non-cooperative*: players act alone, and binding agreements between players are ruled out. The game theoretic solution concepts used in previous studies of concurrent games have therefore been non-cooperative — primarily Nash equilibrium and refinements thereof. In such a non-cooperative setting, the basic questions that we ask of a concurrent game are, for example, whether a particular temporal logic property holds on *some* computation of the system that could arise through players selecting strategies that form a Nash equilibrium (the E-Nash problem) or whether a property holds on *all* such computations (A-Nash) [3–5]. These problems can be understood as game-theoretic counterparts of the conventional model checking problem [7]: in model checking, we are typically interested in whether a particular property could hold on some or all possible computations of a system, whereas in rational verification, we are interested in whether a property holds on some or all computations that could arise through rational choices on the part of individual players. The complexity of the corresponding decision problems in concurrent games have been extensively studied, and there now exist a small number of software tools that support rational verification [8–11].

The aim of the present paper is to extend the study of concurrent games to include *cooperative* solution concepts [12–14]. Thus, we assume there is some (exogenous) mechanism through which players in a concurrent game can reach binding agreements and form coalitions in order to collectively achieve goals (although we emphasise that the nature of such a mechanism is beyond the scope of the present work). The possibility of binding cooperation and coalition formation eliminates some undesirable equilibria that arise in non-cooperative settings, and makes available a range of outcomes that cannot be achieved without cooperation. We focus on the *core* as our key solution concept. The basic idea behind the core is that it consists of those strategy profiles from which no subset of players could benefit by collectively deviating from it. Now, in conventional cooperative games (characteristic function games with transferable utility [14]), this intuition can be given a simple and natural formal definition, and as a consequence the core is probably the most widely-studied solution concept for cooperative games. However, the conventional definition of the core does not naturally map into our concurrent game setting, because in our games, coalitions are subject to *externalities*: whether a coalition has a beneficial deviation depends not just on the makeup of that coalition, *but on the behaviour of the remaining players* in the game too.

We begin by introducing the framework of concurrent games, and then proceed to define two variations of the core for such settings. In the first, a coalition of players are assumed to have a beneficial deviation if they have some course of action available to them which they would benefit from *no matter what the remaining players did* (*cf.*, the concept of the α-core in the game theory literature). However, this "worst case" (maximim) formulation of the core requires a deviation to be beneficial against *all* courses of action by the remaining players—even those that the remaining agents would not rationally choose. This motivates a second definition, where a deviation is only required to be beneficial against all courses of action by remaining players that are *credible*, in the sense that those players would then be no worse off than they were originally. We also consider games where the agents have *quantitative* preferences. In each case, we formally define the relevant solution concept, identify some of its key computational properties, give logical characterisations, and where possible, provide complexity results, which range from properties that can be checked in PSPACE to properties that can be checked in 3EXPTIME. We also study model theoretic properties related to the core: in particular, whether it is guaranteed to be non-empty, and whether temporal logic properties hold across bisimilar systems over plays (computation runs) induced by elements in the core (a highly desirable property from a formal verification perspective).

Structure of the pape.r. The remainder of this paper is organised as follows:

- In the following section, we summarise the key concepts from logic and concurrent games that are used in the remainder of the paper.

- In Section 3, we define the core and the main computational properties associated with it.
- In Section 4, we present our main results.
- Section 5 we study the issue of credible coalition formations, with associated complexity results.
- In Section 6, we study the core in the *quantitative* setting of mean-payoff games.
- We present some concluding remarks and discuss related work in Section 7, including a discussion around the implementation of our concepts using model checkers.

2. Preliminaries

Set and Sequences. Given any set $S = \{s, q, r, \ldots\}$, we use S^*, S^ω, and S^+ for, respectively, the sets of finite, infinite, and non-empty finite sequences of elements in S, respectively. If $w_1 = s^1 s^2 \ldots s^k \in S^*$ and w_2 is any other (finite or infinite) sequence, we write $w_1 w_2$ for their concatenation $s^1 s^2 \ldots s^k w_2$. For $Q \subseteq S$, we write S_{-Q} for $S \setminus Q$ and S_{-i} if $Q = \{i\}$. We extend this notation to tuples $u = (s_1, \ldots, s_k, \ldots, s_n)$ in $S_1 \times \cdots \times S_n$, and write u_{-k} for $(s_1, \ldots, s_{k-1}, s_{k+1}, \ldots, s_n)$, and similarly for sets of elements, that is, by u_{-Q} we mean u without each s_k, for $k \in Q$. Given a sequence w, we write $w[t]$ for the element in position $t+1$ in the sequence; for instance, $w[0]$ is the first element of w. We also use *slice notation*: we write $w[l \ldots m]$ for the sequence $w[l] \ldots w[m-1]$, $w[l \ldots]$ for $w[l]w[l+1] \ldots$, and $w[\ldots m]$ for $w[0] \ldots w[m-1]$; if $m = 0$, we let $w[l \ldots m]$ be the empty sequence, denoted ϵ.

Games. We begin by introducing the model of multi-agent systems that we use throughout the remainder of the paper: **concurrent game structures** [2]. Informally, a concurrent game consist of a set of players, a set of actions for each of those players, a set of system states, and a transition function which describes how the state of the game changes, given a current state and an action for each of the players.

Formally, a concurrent game structure, M, is given by a tuple,

$$M = (\mathrm{Ag}, \mathrm{St}, \{\mathrm{Ac}_i\}_{i \in \mathrm{Ag}}, s^0, \mathrm{tr}),$$

where:

- Ag and St are finite, non-empty sets of **agents** and **states**, respectively. We usually identify Ag with the set $\{1, \ldots, n\}$;
- For each $i \in \mathrm{Ag}$, Ac_i is a finite, non-empty set of **actions** available to agent i. We associate each state s with a set of actions available at that state, $\mathrm{Ac}_i(s) \subseteq \mathrm{Ac}_i$, and we write Ac for $\mathrm{Ac}_1 \times \cdots \times \mathrm{Ac}_n$;
- $s^0 \in \mathrm{St}$ is the **initial/start** state; and finally,
- $\mathrm{tr} : \mathrm{St} \times \mathrm{Ac} \rightarrow \mathrm{St}$ is the **transition function** of the game.

The *size* of M is defined to be $|\mathrm{St}| \times |\mathrm{Ac}|^{|\mathrm{Ag}|}$.

Given a concurrent game structure M, we can *play* a game on it as follows: the game starts in state s^0, and each player $i \in \mathrm{Ag}$ chooses an action available to them, $\mathrm{ac}_i^0 \in \mathrm{Ac}_i(s^0)$. The game then moves to a new state,

$$s^1 = \mathrm{tr}(s^0, \mathrm{ac}_1^0, \ldots, \mathrm{ac}_n^0).$$

This process is then repeated. We typically write s^i for the i^{th} state in the sequence, and $\mathrm{ac}^i = (\mathrm{ac}_1^i, \ldots, \mathrm{ac}_n^i)$ for the i^{th} vector of actions played in the sequence. Thus for all $t \in \mathbb{N}$, we have

$$s^{t+1} = \mathrm{tr}(s^t, \mathrm{ac}^t)$$

A **run**, ρ is a infinite sequence $\rho = s^0 s^1 s^2 \ldots$ such that for every $t \geq 0$, there exists some ac $\in \mathrm{Ac}$ such that $s^{t+1} = \mathrm{tr}(s^t, \mathrm{ac})$. A **path**, π is a finite prefix of a run.

A **strategy** for a player i is a function $\sigma_i \colon \mathrm{St}^+ \to \mathrm{Ac}_i$ such that $\sigma_i(\pi s) \in \mathrm{Ac}_i(s)$ for every $\pi \in \mathrm{St}^*$ and $s \in \mathrm{St}$. That is, for every path π, a strategy for a player i gives an action available to i from the last state of that path. The set of strategies for player i is denoted by Σ_i. A **strategy profile** is a tuple $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ in $\Sigma_1 \times \cdots \times \Sigma_n$. Observe that a strategy profile, $\vec{\sigma}$, along with a state s, induces a unique run, ρ, where $\rho[0] = s$ and $\rho[t+1] = \delta(\rho[t], \sigma_1(\rho[\ldots t]), \ldots, \sigma_n(\rho[\ldots t]))$, for all $t \geq 0$. We write $\rho(\vec{\sigma}, s)$ for such a run, and simply $\rho(\vec{\sigma})$ if $s = s^0$.

Note that viewing strategies as functions $\sigma_i \colon \mathrm{St}^+ \to \mathrm{Ac}_i$ is problematic with respect to computational analysis, because the domain of such a function is infinite. To be able to answer questions relating to (for example) computational complexity, we need a finite representation for strategies that must operate over an infinite number of rounds. We present such a representation later.

LTL Games. In the first class of games we consider, preferences for players are specified by associating with them a temporal logic goal formula that they desire to see satisfied. For this purpose, we make use of *Linear Temporal Logic* (LTL) [15]. LTL is a widely used logic for reasoning about the behaviours of concurrent systems, and while we present the key concepts here, we refer the reader to any standard temporal logic textbook for details (*e.g.*, [16]).

Let AP be a set of propositional variables. Then the syntax of an LTL formula φ is given by the following grammar:

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid \boldsymbol{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi,$$

where $p \in \mathrm{AP}$. The set of all LTL formulae over a set of propositional variables AP is denoted $\mathcal{L}(\mathrm{AP})$. If AP is clear from the context, we may instead just write $\mathcal{L}$. We also introduce the traditional propositional abbreviations, $\cdot \wedge \cdot$, $\cdot \rightarrow \cdot$, $\cdot \leftrightarrow \cdot$, defined in the usual way, as well as the abbreviations $\mathbf{F}\,\varphi$ for $\top\,\mathbf{U}\,\varphi$, and $\mathbf{G}\,\varphi$ for $\neg\,\mathbf{F}\,\neg\varphi$.

Traditionally, the semantics of LTL formulae are defined relative to labelled transition systems [17], or Kripke structures [18], but for our purposes, we define them with respect to the infinite runs generated by concurrent game structures. Formally, let M be a concurrent game structure, and let $\lambda : \mathrm{St} \to \mathcal{P}\,(\mathrm{AP})$ be a labelling function, mapping states to sets of propositional variables. Then given an infinite run $\rho \in \mathrm{St}^\omega$ and an LTL formula φ, we say that (M, λ, ρ) *models* φ and write $(M, \lambda, \rho) \models \varphi$ according to the following inductive definition:

- For $p \in \mathrm{AP}$, we have $(M, \lambda, \rho) \models p$ if and only if $p \in \lambda(\rho[0])$;
- For $\varphi \in \mathcal{L}(\mathrm{AP})$, we have $(M, \lambda, \rho) \models \neg\varphi$ if and only if it is not the case that $(M, \lambda, \rho) \models \varphi$;
- For $\varphi, \psi \in \mathcal{L}(\mathrm{AP})$, we have $(M, \lambda, \rho) \models \varphi \wedge \psi$ if and only if we have both $(M, \lambda, \rho) \models \varphi$ and $(M, \lambda, \rho) \models \psi$;
- For $\varphi \in \mathcal{L}(\mathrm{AP})$, we have $(M, \lambda, \rho) \models \boldsymbol{X}\,\varphi$ if and only if we have $(M, \lambda, \rho[1\ldots]) \models \varphi$;
- For $\varphi, \psi \in \mathcal{L}(\mathrm{AP})$, we have $(M, \lambda, \rho) \models \varphi\,\mathbf{U}\,\psi$ if and only if there exists some $j \geq 0$ such that $(M, \lambda, \rho[j\ldots]) \models \psi$ and for all $i < j$ we have $(M, \lambda, \rho[i\ldots]) \models \varphi$.

For notational convenience, we will write $(G, \rho) \models \varphi$ as shorthand for $(M, \lambda, \rho) \models \varphi$, and if G is apparent from the context, we will just write $\rho \models \varphi$.

We can now define **LTL games**. Formally, an LTL game, G, is given by a structure

$$G = (M, \mathrm{AP}, \lambda, (\gamma_i)_{i \in \mathrm{Ag}}),$$

where M is a concurrent game structure, AP is a set of atomic propositions, $\lambda : \mathrm{St} \to \mathcal{P}(\mathrm{AP})$ is a labelling function, and for each $i \in \mathrm{Ag}$, γ_i is an LTL formula over AP that defines that player's preference relation over runs.

In more detail, we use temporal goal formulae γ_i to define preference relations $\succeq_i$ over runs, as follows. For two runs ρ, ρ', and a player i, we have $\rho \succeq_i \rho'$ if and only if $(M, \lambda, \rho') \models \gamma_i$ implies that $(M, \lambda, \rho) \models \gamma_i$. Strict preference relations $\succ_i$ are defined in the standard way: $\rho \succ_i \rho'$ iff $\rho \succeq_i \rho'$ but not $\rho' \succeq_i \rho$.

With preference relations now defined, we can begin to define game theoretic solution concepts. First, a strategy profile $\vec{\sigma}$ is said to be a **Nash equilibrium** if there is no player $i \in \mathrm{Ag}$ and strategy σ'_i for i such that we have $(\vec{\sigma}_{-i}, \sigma'_i) \succ_i \vec{\sigma}$. That is, $\vec{\sigma}$ is a Nash equilibrium if no player can benefit by unilaterally changing its strategy (assuming all other players leave their strategies unchanged). Let $NE(G)$ denote the Nash equilibria of the game G.

We emphasise that Nash equilibrium only considers *unilateral* deviations, *i.e.*, deviations by individual players. Compare this to the notion of a **strong Nash equilibrium**: a strategy profile $\vec{\sigma}$ is a strong Nash equilibrium if there is no coalition C and no strategy σ'_C such that for all $i \in C$ we have $(\vec{\sigma}_{-i}, \sigma'_i) \succ_i \vec{\sigma}$. Thus, strong Nash equilibria are those strategy profiles which are immune to multilateral deviations.

In an LTL game, given that we have dichotomous preferences, we can identify for every run ρ a set of "winners" and a set of "losers" for each run ρ. Formally, let $\mathcal{W}(\rho)$ denote the set of players that get their goal achieved under ρ, and let $\mathcal{L}(\rho)$ denote the set of players that do not:

$$\mathcal{W}(\rho) = \{i \in \mathrm{Ag} \mid \rho \models \gamma_i\}$$
$$\mathcal{L}(\rho) = \mathrm{Ag} \setminus \mathcal{W}(\rho).$$

For the remainder of the paper, we will primarily consider LTL games, before studying the quantitative setting in Section 6.

3. Cooperative Rational Verification

Defining the Core. We want to define counterparts of the rational verification problems E-NASH and A-NASH, as studied in [3, 4], but for cooperative settings. For this, we need a version of the core for our concurrent game setting. The core is probably the best-known solution concept in cooperative game theory. Like Nash equilibrium in the non-cooperative setting, the core defines a notion of stability for games, but whereas Nash equilibrium only requires that an outcome is stable in the sense that it admits no *individual* beneficial deviations, the core requires that an outcome admits no beneficial deviations by *coalitions*. In the "standard" model of cooperative games, this intuition is easily formalised, but in concurrent games, there is an important difficulty. Suppose a coalition of players $C \subseteq \mathrm{Ag}$ are contemplating participating in a strategy profile $\vec{\sigma}$, and in particular, are attempting to determine whether they have a cooperative beneficial deviation from $\vec{\sigma}$. Now, as they consider possible beneficial deviations — collective strategies $\vec{\sigma}_C$ — *what assumptions should C make about the behaviour of the remaining players* $\mathrm{Ag} \setminus C$*?* In particular, assuming that the remaining players will not alter their strategy is implausible in a cooperative setting[1], rational players who can cooperate will respond to the deviation rationally and in a cooperative way against the players in C. And, crucially, whether or not C's putative deviation is in fact beneficial may well depend upon the behaviour of the remaining players. In game theoretic terms, our concurrent game setting is subject to externalities: the performance of the coalition C depends not just on the coalition C, but on the behaviour of the remaining players.

[1]This is the kind of behaviour that one has to assume to define strong Nash equilibrium, a non-cooperative solution concept.

It is well-known that cooperative solution concepts are difficult to define in the presence of externalities [14]. In particular, there is no universally accepted definition of the core for games with externalities. Our first definition of the core for concurrent games, therefore, captures *worst case* reasoning. Thus, when coalition C is contemplating a deviation, it requires that this deviation will be beneficial *no matter what the remaining players do*. This idea has been explored in the concept of the α-core in cooperative games [19]. To make this idea formal, we need to define the notion of a **beneficial deviation**. Let $\vec{\sigma}$ be a strategy profile and let C be a coalition; then we say that $\vec{\sigma}'_C$ is a *beneficial deviation* from $\vec{\sigma}$ if:

1. $C \subseteq \mathcal{L}(\vec{\sigma})$
2. For all $\vec{\sigma'}_{-C}$, we have $C \subseteq \mathcal{W}((\vec{\sigma}'_C, \vec{\sigma}'_{-C}))$.

In other words, $\vec{\sigma}'_C$ is said to be a beneficial deviation from $\vec{\sigma}$ if the players in C would be better off playing $\vec{\sigma}'_C$, rather than their respective strategies in $\vec{\sigma}$, *no matter what strategies the players outside C play*. The core of a game G, denoted $core(G)$, is then defined to be the set of outcomes of G that admit no beneficial deviation.

Example 1. Consider the following game, which contains a poor quality Nash equilibrium that is not in the core: the ability to cooperate makes it possible for agents to avoid the undesirable equilibrium. The game contains two players, $\mathrm{Ag} = \{1, 2\}$ and two variables $\mathrm{AP} = \{p, q\}$, with player 1's action set being $\mathrm{Ac}_1 = \{pt, pf\}$ and player 2's action set being $\mathrm{Ac}_2 = \{qt, qf\}$, satisfying that, for every reachable state, if player 1/2 plays pt/qt then p/q will hold, and will not hold if pf/qf is played instead (*i.e.*, player 1 "controls" the value of p and player 2 the value of q). Their goals are identical (and so the game is a coordination game): $\gamma_1 = \gamma_2 = \mathbf{G}(p \wedge q)$. Now, consider the strategy profile $\vec{\sigma}$ in which both players simply fix their respective variables to be false forever (*i.e.*, play pf and qf forever). Neither player will have their goal achieved by such a strategy profile. However, the strategy profile forms a

Nash equilibrium, because unilateral deviation cannot improve the situation: neither player has an alternative strategy which would make them better off. In fact, there are infinitely many such poor quality Nash equilibria in this game, where neither player gets their goal achieved. However, this strategy profile is *not* in the core, because there is a cooperative beneficial deviation to the strategy profile in which both players fix their variables to be true forever (*i.e.*, play pt and qt forever). And, in fact, in every strategy profile which lies in the core, both players get their goal achieved. Thus, using the core instead of Nash equilibrium eliminates poor quality equilibria from the game, leading to socially more desirable outcomes.

Decision Problems. In Rational Verification [3–5] we are mainly interested in checking which temporal logic properties are satisfied in a given solution concept of a game; typically, in the non-cooperative setting, we study what LTL formulae hold in the Nash equilibria $NE(G)$ of a game G. In the cooperative setting, as introduced here, we are instead interested in what properties hold in the core of the game. The two main decision problems in rational verification are checking whether a temporal logic formula is satisfied by some/every stable strategy profile of the game. For the core, these problems are defined as follows—*cf.* [3–5].

E-CORE:
Given: Game G, LTL formula φ.
Question: Does there exist some strategy profile $\vec{\sigma}$ in the core of G such that $\rho(\vec{\sigma}) \models \varphi$ holds?

A-CORE:
Given: Game G, LTL formula φ.
Question: Is it the case that for all strategy profiles $\vec{\sigma}$ in the core of G, we have $\rho(\vec{\sigma}) \models \varphi$?

In addition to the two above decision problems, the third main decision problem in rational verification is checking whether, given a game G, its set of stable strategy profiles — the core of G in this case — is non-empty. As will be shown in the next section, in our setting, the core of every game G is never empty, a desirable game-theoretic property as it ensures the existence of stable strategy profiles for every game, making them rationally implementable in practice.

We will also be interested in two additional decision problems. Namely, checking whether a given strategy profile is in the core (CORE MEMBERSHIP), and checking whether a given strategy vector for a coalition is a beneficial deviation with respect to a strategy profile (BENEFICIAL DEVIATION). These two decision problems are formally defined as follows:

CORE MEMBERSHIP:

Given: Game G, outcome $\vec{\sigma}$.

Question: Is it the case that $\vec{\sigma}$ is a member of the core?

BENEFICIAL DEVIATION:

Given: Game G, outcome $\vec{\sigma}$, deviation $\vec{\sigma}'_C$.

Question: Is $\vec{\sigma}'_C$ a beneficial deviation from $\vec{\sigma}$?

One might think that every coalition that has a beneficial deviation from some outcome of the game will get their goals achieved under a run induced by some member of the core, but that actually is not the case. To formalise this idea, let us introduce the concept of a **fulfilled coalition**. We say that a coalition of players is fulfilled if they are able to achieve their goals irrespective of what other players do. Formally, we say that a coalition of players C is fulfilled if there is a joint strategy $\vec{\sigma}_C$ for $C \subseteq \mathrm{Ag}$ such that for all joint strategies $\vec{\sigma}_{-C}$ for $\mathrm{Ag} \setminus C$ we have

$$\rho((\vec{\sigma}_C, \vec{\sigma}_{-C})) \models \bigwedge_{i \in C} \gamma_i \; .$$

In other words, a fulfilled coalition has a winning strategy to collectively achieve their goals. Since we are considering cooperative games, the issue/question is whether such a coalition will form. Using the above definition, we can make some useful observations about (fulfilled) coalitions and members of the core. These observations are formally presented in the following lemma, which relates winning strategies and the core in a critical way.

Lemma 1.

1. *There are games G, with strategy profiles $\vec{\sigma} \in core(G)$, containing fulfilled coalitions $C \subseteq \mathrm{Ag}$ such that $C \not\subseteq \mathcal{W}(\rho(\vec{\sigma}))$;*
2. *For every game G, strategy profile $\vec{\sigma} \in core(G)$, and fulfilled coalition C, we have that $C \cap \mathcal{W}(\rho(\vec{\sigma})) \neq \emptyset$;*
3. *For every game G and fulfilled coalition C, if $core(G) \neq \emptyset$, then there is $\vec{\sigma} \in core(G)$ such that $C \subseteq \mathcal{W}(\rho(\vec{\sigma}))$.*

Informally, the first part of the lemma says that the fact that a coalition is fulfilled does not mean that every player in such a coalition is guaranteed to get its goal achieved under an arbitrary member of the core. However, the second part of the lemma says that in any member of the core, some agents of every fulfilled coalition must get their goals achieved. And, the third part of the lemma says that for every fulfilled coalition the core contains a strategy profile in which every player of this coalition gets its goal achieved. Because fulfilled coalitions can help us understand the coalition formation power in a game, we will also be interested in the following decision problem about coalitions.

FULFILLED COALITION:

Given: Game G, coalition $C \subseteq \mathrm{Ag}$.

Question: Is C a fulfilled coalition of G?

In the next section, we will investigate the decision problems defined here as well as some model-theoretic properties of the core.

4. Reasoning about the Core

In this section we will study the computational complexity of the decision problems defined in the previous section, and will show some other properties of the core of an LTL game, in particular, that such a set is never empty and that the satisfaction of an LTL property on some/every outcome in the core is bisimulation-invariant [20]. These two results sharply contrast with the Rational Verification problem for non-cooperative games; in these, the set of Nash equilibria of a game is not guaranteed to always be non-empty [3] nor does bisimulation-invariance hold in the general case [21].

Before proceeding, we need to introduce an additional logic, ATL*, so that we can reason about coalitions more effectively.

Logics. Alternating-time Temporal Logic (ATL* [2]) is an extension of CTL* [22], a branching-time temporal logic, that allows for reasoning about games and strategies. More specifically, given a set of atomic propositions AP and a set of agents Ag, the language of ATL* formulae is generated from φ in the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle C \rangle\rangle\, \psi$$
$$\psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \boldsymbol{X}\, \psi \mid \psi\, \mathbf{U}\, \psi$$

such that $p \in \mathrm{AP}$ and $C \subseteq \mathrm{Ag}$. We call the formulae produced by φ in the above grammar *ATL* state formulae*, and denote the set that contains them by $\mathcal{L}_s(\mathrm{AP}, \mathrm{Ag})$ and those generated by ψ in the above grammar *ATL* path formulae*, denoted by $\mathcal{L}_p(\mathrm{AP}, \mathrm{Ag})$. They are given these names as their semantics are defined relative to states and paths respectively. However, to reiterate, only ATL* state formulae are valid ATL* formulae, and thus, we write $\mathcal{L}(\mathrm{AP}, \mathrm{Ag})$ as shorthand for $\mathcal{L}_s(\mathrm{AP}, \mathrm{Ag})$. When either AP or Ag, or both, are known, we may omit them. With $\mathrm{AP}' \subseteq \mathrm{AP}$, we may write $\varphi|_{\mathrm{AP}'}$ if $\varphi \in \mathcal{L}(\mathrm{AP}', \mathrm{Ag})$ for some set of agents Ag.

We use the following abbreviations: we write $\top$ for $p \vee \neg p$, $\bot$ for $\neg\top$, $\mathbf{F}\,\varphi$ for $\top\,\mathbf{U}\,\varphi$, $\mathbf{G}\,\varphi$ for $\neg\,\mathbf{F}\,\neg\varphi$, $\mathbf{E}\,\varphi$ for $\langle\langle \mathrm{Ag} \rangle\rangle\,\varphi$, $\mathbf{A}\,\varphi$ for $\langle\langle \emptyset \rangle\rangle\,\varphi$, and $[\![C]\!]\,\varphi$ for $\neg\,\langle\langle C \rangle\rangle\,\neg\varphi$; we also use the conventional abbreviations for other classical propositional logic operators. Finally, we define the *size* of an ATL* formulae φ as its number of subformulae.

To define the semantics of ATL* formulae, we actually need to define two semantic relations, $\models_s$ (for state formulae) and $\models_p$ (for path formulae). So, let M be some concurrent game structure, along with a labelling function $\lambda : \mathrm{St} \to \mathcal{P}(\mathrm{AP})$. Then given a state, $s \in St$ and an ATL* formula φ, we say that (M, λ, s) *models* φ and write $(M, \lambda, s) \models \varphi$ according to the following inductive definition:

- For $\varphi \in \mathcal{L}_s(\mathrm{AP}, \mathrm{Ag})$, we have $(M, \lambda, s) \models \varphi$ if and only if $(M, \lambda, s) \models_s \varphi$;
- For $p \in \mathrm{AP}$, we have $(M, \lambda, s) \models_s p$ if and only if $p \in \lambda(s)$;
- For $\varphi \in \mathcal{L}_s(\mathrm{AP}, \mathrm{Ag})$, we have $(M, \lambda, s) \models_s \neg\varphi$ if and only if it is not the case that $(M, \lambda, s) \models_s \varphi$;
- For $\varphi, \psi \in \mathcal{L}_s(\mathrm{AP}, \mathrm{Ag})$, we have $(M, \lambda, s) \models_s \varphi \wedge \psi$ if and only if $(M, \lambda, s) \models_s \varphi$ and $(M, \lambda, s) \models_s \psi$;
- For $\varphi \in \mathcal{L}_p(\mathrm{AP}, \mathrm{Ag})$, we have $(M, \lambda, s) \models_s \langle\langle C \rangle\rangle\,\varphi$ if and only if there is some strategy vector $\vec{\sigma}_C$ for the coalition C, such that for all complementary strategy profiles, $\vec{\sigma}_{\mathrm{Ag}\setminus C}$, it is the case that $(M, \lambda, \rho((\vec{\sigma}_{\mathrm{Ag}\setminus C}, \vec{\sigma}_C), s)) \models_p \varphi$ holds;
- For $\varphi \in \mathcal{L}_s(\mathrm{AP}, \mathrm{Ag})$, we have $(M, \lambda, \rho) \models_p \varphi$ if and only if $(M, \lambda, \rho[0]) \models_s \varphi$;

- For $\varphi \in \mathcal{L}_p(\mathrm{AP}, \mathrm{Ag})$, we have $(M, \lambda, \rho) \models_p \neg\varphi$ if and only if it is not the case that $(M, \lambda, \rho) \models_p \varphi$;

- For $\varphi, \psi \in \mathcal{L}_p(\mathrm{AP}, \mathrm{Ag})$, we have $(M, \lambda, \rho) \models_p \varphi \wedge \psi$ if and only if $(M, \lambda, \rho) \models_p \varphi$ and $(M, \lambda, \rho) \models_p \psi$;

- For $\varphi \in \mathcal{L}_p(\mathrm{AP}, \mathrm{Ag})$, we have $(M, \lambda, \rho) \models_p \boldsymbol{X}\, \varphi$ if and only if we have $(M, \lambda, \rho[1 \ldots]) \models_p \varphi$

- For $\varphi, \psi \in \mathcal{L}_p(\mathrm{AP}, \mathrm{Ag})$, we have $(M, \lambda, \rho) \models_p \varphi\, \mathbf{U}\, \psi$ if and only if there exists some j such that $(M, \lambda, \rho[j \ldots]) \models_p \psi$ and for all $0 \leq i < j$, we have $(M, \lambda, \rho[i \ldots]) \models_p \varphi$.

Given a concurrent game structure M and a labelling function $\lambda : \mathrm{St} \to \mathcal{P}\,(\mathrm{AP})$, we say that φ is **satisfiable** if there exists some state $s \in \mathrm{St}$ such that $(M, \lambda, s) \models \varphi$. Moreover, we say that φ is **equivalent** to ψ if $(M, \lambda, s) \models \varphi \iff (M, \lambda, s) \models \psi$ for all states $s \in \mathrm{St}$.

Note that LTL can been seen as the sublogic of ATL* given by all formulae $\mathbf{A}\, \varphi$, where the formula φ does not contain the "coalition" quantifiers $\langle\!\langle C \rangle\!\rangle$ or $[\![C]\!]$. Thus ATL* is a particularly effective tool for reasoning about the LTL properties that coalitions can achieve, and this is exactly how we shall use it in the proofs of the following theorems. Specifically,if we have an LTL game G, we can write $(G, s) \models \varphi$ as shorthand for $(M, \lambda, s) \models \varphi$, and furthermore, if the game G is apparent from the context, we shall drop it and simply write $s \models \varphi$ instead.

The relevant decision problem here is the *model-checking problem for ATL**, which we utilise heavily in the following section:

<u>ATL*-MODEL-CHECKING</u>:
Given: LTL G, ATL* formula φ, state $s \in \mathrm{St}$.
Question: Do we have $s \models \varphi$?

This problem is 2EXPTIME-complete [2] for games with two or more players, and PSPACE-complete for one player games (as then, the problem reduces to LTL satisfiability) [23]. The first of our own decision problems we will consider is FULFILLED COALITION, which we solve in the general case through a logical characterisation using ATL*.

Theorem 1. *FULFILLED COALITION is PSPACE-complete for one-player games, and it is 2EXPTIME-complete for games with more than one player.*

Proof. For membership we observe that given a game $G = (M, \gamma_1, \ldots, \gamma_n)$ and a coalition $C \subseteq \mathrm{Ag}$, it is the case that C is fulfilled if and only if $s^0 \models \langle\!\langle C \rangle\!\rangle \bigwedge_{i \in C} \gamma_i$ holds. By appealing to ATL*-MODEL-CHECKING, the two upper bounds immediately follow. For the lower bounds, we can reduce the problem of checking for the existence of a winning strategy in a two-player game with LTL goals as defined in [24] for 2EXPTIME-hardness and existential LTL model checking for PSPACE-hardness [23]. □

Fulfilled coalitions give an indication of which stable coalitions may form, but are insufficient to characterise the core, and therefore, to check E-CORE and A-CORE properties of a multi-agent system. To do this, we follow a different strategy and show that these two decision problems are, in general, also 2EXPTIME-complete.

Theorem 2. *E-CORE and A-CORE are PSPACE-complete for one-player games and 2EXPTIME-complete for games with more than one player.*

Proof. Let us consider E-CORE first. For membership we observe that given a game G and an LTL formula φ, it is the case that $(G, \varphi) \in$ E-CORE if and only if $s^0 \models \varphi_{\text{E-CORE}}(G, \varphi)$ holds, such that $\varphi_{\text{E-CORE}}(G, \varphi)$ is the following ATL* formula:

$$\bigvee_{W \subseteq \mathrm{Ag}} \left(\langle\!\langle \mathrm{Ag} \rangle\!\rangle \left(\varphi \wedge \bigwedge_{i \in W} \gamma_i \wedge \bigwedge_{j \in \mathrm{Ag} \setminus W} \neg \gamma_j \right) \wedge \bigwedge_{L \subseteq \mathrm{Ag} \setminus W} [\![L]\!] \bigvee_{j \in L} \neg \gamma_j \right)$$

which states that there is a path in G that satisfies φ as well as the goals of a set of players W (the "winners"), and that for every subset of players L that do not get their goals achieved in such a path (the "losers"), it is not the case that those players have a beneficial deviation from the path. As before, we can use the ATL*-MODEL-CHECKING problem to obtain the upper bounds; here we need to call the a 2EXPTIME algorithm an exponential number of times — one for each coalition of winners, and accept if any single one of them accepts. For the lower bounds, it is sufficient to show that $(G, \varphi) \in$ E-CORE if and only if $(G, \{1\}) \in$ FULFILLED COALITION, whenever $\gamma_1 = \varphi$ and $\gamma_j = \neg\varphi$, for every $j \in \mathrm{Ag} \setminus \{1\}$, which can be proved using Lemma 1. Since this is true even if Ag is a singleton set, both lower bounds follow from Theorem 1, that is, PSPACE-hardness in case of one-player games, and 2EXPTIME-hardness even for two-player games.

Finally, for A-CORE, we observe that $(G, \varphi) \notin$ A-CORE if and only if $(G, \neg\varphi) \in$ E-CORE. Since both PSPACE and 2EXPTIME are deterministic complexity classes, we can conclude that A-CORE is PSPACE-complete if $|\mathrm{Ag}| = 1$ and 2EXPTIME-complete if $|\mathrm{Ag}| > 1$, as it is for E-CORE. □

We now study CORE MEMBERSHIP and BENEFICIAL DEVIATION. For these two problems we first need to define how we will represent outcomes, as at present they are defined as infinite-state objects that map finite histories to players' actions. Following standard practice in the concurrent games literature, we model strategies as finite state machines with output (transducers) [3, 4]. Note that, for players with LTL goals, such strategies are sufficient: no more powerful model of strategies is necessary [3, 4]. Formally, a strategy for player i is a structure $\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$, where

- Q_i is a finite, non-empty set of **strategy states**;
- $q_i^0 \in Q_i$ is the **initial** strategy state;

- $\delta_i : Q_i \times \mathrm{St} \rightarrow Q_i$ is a **transition function**;
- $\tau_i : Q_i \rightarrow \mathrm{Ac}_i$ is an **output function**.

A machine strategy works as follows — it begins in the initial state, q_i^0, and produces an action based on this, $\tau(q_i^0)$. The state of the game then follows the transition function into a new state. Based on this new (game) state, and the state of the strategy, the strategy then also moves into a new state, based on the strategy transition function. This process repeats, yielding a new action at each timestep. By the pidgeonhole principle, it is easy to see that a strategy profile consisting solely of machine strategies will be eventually periodic.

With this definition in place, we can now establish the complexity of CORE MEMBERSHIP and BENEFICIAL DEVIATION. Formally, we have the following result.

Theorem 3. *CORE MEMBERSHIP is PSPACE-complete for one-player games and 2EXPTIME-complete for games with more than one player.*

Proof. For membership we first compute the winners and losers with respect to $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$, the outcome of the game. This can be done in PSPACE (it is equivalent to LTL model checking over a "product automata" or "concurrent program" [25]). Once we have computed W, we can check, for every $L \subseteq \mathrm{Ag} \setminus W$, whether L has a beneficial deviation. This is true if and only if L is a fulfilled coalition. Because this can be checked in PSPACE for one-player games and in 2EXPTIME for games with more than one player, the two upper bounds immediately follow. For the lower bounds, we use Lemma 1 and Theorem 1 again. Consider the following game. Let φ be a satisfiable LTL formula and $\vec{\sigma}$ an outcome that does not satisfy φ. Then, $(G, \vec{\sigma}) \in$ CORE MEMBERSHIP if and only if $(G, \{1\}) \notin$ FULFILLED COALITION, whenever $\gamma_1 = \varphi$ and $\gamma_j = \neg\varphi$, for every player $j \in \mathrm{Ag} \setminus \{1\}$. □

Let us now consider BENEFICIAL DEVIATION. This is the only "easy" problem for multi-player games, as it can be solved in PSPACE. To show this, we again need to find a different proof strategy. Consider any input instance $(G, \vec{\sigma}, \vec{\sigma}'_C)$ of the problem. We observe that, because $\vec{\sigma}'_C$ is fixed, we can make it part of the arena where the game is played, and then check if players not in C have a joint strategy for $\bigvee_{j \in C} \neg\gamma_j$. Due to the definition of beneficial deviation, we also need to check if $\rho(\vec{\sigma}) \models \bigwedge_{j \in} \neg\gamma_j$ holds or not.

In other words, the reason why this problem can be solved in PSPACE for multi-player games, unlike all other decision problems we have studied so far (which, in general, can be solved in doubly exponential time), is that this decision problem can be reduced to a one-player game (given by coalition $\mathrm{Ag} \setminus C$) with an LTL goal (given by $\gamma_{\mathrm{Ag}\setminus C} = \bigvee_{j \in C} \neg\gamma_j$) over a "product arena" (denoted by M_C) built from a concurrent game structure M and the joint strategy $\vec{\sigma}'_C$ that we want to check.

Theorem 4. *BENEFICIAL DEVIATION is PSPACE-complete, even for one-player games.*

Proof. Checking that $\rho(\vec{\sigma}) \models \bigwedge_{j \in C} \neg\gamma_j$ holds can be done in PSPACE. Again, this is equivalent to model checking LTL formulae over a "product automata" or "concurrent program" [25]. If the statement does not hold, then, by definition, $\vec{\sigma}'_C$ is not a beneficial deviation, as at least one player in C already has its goal satisfied by $\vec{\sigma}$. If the statement holds, then we check that $\rho(\vec{\sigma}'_{-C}, \vec{\sigma}'_C) \models \bigwedge_{j \in C} \gamma_j$ holds, for all joint strategies $\vec{\sigma}'_{-C}$ for players not in C. We do this in PSPACE by checking whether it is not the case that $(M_C, \lambda', s^{0'}) \models \bigvee_{j \in C} \neg\gamma_j$ holds, where $M_C = (\mathrm{Ag}', \mathrm{Ac}', \mathrm{St}', s^{0'}, \mathrm{tr}')$ is the concurrent game structure defined as follows:

- $\mathrm{Ag}' = \{0\}$, $\mathrm{Ac}' = \Pi_{i \in \mathrm{Ag}\setminus C} \mathrm{Ac}_i$;
- $\mathrm{St}' = \mathrm{St} \times \Pi_{j \in C} Q_j$;
- $s^{0'} = (s^0, q^0_x, \ldots, q^0_y)$, such that $\sigma_z = (Q_z, q^0_z, \delta_z, \tau_z)$, $\vec{\sigma}'_C = (\sigma_x, \ldots, \sigma_y)$, and $z \in \{x, \ldots, y\}$;

- $\text{tr}'((s, q_x, \ldots, q_y), (a, \ldots, b)) = (s', q'_x, \ldots, q'_y)$ such that
    - $s' = \text{tr}(s, \tau(q_x), \ldots, \tau(q_y), a, \ldots, b)$, and
    - $q'_z = \delta(q_z, s)$, with $z \in \{x, \ldots, y\}$.

and λ' is defined as,

$$\lambda'(s, q_x, \ldots, q_y) = \lambda(s).$$

In other words, M_C transitions just like M save that it is restricted to the behaviour already defined by $\vec{\sigma}'_C$.

For the lower bound we use LTL model checking. □

In addition to the above complexity results, we also have two model-theoretic results, one ensuring that the core is never empty and another one stating that checking whether an LTL formula is satisfied by some outcome in the core is a bisimulation-invariant property[2]. The latter result is easy, and follows directly from the membership proof of E-Core.

Corollary 1. *Let $G = (M, \gamma_1, \ldots, \gamma_n)$ be a game, φ be an LTL formula, and M' be a concurrent game structure that is bisimilar to M. Then, $(G, \varphi) \in$ E-Core if and only if $(G', \varphi) \in$ E-Core, where $G' = (M', \gamma_1, \ldots, \gamma_n)$.*

Proof. Because ATL* is a bisimulation-invariant temporal logic, and the core can be characterised in ATL* using $\varphi_{\text{E-Core}}$, as defined in the membership proof of E-Core. More specifically, it follows from the fact that $(M, \lambda, s^0) \models \varphi_{\text{E-Core}}(G, \varphi)$ if and only if $(M', \lambda', s^{0\prime}) \models \varphi_{\text{E-Core}}(G', \varphi)$. □

To finish this section, we show an important property of the core, namely, that it is never empty.

[2]The reader is referred to [21] for the definition of bisimulation-invariance over the model of concurrent game structures.

Theorem 5. *$core(G) \neq \emptyset$, for every game G.*

Proof. Take any run ρ in the game $G = (M, \gamma_1, \ldots, \gamma_n)$. Either $\rho \models_M \bigwedge_{i \in \mathrm{Ag}} \gamma_i$ holds or not. If the former, then the core is not empty: every strategy $\vec{\sigma}$ such that $\rho = \rho(\vec{\sigma})$ is in the core. If the latter, then there is a set of players L_1 that do not get their goals achieved in ρ. If no subset of L_1 is fulfilled, then, again, every outcome $\vec{\sigma}$ such that $\rho = \rho(\vec{\sigma})$ is in the core, since no set of losers would be able to beneficially deviate. Otherwise, there is a set of players $C_1 \subseteq L_1$ that have a joint strategy $\vec{\sigma}_{C_1}$ such that $\rho(\vec{\sigma}_{C_1}, \vec{\sigma'}_{-C_1}) \models \bigwedge_{i \in C_1} \gamma_i$, for all joint strategies $\vec{\sigma'}_{-C_1}$ for $\mathrm{Ag} \setminus C_1$.

Now, let $L_2 = \mathcal{L}(\vec{\sigma}_{C_1}, \vec{\sigma'}_{-C_1})$. Again, if no subset of L_2 is fulfilled, then $(\vec{\sigma}_{C_1}, \vec{\sigma'}_{-C_1}) \in core(G)$. Otherwise, there are players $C_2 \subseteq L_2$ that have a joint strategy $\vec{\sigma}_{C_2}$ such that

$$\rho(\vec{\sigma}_{C_1}, \vec{\sigma}_{C_2}, \vec{\sigma'}_{-(C_1 \cup C_2)}) \models \bigwedge_{i \in C_1 \cup C_2} \gamma_i$$

for all joint strategies $\vec{\sigma'}_{-(C_1 \cup C_2)}$ for $\mathrm{Ag} \setminus (C_1 \cup C_2)$.

Reasoning recursively, we can repeat this process, taking an arbitrary strategy $\vec{\sigma'}_{-\bigcup_{1 \leq ik} C_i}$, and defining,

$$L_k = \mathcal{L}\left(\vec{\sigma}_{C_1}, \ldots, \vec{\sigma}_{C_{k-1}}, \vec{\sigma'}_{-\bigcup_{1 \leq ik} C_i}\right),$$

as long as no subset of L_{k-1} is fulfilled. Since we have $L_k \subset L_{k-1}$, there are two possibilites. The first is that there exists some k such that no subset of L_k is fulfilled, in which case, any strategy profile of the form,

$$\left(\vec{\sigma}_{C_1}, \ldots, \vec{\sigma}_{C_{k-1}}, \vec{\sigma'}_{-\bigcup_{1 \leq ik} C_i}\right),$$

with $\vec{\sigma'}_{-\bigcup_{1 \leq ik} C_i}$ chosen arbitrarily, is in the core. The second possibility is that there exists some k such that L_k is empty. In this situation, the strategy profile which defines L_k models every agent's goal, and is therefore necessarily in the core. $\square$

Theorem 5, ensuring that the core is never empty, can be used to strengthen numeral 3 of Lemma 1.

Corollary 2. *For every game G and fulfilled coalition C, there is $\vec{\sigma} \in core(G)$ such that $C \subseteq \mathcal{W}(\vec{\sigma})$.*

5. On Credible Coalition Formation

As we noted above, our definition of the core assumes worst-case reasoning: a deviation must be beneficial against *all* counter-responses. This definition is robust in the sense that any core-stable outcome is stable in a very strong sense, but one could argue that in some cases it is *too* strong. In particular, when a coalition C is contemplating a deviation $\vec{\sigma}_C$, it can surely assume that the remaining players will not act against their own interests. Thus, one could argue that a deviation need not be beneficial for *all* behaviours of the remaining players, but only those behaviours that are *credible*, in the sense that the remaining players might rationally choose them according to their own preferences.

To make this discussion concrete, consider a two-player game G, with a start state, s_0 and three sink states, s_1, s_2, and s_3. Each player has two actions available to them, a and b, and the transition function from the start state is defined as follows:

$$\text{tr}(s_0, (a, a)) = s_1,$$

$$\text{tr}(s_0, (a, b)) = s_2,$$

$$\text{tr}(s_0, (b, a)) = s_3,$$

$$\text{tr}(s_0, (b, b)) = s_3$$

Thus the game is as illustrated in Figure 5. Additionally, suppose that the infinite run that ends up in s_1 is the one run which satisfies player one's goal, and the run that ends up in s_2 is the one which satisfies player two's goal. Now, in this game, the run which ends up in s_1 lies in the core, but with the use of a non-credible (punishing) strategy by player 1. Notice that the only possible deviation from (a, a) for player 2 is to play b, to which player 1 could respond by also playing b. Although this behaviour would prevent player 2 from achieving its goal, such a way of playing can be regarded as not rational for player 1 given their preference relation: player 1 certainly prefers the run which ends in s_1 over the other two possible runs, but is indifferent otherwise.

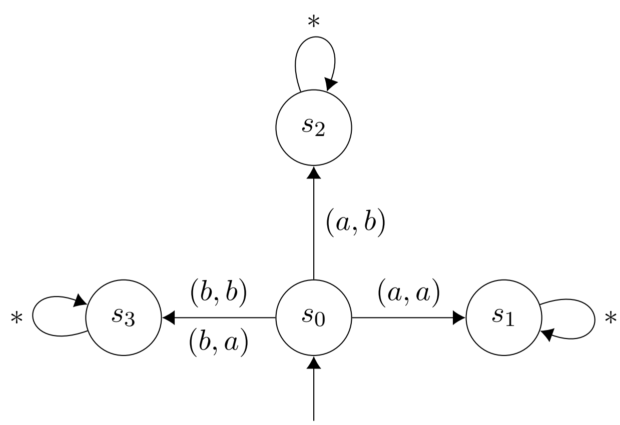

Figure 1: A game with a non-credible strategy.

Motivated by this phenomenon, we propose a stronger definition for the core in which the way that deviating players are punished is more credible. More specifically, with this new definition we require that if a coalition C wants to deviate from a given strategy profile, then the remaining players can only credibly threaten C when they have a counter-response in which both at least one player in C does not get its goal achieved and every winner in the original strategy profile remains a winner in the new one, i.e., the counter-coalition act in accordance with their preference relations. We then reformulate the definition of a beneficial deviation and say that a deviation $\vec{\sigma}'_C$ is a *beneficial deviation from* $\vec{\sigma}$ if:

1. $C \subseteq \mathcal{L}(\vec{\sigma})$, and
2. $C \subseteq \mathcal{W}(\vec{\sigma}_{-C}, \vec{\sigma'}_C)$, and
3. for every joint strategy $\vec{\sigma}'_{-C}$ for $\mathrm{Ag} \setminus C$, we have $\mathcal{W}(\vec{\sigma}) \subseteq \mathcal{W}(\vec{\sigma}'_{-C}, \vec{\sigma}'_C) \Rightarrow C \subseteq \mathcal{W}(\vec{\sigma}'_{-C}, \vec{\sigma}'_C)$.

With this definition in place we can say that the **strong core** of a game (CORE$^+$), denoted $core^+(G)$, is the set of outcomes of G that admit no beneficial deviation as above. Then, we see that while $\vec{\sigma}_{a_1 a_1}$ is in $core(G)$, it is not the case that $\vec{\sigma}_{a_1 a_1}$ is in $core^+(G)$, since player 2 can now beneficially deviate from $\vec{\sigma}_{a_1 a_1}$ to $\vec{\sigma}_{a_1 b_1}$.

Note on credible threats in games with externalities. The game theory literature on this topic is vast. The reason is that the existence of externalities leads to many different definitions of stable behaviour (see, *e.g.*, [12, 19, 26, 27] for many variants of the core). Here, we propose one definition but by no means we claim it is the strongest anyone may wish to consider. Essentially, with our definition, we require that for a punishing joint strategy to be credible, winners must remain winners after the presenting the threat.

We will now study the complexity of the decision problems defined in previous sections, but with respect to CORE$^+$. There are four decision problems whose definition depends on the nature of the core: E-Core, A-Core, Core Membership, and Beneficial Deviation. To simplify notations, we will call them here in the same way but with the understanding that results in this section are with respect to CORE$^+$. As we will show next, these four problems have the same complexities as with core, but require a more complex logical characterisation, which we provide here using the two-alternation fragment of Strategy Logic (SL [28, 29])[3].

[3] A logical characterisation of CORE$^+$ using ATL* was not found. In fact, we believe that such a logical characterisation in ATL* is not possible for multi-player games.

SL extends LTL with two **strategy quantifiers**, $\langle\langle x\rangle\rangle$ and $[[x]]$, and an **agent binding** operator (i, x), where i is an agent and x is a variable. These operators can be read as *"there exists a strategy x"*, *"for every strategy x"*, and *"bind agent i to the strategy associated with variable x"*, respectively. Formally, SL formulae are inductively built from a set of propositions AP, variables Var, and agents Ag, using the following grammar, where $p \in \mathrm{AP}$, $x \in \mathrm{Var}$, and $i \in \mathrm{Ag}$:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \boldsymbol{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \mid \langle\langle x\rangle\rangle\varphi \mid [[x]]\varphi \mid (i, x)\varphi.$$

We can now present the semantics of SL, where Str denotes the set of all strategies. Given a concurrent game structure M, for all SL formulae φ, states $s \in \mathrm{St}$ in M, and assignments $\chi \in \mathrm{Asg} = (\mathrm{Var} \cup \mathrm{Ag}) \to \mathrm{Str}$, mapping variables and agents to strategies, the relation $M, \chi, s \models \varphi$ is defined as follows:

1. For the Boolean and temporal cases, the semantics is standard
2. For all formulae φ and variables $x \in \mathrm{Var}$ we have:
   (a) $M, \chi, s \models \langle\langle x\rangle\rangle\varphi$ if $\exists f \in \mathrm{Str}.\ M, \chi[x \mapsto f], s \models \varphi$;
   (b) $M, \chi, s \models [[x]]\varphi$ if $\forall f \in \mathrm{Str}.\ M, \chi[x \mapsto f], s \models \varphi$;
3. For every agent $i \in \mathrm{Ag}$ and variable $x \in \mathrm{Var}$,
   if $M, \chi[i \mapsto \chi(x)], s \models \varphi$ then $M, \chi, s \models (i, x)\varphi$.

For a sentence φ, that is, a formula with no free variables and agents [28, 29], we say that M satisfies φ, and write $M \models \varphi$ in that case, if $M, \emptyset, s^0 \models \varphi$, where $\emptyset$ is the empty assignment. We use the following abbreviations: $\langle i\rangle\varphi$ for $\langle\langle x\rangle\rangle(i, x)\varphi$ and $[i]\varphi$ for $[[x]](i, x)\varphi$, which can be intuitively understood as "there is a strategy for agent i such that φ holds" and "φ holds, for all strategies of agent i", respectively. We extend this notation to sets of players and write, for instance, $\langle C\rangle\varphi$ instead of $\langle i\rangle \ldots \langle j\rangle\varphi$, where $C = \{i, \ldots, j\}$, and similarly for the universal quantifier operator. Then, with $\langle C\rangle\varphi$ we mean that *"coalition C has a joint strategy such that φ holds."*

We then find that for a game $G = (M, \gamma_1, \ldots, \gamma_n)$ and LTL formula φ, we have $(G, \varphi) \in$ E-CORE if and only if $M \models \varphi^{+}_{\text{E-CORE}}(G, \varphi)$, where $\varphi^{+}_{\text{E-CORE}}(G, \varphi)$ is the SL formula:

$$\varphi^{+}_{\text{E-CORE}}(G, \varphi) = \bigvee_{W \subseteq \mathrm{Ag}} \langle \mathrm{Ag} \rangle \left(\varphi \wedge \bigwedge_{i \in W} \gamma_i \wedge \bigwedge_{j \in \mathrm{Ag} \setminus W} \neg\gamma_j \wedge \bigwedge_{C \subseteq \mathrm{Ag} \setminus W} \varphi_{\mathrm{NoBD}}(G, W, C) \right)$$

$$\varphi_{\mathrm{NoBD}}(G, W, C) = [C] \left(\bigwedge_{j \in C} \gamma_j \rightarrow \langle \mathrm{Ag} \setminus C \rangle \left(\bigwedge_{i \in W} \gamma_i \wedge \bigvee_{j \in C} \neg\gamma_j \right) \right)$$

This SL formula expresses that in the concurrent game structure, there exists a path $\langle \mathrm{Ag} \rangle (\ldots)$ under which

1. The formula φ holds;
2. Some players get their goals achieved: $\bigwedge_{i \in W} \gamma_i$;
3. The remaining players do not: $\bigwedge_{j \in \mathrm{Ag} \setminus W} \neg\gamma_j$;
4. No coalition of losers has a beneficial deviation: $\bigwedge_{C \subseteq \mathrm{Ag} \setminus W} \varphi_{\mathrm{NoBD}}(G, W, C)$.

We express the condition of a coalition of losers C not having a beneficial deviation with the SL formula $\varphi_{\mathrm{NoBD}}(G, W, C)$; this is broken down as follows: for every joint strategy of C, if every player in C is better off $\left(\bigwedge_{j \in C} \gamma_j\right)$, then the coalition of players outside C have a joint strategy $(\langle \mathrm{Ag} \setminus C \rangle \ldots)$ such that both the winners in the original outcome remain winners after the threat is presented $\left(\bigwedge_{i \in W} \gamma_i\right)$, and at least one player in the deviating coalition, C, does not get its goal achieved $\left(\bigvee_{j \in C} \neg\gamma_j\right)$.

At this point, we would like to make a couple of observations. First, that the complexity of checking SL formulae is non-elementary and depends on the alternation-depth of the formula ([28]): SL formulae of alternation-depth n can be checked in $(n+1)$-EXPTIME, and in PSPACE for formulae that are semantically equivalent to CTL* formulae. Since $\varphi^{+}_{\text{E-CORE}}(G, \varphi)$ is an SL formula with two alternations, it can be checked in 3EXPTIME (and in PSPACE if $|\mathrm{Ag}| = 1$). Second, we also would like to recall that finite-state machine strategies, as those we use here, can be characterised in LTL using the technique presented in [3, 4]. Using these logical characterisations, we can show the following complexity results.

Theorem 6. *For multi-player games, while E-CORE and A-CORE are in 3EXPTIME, CORE MEMBERSHIP is 2EXPTIME-complete and BENEFICIAL DEVIATION is PSPACE-complete. For one-player games, all problems are PSPACE-complete.*

Because CORE$^+$ was characterised using SL (which is not a bisimulation-invariant logic), we cannot conclude that the satisfaction of LTL properties by outcomes in $core^+$ is a bisimulation-invariant property. We believe that this is not the case.

In the α-core setting, we saw that the core is always non-empty. Thus, a natural question is to ask whether $core^+$ is always non-empty. We begin by showing that this is the case for games with three or fewer players:

Theorem 7. *For every two-player and three-player game, G, we have $core^+(G) \neq \emptyset$.*

Proof. First, consider games with only two players. For a contradiction, let us suppose that for some game G, the set of outcomes $core^+(G)$ is empty. This means that for every outcome either player 1 or player 2 or both have a beneficial deviation. Then, we know that no outcome can satisfy both goals, γ_1 and γ_2. Let us then consider the three remaining possible cases: outcomes that only satisfy γ_1 (case 1), outcomes that only satisfy γ_2 (case 2), and outcomes that satisfy neither γ_1 nor γ_2 (case 3). Let $\vec{f} = (f_1, f_2)$ be an outcome, f'_1 be a deviation by player 1, and f'_2 be a deviation by player 2, and consider the three cases above. In case 1, only player 2 would deviate. Then, outcome (f_1, f'_2) only satisfies γ_2. Because (f_1, f'_2) is not in the core either, from this outcome only player 1 would deviate, to another outcome (f'_1, f'_2). Then, outcome (f'_1, f'_2) only satisfies γ_1. But, then, we have a contradiction, since this means that (f_1, f_2) would be in $core^+(G)$. We can reason symmetrically to show that case 2 is not possible either. For case 3 we note that only single deviations would be possible. But any such deviations would be to an outcome that either only satisfies γ_1 or only satisfies γ_2, which are no longer possible. Since no other cases are possible, we have to reject our assumption and conclude that, for two-player games, $core^+(G)$ is not empty.

For three player games, the proof is similar, but requires a careful case-by-case analysis. □

For games with four or more players, the outlook is not as good — there exist games with a non-empty core. We wrote a Python program to search the space of four player games via dynamic programming, testing each one for an empty core. Whilst the algorithm was capable of producing false negatives, it also produced multiple examples of four player games with empty cores — the following example is one of these, which we also checked manually.

Example 2. Consider the following 4 player game, with a start state, s^0, and six sink states, $s^1, \ldots, s^6$. Let each player have two actions each, $\{0, 1\}$, and writing $a_1a_2a_3a_4$ for the action (a_1, a_2, a_3, a_4), our transition function from the start state looks like the following:

$$\begin{aligned}
&\text{tr}(0000) = s^1, && \text{tr}(0001) = s^1,\\
&\text{tr}(0010) = s^2, && \text{tr}(0011) = s^2,\\
&\text{tr}(0100) = s^1, && \text{tr}(0101) = s^3,\\
&\text{tr}(0110) = s^2, && \text{tr}(0111) = s^5,\\
&\text{tr}(1000) = s^6, && \text{tr}(1001) = s^4,\\
&\text{tr}(1010) = s^4, && \text{tr}(1011) = s^4,\\
&\text{tr}(1100) = s^1, && \text{tr}(1101) = s^3,
\end{aligned}$$

$$\text{tr}(1110) = s^4, \qquad \text{tr}(1111) = s^3.$$

Moreover, suppose player one prefers the runs that end up in the states s^1, s^2, s^3, player two prefers those that end up in s^1, s^4, s^5, player three s^2, s^4, s^6, and player four s^3, s^5, s^6.

Thus, our game looks like the following graph:

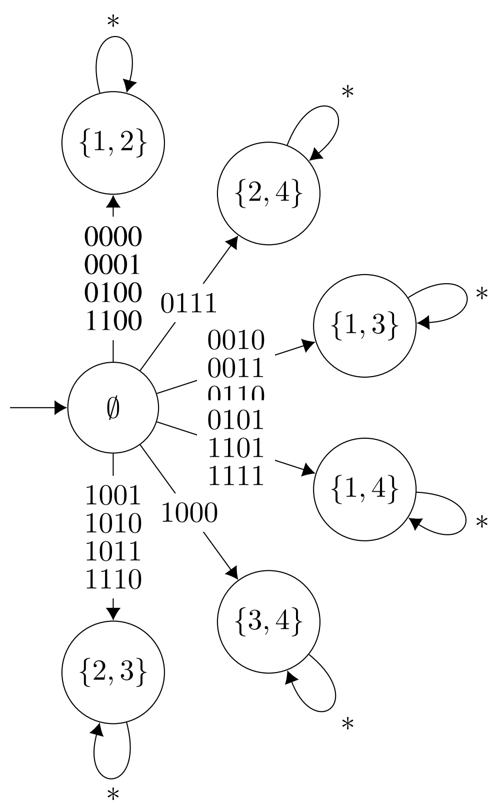

Figure 2: A game with an empty CORE$^+$.

With a careful case-by-case analysis, one can verify that for every state, there exists some coalition with a beneficial deviation. Thus, this game has an empty CORE$^+$.

6. Mean-Payoff Games

Thus far, we have considered games with *qualitative* preferences — each player has a goal given by some temporal logic formula, which, under a given run, is either satisifed or unsatisfied. But this is arguably quite a coarse-grained approach to specifying agent goals; it does not offer a way of expressing the *intensity* of the individual player's preferences. One way around this is to introduce multiple LTL goals for each player, and define some mapping from satisfied formulae to the real numbers [30–32]. However, an arguably more natural (and computationally amenable) approach is to sidestep temporal logics entirely and put assign weights to states, rather than atomic propositions. We then assign a *mean-payoff* to runs, with the idea that agents prefer runs which maximise their mean-payoff [33–35].

Formally, a mean-payoff game, G, is a tuple,

$$G = \left(M, \{w_i\}_{i \in \mathrm{Ag}}\right),$$

where M is a concurrent game structure, and for each $i \in \mathrm{Ag}$, $w_i : \mathrm{St} \to \mathbb{Z}$ is a **weight** function, mapping states to integers. Games are played in an identical way to the LTL setting, but the agents' preference relations are defined differently here. Let $\beta \in \mathbb{R}^\omega$ be an infinite sequence of real numbers. Then the mean-payoff of β, denoted by $\mathsf{mp}(\beta)$, is defined by the following expression:

$$mp(\beta) = \liminf_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} \beta_i.$$

In a mean-payoff game, a run of states, $\pi = s^0 s^1 \ldots$ induces an infinite sequence of weights for each player, $w_i(s^0)w_i(s^1)\ldots$ — we denote this sequence by $w_i(\pi)$ and for notational convenience, we will write $\mathsf{pay}_i(\pi)$ for $\mathsf{mp}(w_i(\pi))$. With this, we are ready to define the preference relation for each player — given two runs, ρ and ρ', we say that player i *prefers* ρ to ρ', and write $\rho \succeq_i \rho'$ if we have $\mathsf{pay}_i(\rho) \geq \mathsf{pay}_i(\rho)$.

Now, our definition of the core in the setting of LTL games relies on the notion of *winners* and *losers* of a game. As we had dichotomous preferences in this setting, this was a reasonable way of defining the core. However, in the mean-payoff setting, it does not make sense to classify players as either winners or losers — they can receive a wide spectrum of payoffs. Thus, we need a more general definition of what a beneficial deviation is. We say that given a game G, a strategy profile $\vec{\sigma}$, a *beneficial deviation* by a coalition C, is a strategy vector $\vec{\sigma}'_C$ su ch that for all complementary strategy profiles $\vec{\sigma}'_{\mathrm{Ag}\setminus C}$, we have $\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\mathrm{Ag}\setminus C}) \succ_i \rho(\vec{\sigma})$ for all $i \in C$. We can then say that $\vec{\sigma}$ is a member of the core, if there exists no coalition C which has a beneficial deviation from $\vec{\sigma}$. Note this formulation subsumes our earlier definition of the core, and allows us to reason about this solution concept in our numeric setting.

We begin by asking whether mean-payoff games always have a non-empty core, a property that holds for games with LTL goals and specifications. We find that this does not hold in general for mean-payoff games.

Theorem 8. *In mean-payoff games, if* $|\mathrm{Ag}| \leq 2$*, then the core is non-empty. For* $|\mathrm{Ag}| > 2$*, there exist games with an empty core.*

Proof. If $|\mathrm{Ag}| = 1$, it is straighforward to see that the core is always non-empty; we use Karp's algorithm for determining the maximum cycle in a weighted graph [36] to determine the maximum payoff that one player can achieve. For two-player games, let $\vec{\sigma} = (\sigma_1, \sigma_2)$ be any strategy profile. If $\vec{\sigma}$ is not in the core, then either Player 1, or Player 2, or the coalition consisting of both players has a beneficial deviation. If the latter is true, then there is a strategy profile, $\vec{\sigma}' = (\sigma'_1, \sigma'_2)$ such that $\vec{\sigma}' \succ_i \vec{\sigma}$ for both $i \in \{1, 2\}$. We repeat this process until the coalition of both players does not have a beneficial deviation. This must eventually be the case as each player's payoff is capped by their maximum weight, so either they both reach their corresponding maximum weight, or there comes a point when they cannot beneficially deviate together. At this point, we must either be in the core, or either player 1 or player 2 has a beneficial deviation. If player j ($j \in \{1, 2\}$) has a beneficial deviation, say σ_j, then any strategy profile (σ_j, σ_i), with $i \neq j$, that maximises Player i's mean-payoff is in the core. Thus, for every two-player game, there exists some strategy profile that lies in the core.

However, for mean-payoff games with three or more players, the core of a game may be empty. Consider the following three-player game G, where each player has two actions, **H**, **T**, and there are four states, P, R, B, Y. The states are weighted for each player as follows:

$w_i(s)$	1	2	3
P	-1	-1	-1
R	2	1	0
B	0	2	1
Y	1	0	2

If the game is in any state other than P, then no matter what set of actions is taken, the game will remain in that state. Thus, we only specify the transitions for the state P:

Ac	St
$(\mathbf{H}, \mathbf{H}, \mathbf{H})$	R
$(\mathbf{H}, \mathbf{H}, \mathbf{T})$	R
$(\mathbf{H}, \mathbf{T}, \mathbf{H})$	B
$(\mathbf{H}, \mathbf{T}, \mathbf{T})$	P
$(\mathbf{T}, \mathbf{H}, \mathbf{H})$	P
$(\mathbf{T}, \mathbf{H}, \mathbf{T})$	Y
$(\mathbf{T}, \mathbf{T}, \mathbf{H})$	B
$(\mathbf{T}, \mathbf{T}, \mathbf{T})$	Y

Thus the structure of this game looks like the following graph:

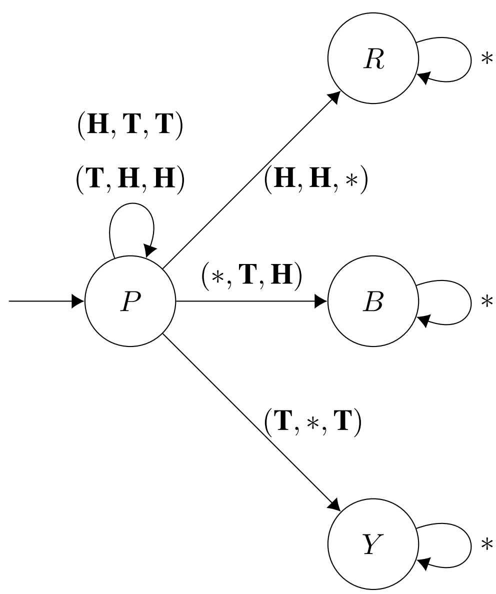

Figure 3: A game with an empty core.

Note that strategies are characterised by the state that the game eventually ends up in. If the players stay in P forever, then they can all collectively change strategy to move to one of R, B, Y, and each get a better payoff. Now, if the game ends up in R, then players 2 and 3 can deviate by playing $(\mathbf{T}, \mathbf{H})$, and no matter what player 1 plays, they will be in state B, and will be better off. But similarly, if the game is in B, then players 1 and 3 can deviate by playing $(\mathbf{T}, \mathbf{T})$ to enter state Y, in which they both will be better off, regardless of what player 2 does. And finally, if in Y, then players 1 and 2 can deviate by playing $(\mathbf{H}, \mathbf{H})$ to enter R and will be better off regardless of what player 3 plays. Thus, no strategy profile lies in the core. □

With this established, we now turn our attention to the decision problems relating to the core in the mean-payoff setting. However, from a computational perspective, there is an immediate concern here — given a potential beneficial deviation, how can we verify that it is preferable to the status quo under *all* possible counter-responses? Given that strategies can be arbitrary mathematical functions, how can we reason about that universal quantification effectively? Fortunately, as we show in the following lemma, we can restrict our attention to memoryless strategies when thinking about potential counter-responses to players' deviations:

Lemma 2. *Let G be a game, $C \subseteq \mathrm{Ag}$ be a coalition and $\vec{\sigma}$ be a strategy profile. Further suppose that $\vec{\sigma}'_C$ is a strategy vector such that for all memoryless strategy vectors $\vec{\sigma}'_{\mathrm{Ag}\setminus C}$, we have,*

$$\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\mathrm{Ag}\setminus C}) \succ_i \rho(\vec{\sigma}).$$

Then, for all strategy vectors, $\vec{\sigma}'_{\mathrm{Ag}\setminus C}$, not necessarily memoryless, we have,

$$\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\mathrm{Ag}\setminus C}) \succ_i \rho(\vec{\sigma}).$$

Before we prove this, we need to introduce an auxillary concept of *two-player, turn-based, zero-sum, multi-mean-payoff games* [37] (we will just call these multi-mean-payoff games moving forward). Informally, these are similar to two-player, turn-based, zero-sum mean-payoff games, except player 1 has k weight functions associated with the edges, and they are trying to ensure the resulting k-vector of mean-payoffs is component-wise greater than a vector threshold. Formally, a multi-mean-payoff game is a 5-tuple, $G = (V_1, V_2, v^0, E, w, z^k)$, where V_1, V_2 are sets of states controlled by players 1 and 2 respectively, with $V = V_1 \cup V_2$ the state space, $v^0 \in V$ the start state, $E \subseteq V \times V$ a set of edges and $w : E \to \mathbb{Z}^k$ a weight function, assigning to each edge a vector of weights.

The game is played by starting in the start state, $s^0 \in S_i$, and player i choosing an edge (s^0, s^1), and traversing it to the next state. From this new state, $s^1 \in S_j$, player j chooses an edge and so on, repeating this process forever. Runs are defined in the usual way and the payoff of a run π, $\mathsf{pay}(\pi)$, is simply the vector $(\mathsf{mp}(w_1(\pi)), \ldots, \mathsf{mp}(w_k(\pi)))$. Finally, $z^k \in \mathbb{Q}^k$ is a threshold vector and player 1 wins if the $\mathsf{pay}_i(\pi) \geq z_i$ for all $i \in \{1, \ldots, k\}$, and loses otherwise. An important question associated with these games is whether player 1 can force a win. We can encapsulate this with the following decision problem:

MULTI-MEAN-PAYOFF-THRESHOLD:

Given: Multi-mean-payoff game G.

Question: Is it the case that player 1 has a winning strategy?

As shown in [37], this problem is co-NP-complete. Whilst we do not need to utilise this result right now, this sets us up to prove Lemma 2.

Proof of Lemma 2. Let $\vec{\sigma}_{\mathrm{Ag}\setminus C}$ be an arbitrary strategy and let $i \in C$ be an arbitrary agent. Denote $\rho(\vec{\sigma})$ by ρ and $\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\mathrm{Ag}\setminus C})$ by ρ'. We aim to show that $\rho' \succ_i \rho$. Suppose instead it is the case that $\rho \succeq_i \rho'$. Thus, we have $\rho(\vec{\sigma}) \succeq_i \rho(\vec{\sigma}'_C, \vec{\sigma}'_{\mathrm{Ag}\setminus C})$. Considering this as a two-player multi-mean-payoff game, where player 1's strategy is fixed and encoded into the game structure (*i.e.*, player 1 follows $\vec{\sigma}'_C$, but has no say in the matter), and the payoff threshold is $\mathsf{mp}(\rho(\vec{\sigma}))$, then $\vec{\sigma}'_{\mathrm{Ag}\setminus C}$ is a winning strategy for player 2 in this game. Now, by [37, 38], if player 2 has a winning strategy, then they have a memoryless winning strategy. Thus, there is a memoryless strategy $\vec{\sigma}''_{\mathrm{Ag}\setminus C}$ such that $\rho(\vec{\sigma}) \succeq_i \rho(\vec{\sigma}'_C, \vec{\sigma}''_{\mathrm{Ag}\setminus C})$. But this contradicts the assumptions of the lemma, and thus we must have $\rho' \succ_i \rho$. $\square$

We are now in a position to look at some complexity bounds for mean-payoff games in the cooperative setting. Let us begin by considering the following decision problem relating to beneficial deviations:

> BENEFICIAL-DEVIATION (BEN-DEV):
>
> *Given*: Game G and strategy profile $\vec{\sigma}$.
>
> *Question*: Is there $C \subseteq \mathrm{Ag}$ and $\vec{\sigma}'_C \in \Sigma_C$ such that for all $\vec{\sigma}'_{\mathrm{Ag}\setminus C} \in \Sigma_{\mathrm{Ag}\setminus C}$ and for all $i \in C$, we have:
>
> $$\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\mathrm{Ag}\setminus C}) \succ_i \rho(\vec{\sigma})?$$

Using this new problem, we can prove the following statement.

Lemma 3. *Let G be a game, and $\vec{\sigma}$ a strategy profile. Then, $(G, \vec{\sigma}) \in$ MEMBERSHIP if and only if $(G, \vec{\sigma}) \notin$ BEN-DEV.*

Proof. Proof follows directly from definitions. $\square$

The above lemma characterises the MEMBERSHIP problem for cooperative games in terms of beneficial deviations, and, in turn, provides a direct way to study its complexity. In the remainder of this section we concentrate on the memoryless case.

Theorem 9. *For memoryless strategies, BEN-DEV is* NP-complete.

Proof. First correctly guess a deviating coalition C and a strategy profile $\vec{\sigma}'_C$ for such a coalition of players. Then, use the following three-step algorithm. First, compute the mean-payoffs that players in C get on $\rho(\vec{\sigma})$, that is, a set of values

$z_j^* = \mathsf{pay}_j(\rho(\vec{\sigma}))$ for every $j \in C$ — this can be done in polynomial time simply by 'running' the strategy profile $\vec{\sigma}$. Then compute the graph $G[\vec{\sigma}'_C]$, which contains all possible behaviours (*i.e.*, strategy profiles) for $\mathrm{Ag} \setminus C$ with respect to $\vec{\sigma}$ — this construction is similar to the one used in the proof of Theorem 4, that is, the game when we fix $\vec{\sigma}'_C$, and can be done in polynomial time. Finally, we ask whether every path π in $G[\vec{\sigma}'_C]$ satisfies $\mathsf{pay}_j(\pi) > z_j^*$, for every $j \in C$ — for this step, we can use Karp's algorithm [36] to answer the question in polynomial time for every $j \in C$. If every path in $G[\vec{\sigma}'_C]$ has this property, then we accept; otherwise, we reject.

For hardness, we reduce from 3SAT, using a small variation of the construction in [23]. Let $P = \{x_1, \dots, x_n\}$ be a set of atomic propositions. Given a Boolean formula $\varphi = \bigwedge_{1 \le c \le m} C_c$ (in conjunctive normal form) over P — where each $C_c = l_{c1} \vee l_{c2} \vee l_{c3}$, and each literal $l_{ck} = x_j$ or $\neg x_j$, with $1 \le k \le 3$, for some $1 \le j \le n$ — we construct $M = \Big(\mathrm{Ag}, \mathrm{St}, s^0, (\mathrm{Ac}_i)_{i \in \mathrm{Ag}}, \mathrm{tr}\Big)$, an m-player concurrent game structure defined as follows, and illustrated in Figure 4.:

- $\mathrm{Ag} = \{1, \dots, m\}$;
- $\mathrm{St} = \{x_v \mid 1 \le v \le n\} \cup \{x'_v \mid 1 \le v \le n\} \cup \{y_0, y_n, y^0, y^*\}$;
- $s^0 = y^0$;
- $\mathrm{Ac}_i = \{t, f\}$, for every $i \in \mathrm{Ag}$, and $\mathrm{Ac} = \mathrm{Ac}_1 \times \cdots \times \mathrm{Ac}_m$;
- For tr, refer to the Figure 4, such that $T = \{(t_1, \dots, t_m)\}$ and $F = \mathrm{Ac} \setminus T$.

With M at hand, we build a mean-payoff game using the following weight function:

- $w_i(x_v) = 1$ if x_v is a literal in C_i and $w_i(x_v) = 0$ otherwise, for all $i \in \mathrm{Ag}$ and $1 \le v \le n$

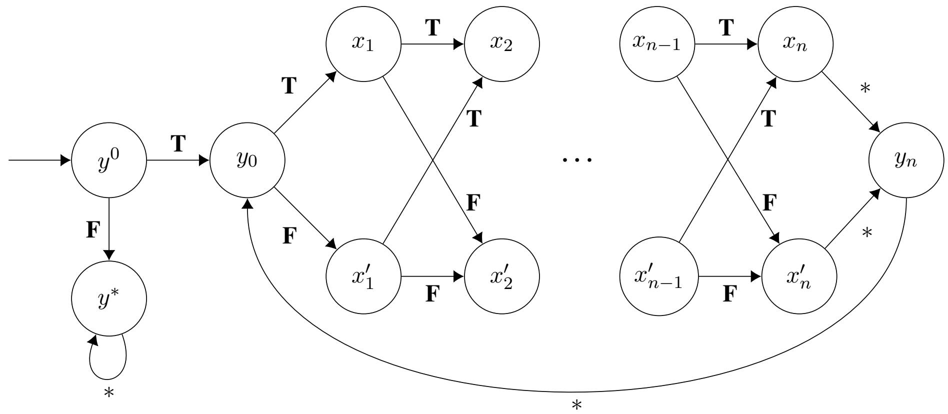

Figure 4: Concurrent game structure for the reduction from 3SAT.

- $w_i(x'_v) = 1$ if $\neg x_v$ is a literal in C_i and $w_i(x'_v) = 0$ otherwise, for all $i \in \mathrm{Ag}$ and $1 \leq v \leq n$

- $w_i(y_0) = w_i(y_n) = w_i(y^0) = w_i(y^*) = 0$, for all $i \in \mathrm{Ag}$

Then, we consider the game G over M and any strategy profile (in memoryless strategies) such that $\vec{\sigma}(s^0) = y^*$. For any of such strategy profiles the mean-payoff of every player is 0. However, if φ is satisfiable, then there is a path in M, from y_0 to y_n, such that in such a path, for every player, there is a state in which its payoff is not 0. Thus, the grand coalition Ag has an incentive to deviate since traversing that path infinitely often will give each player a mean-payoff strictly greater than 0. Observe two things. Firstly, that only if the grand coalition Ag agrees, the game can visit y_0 after y^0. Otherwise, the game will necessarily end up in y^* forever after. Secondly, because we are considering memoryless strategies, the path from y_0 to y_n followed at the beginning is the same path that will be followed thereafter, infinitely often. Then, we can conclude that there is a beneficial deviation (necessarily for Ag) if and only if φ is satisfiable, as otherwise at least one of the players in the game will not have an incentive to deviate (because its mean-payoff would continue to be 0). Then, formally, we can conclude that $(G, \sigma) \in$ BEN-DEV if and only if φ is satisfiable. □

From Theorem 9 it follows that checking if no coalition of players has a beneficial deviation with respect to a given strategy profile is a co-NP problem; that is, we have:

Theorem 10. *For memoryless strategies, MEMBERSHIP is co-NP-complete.*

Proof. Immediately follows from Lemma 3 and Theorem 9. □

We can also leverage BEN-DEV to solve E-CORE in the case of memoryless strategies:

Theorem 11. *For memoryless strategies, E-CORE is in* Σ_2^P.

Proof. Given a game G, we guess a strategy profile $\vec{\sigma}$ and check that $(G, \vec{\sigma})$ is not an instance of BEN-DEV. While the former can be done in polynomial time, the latter can be solved in co-NP using an oracle for BEN-DEV. Thus, we have a procedure that runs in $\text{NP}^{\text{co-NP}} = \text{NP}^{\text{NP}} = \Sigma_2^{\text{P}}$. □

Theorem 11 sharply contrasts with that for Nash equilibrium, where the same problem lies in NP [35]. More importantly, the result also shows that the (complexity) dependence on the type of coalitional deviation is only weak, in the sense that different types of beneficial deviations may be considered within the same complexity class, as long as such deviations can be checked with an NP or co-NP oracle.

We now wish to generalise the idea of a *fulfilled coalition*, which informally characterises coalitions that have the strategic power (a joint strategy) to ensure a minimum given payoff *no matter what the other players in the game do*. Generalising to our setting, from qualitative to quantitative payoffs, we introduce the notion of a *lower bound*. Let $C \subseteq \mathrm{Ag}$ be a coalition in a game G and let $\vec{z}_C \in \mathbb{Q}^C$. We say that $\vec{z}_C$ is a *lower bound* for C if there is a joint strategy $\vec{\sigma}_C$ for C such that for all strategies $\vec{\sigma}_{-C}$ for $\mathrm{Ag} \setminus C$, we have $\mathsf{pay}_i(\rho(\vec{\sigma}_C, \vec{\sigma}_{-C})) \geq z_i$, for every $i \in C$.

Based on this definition, we can prove the following lemma, which characterises the core in terms of lower bounds:

Lemma 4. *Let π be a run in G. There is $\vec{\sigma} \in core(G)$ such that $\pi = \rho(\vec{\sigma})$ if and only if for every coalition $C \subseteq \mathrm{Ag}$ and lower bound $\vec{z}_C \in \mathbb{Q}^C$ for C, there is some $i \in C$ such that $z_i \leq \mathsf{pay}_i(\pi)$.*

Proof. To show the left-to-right direction, suppose that there exists a member of the core $\vec{\sigma} \in core(G)$ with $\pi = \rho(\vec{\sigma})$ and suppose further that there is some coalition $C \subseteq \mathrm{Ag}$ and lower bound $\vec{z}_C \in \mathbb{Q}^C$ for C, such that for every $i \in C$ we have $z_i > \mathsf{pay}_i(\pi)$. Because $\vec{z}_C$ is a lower bound for C, and $z_i > \mathsf{pay}_i(\pi)$, for every $i \in C$, then there is a joint strategy $\vec{\sigma}_C$ for C such that for all strategies $\vec{\sigma}_{-C}$ for $\mathrm{Ag} \setminus C$, we have $\mathsf{pay}_i(\rho(\vec{\sigma}_C, \vec{\sigma}_{-C})) \geq z_i > \mathsf{pay}_i(\pi)$, for every $i \in C$. Then, it follows that $(G, \vec{\sigma}) \in$ BEN-DEV, which further implies that $\vec{\sigma}$ cannot be in the core of G — a contradiction to our initial hypothesis.

For the right-to-left direction, suppose that there is π in G such that for every coalition $C \subseteq \mathrm{Ag}$ and lower bound $\vec{z}_C \in \mathbb{Q}^C$ for C, there is $i \in C$ such that $z_i \leq \mathsf{pay}_i(\pi)$. We then simply let $\vec{\sigma}$ be any strategy profile such that $\pi = \rho(\vec{\sigma})$. Now, let $C = \{j, \ldots, k\} \subseteq \mathrm{Ag}$ be any coalition and $\vec{\sigma}'_C$ be any possible deviation of C from $\vec{\sigma}$. Either $\vec{z'}_C = (\mathsf{pay}_j(\rho(\vec{\sigma}_{-C}, \vec{\sigma}'_C)), \ldots, \mathsf{pay}_k(\rho(\vec{\sigma}_{-C}, \vec{\sigma}'_C)))$ is a lower bound for C or it is not.

If we have the former, by hypothesis, we know that there is $i \in C$ such that $\mathsf{pay}_i(\rho(\vec{\sigma}_{-C}, \vec{\sigma}'_C)) \leq \mathsf{pay}_i(\pi)$. Therefore, i will not have an incentive to deviate along with $C \setminus \{i\}$ from $\vec{\sigma}$, and as a consequence coalition C will not be able to beneficially deviate from $\vec{\sigma}$.

If, on the other hand, $\vec{z'}_C$ is not a lower bound for C, then, by the definition of lower bounds, we know that it is not the case that $\vec{\sigma}'_C$ is a joint strategy for C such that for all strategies $\vec{\sigma}'_{-C}$ for $\mathrm{Ag} \setminus C$, we have $\mathsf{pay}_i(\rho(\vec{\sigma}'_C, \vec{\sigma}'_{-C})) \geq \mathsf{pay}_i(\rho(\vec{\sigma}_{-C}, \vec{\sigma}'_C))$, for every $i \in C$. That is, there exists $i \in C$ and $\vec{\sigma}'_{-C}$ for $\mathrm{Ag} \setminus C$ such that $\mathsf{pay}_i(\rho(\vec{\sigma}'_C, \vec{\sigma}'_{-C})) < \mathsf{pay}_i(\rho(\vec{\sigma}_{-C}, \vec{\sigma}'_C))$. We will now choose $\vec{\sigma}'_{-C}$ so that, in addition, $\mathsf{pay}_i(\pi) \geq \mathsf{pay}_i(\rho(\vec{\sigma}'_C, \vec{\sigma}'_{-C}))$ for some i.

Let $\vec{z''}_C = (\mathsf{pay}_j(\rho(\vec{\sigma}^j_{-C}, \vec{\sigma}'_C)), \ldots, \mathsf{pay}_k(\rho(\vec{\sigma}^k_{-C}, \vec{\sigma}'_C)))$ where $\mathsf{pay}_i(\rho(\vec{\sigma}^i_{-C}, \vec{\sigma}'_C))$ is defined to be $\min_{\vec{\sigma}'_{-C} \in \Sigma_{-C}} \mathsf{pay}_i(\rho((\vec{\sigma}'_{-C}, \vec{\sigma}'_C)))$. That is, $\vec{\sigma}^i_{-C}$ is a strategy for $\mathrm{Ag} \setminus C$ which ensures the lowest mean-payoff for i assuming that C is playing the joint strategy $\vec{\sigma}'_C$. By construction $\vec{z''}_C$ is a lower bound for C — since each $z''_i = \mathsf{pay}_i(\rho(\vec{\sigma}^i_{-C}, \vec{\sigma}'_C))$ is the greatest mean-payoff value that i can ensure for itself when C is playing $\vec{\sigma}'_C$, no matter what coalition $\mathrm{Ag} \setminus C$ does — and therefore, by hypothesis we know that for some $i \in C$ we have $\mathsf{pay}_i(\rho(\vec{\sigma}^i_{-C}, \vec{\sigma}'_C)) \leq \mathsf{pay}_i(\pi)$. As a consequence, as before, i will not have an incentive to deviate along with $C \setminus \{i\}$ from $\vec{\sigma}$, and therefore coalition C will not be able to beneficially deviate from $\vec{\sigma}$. Because C and $\vec{\sigma}'_C$ where arbitrarily chosen, we conclude that $\vec{\sigma} \in core(G)$, proving the right-to-left direction and finishing the proof. $\square$

With this lemma in mind, we want to determine if a given vector, $\vec{z}_C$, is in fact a lower bound and importantly, how efficiently we can do this. That is, to understand the following decision problem:

LOWER-BOUND:

Given: Game G, coalition $C \subseteq \mathrm{Ag}$, and vector $\vec{z}_C \in \mathbb{Q}^{\mathrm{Ag}}$.

Question: Is $\vec{z}_C$ is a lower bound for C in G?

Using the MULTI-MEAN-PAYOFF-THRESHOLD decision problem introduced earlier, we can prove the following theorem:

Theorem 12. *Lower-Bound is co-NP-complete.*

Proof. We prove membership as well as hardness by reducing to and from Multi-Mean-Payoff-Threshold in the obvious way. First, we show that Lower-Bound lies in co-NP by reducing it to Multi-Mean-Payoff-Threshold. Suppose we have an instance, $(G, C, \vec{z}_C)$, and we want to determine if it is in Lower-Bound. We can do this by forming a two-player, multi-mean-payoff game, $G' = (V_1, V_2, v^0, E, w', z^k)$. Here we have $V_1 = \mathrm{St}$, $V_2 = \mathrm{St} \times \mathrm{Ac}_C$ and $v^0 = s^0$. Additionally, the set of edges of G', E, is defined as,

$$E = \{(s, (s, \mathrm{ac}_C)) \mid s \in \mathrm{St}, \mathrm{ac}_C \in \mathrm{Ac}_C\}$$
$$\cup \{((s, \mathrm{ac}_C), \mathrm{tr}(s, (\mathrm{ac}_C, \mathrm{ac}_{\mathrm{Ag}\setminus C}))) \mid \mathrm{ac}_{\mathrm{Ag}\setminus C} \in \mathrm{Ac}_{\mathrm{Ag}\setminus C}\},$$

and the weight function, $w' : E \to \mathbb{Z}^{|C|}$, is defined by the following two patterns:

$$w'_i(s, (s, \mathrm{ac}_C)) = w_i(s);$$
$$w'_i((s, \mathrm{ac}_C), \mathrm{tr}(s, (\mathrm{ac}_C, \mathrm{ac}_{\mathrm{Ag}\setminus C}))) = w_i(s).$$

Finally, we set $z^{|C|}$ to be $\vec{z}_c$.

Informally, the two players of the game are C and $\mathrm{Ag} \setminus C$, the vector weight function is given by aggregating the weight functions of C and the threshold is $\vec{z}_C$. Now, if in this game, player 1 has a winning strategy, then there exists some strategy $\vec{\sigma}_C$ such that for all strategies of player 2, $\vec{\sigma}_{\mathrm{Ag}\setminus C}$, we have that $\rho(\vec{\sigma}_C, \vec{\sigma}_{\mathrm{Ag}\setminus C})$ is a winning run for player 1. But this means that $\mathsf{pay}_i(\rho(\vec{\sigma}_C, \vec{\sigma}_{\mathrm{Ag}\setminus C})) \geq z_i$ for all $i \in C$. But it is easy to verify that this implies that $\vec{z}_C$ is a lower bound for C in G. Conversely, if player 1 has no winning strategy, then for all strategies, $\vec{\sigma}_C$, there exists some strategy $\vec{\sigma}_{\mathrm{Ag}\setminus C}$ such that $\rho(\vec{\sigma}_C, \vec{\sigma}_{\mathrm{Ag}\setminus C})$ is not a winning run. This is turn implies that for some $j \in C$, we have that $\mathsf{pay}_j(\rho(\vec{\sigma}_C, \vec{\sigma}_{\mathrm{Ag}\setminus C})) < z_j$, which means that $\vec{z}_C$ is not a lower bound for C in G. Also note that this construction can be performed in polynomial time, giving us the co-NP upper bound.

For the lower bound, we go the other way and reduce from MULTI-MEAN-PAYOFF-THRESHOLD. Suppose we would like to determine if an instance G is in MULTI-MEAN-PAYOFF-THRESHOLD. Then we form a concurrent mean-payoff game, G', with $k+1$ players, where the states of G' coincide exactly with the states of G. In this game, only the 1^{st} and $(k+1)^{\text{th}}$ player have any influence on the strategic nature of the game. If the game is in a state in V_1, player one can decide which state to move into next. Otherwise, if the game is in a state within V_2, then the $(k+1)^{\text{th}}$ player makes a move. Note we only allow moves that agree with moves allowed within G.

Now, in G', the first k players have weight functions corresponding correspond to the k weight functions of player 1 in G. The last player can have any arbitrary weight function. With this machinery in place, we ask if z^k is a lower bound for $\{1, \ldots, k\}$. In a similar manner of reasoning to the above, it is easy to verify that G is an instance of MULTI-MEAN-PAYOFF-THRESHOLD if and only if z^k is a lower bound for $\{1, \ldots, k\}$ in the constructed concurrent mean-payoff game. Moreover, this reduction can be done in polynomial time and we can conclude that LOWER-BOUND is co-NP-complete. □

A notable omission from this section is that we have not presented any bounds for the complexity of E-CORE in the general case. One possible reason for the upper bounds remaining elusive to us is due to the fact that whilst in a multi-mean-payoff game, player 2 can act optimally with memoryless strategies, player 1 may require infinite memory [37, 38]. Given the close connection between the core in our concurrent, multi-agent setting and winning strategies in multi-mean-payoff games, this raises computational concerns for the E-CORE problem. Additionally, in [39], the authors study the Pareto frontier of multi-mean-payoff games, and provide a way of constructing a representation of it, but this procedure has an exponential time dependency. The same paper also establishes Σ_2^P-completeness for the *polyhedron value problem*. Both of these problems appear to be intimately related to the core, and we hope we might be able to use these results to gain more insight into the E-CORE in the future.

With this having been said, we conclude this section by establishing a link between traditional non-transferable utility (NTU) games and our mean-payoff games — as NTU games are very well studied, and there is a wealth of results relating to core non-emptiness in this setting [40–42], we hope that some of these results could be utilised in order to understand the core of mean-payoff games.

Formally, an n-person game with non-transferable utility is a function, $V : \mathcal{P}(\mathrm{Ag}) \to \mathbb{R}^{|\mathrm{Ag}|}$, such that,

1. For all $C \subseteq \mathrm{Ag}$, $V(C)$ is a non-empty, proper, closed subset of $\mathbb{R}^{|\mathrm{Ag}|}$;
2. For all $C \subseteq \mathrm{Ag}$, if we have $x \in V(C)$ and $y \in \mathbb{R}^{|\mathrm{Ag}|}$ such that $y_i \leq x_i$ for all $i \in C$, then we have $y \in V(C)$;
3. We have that $V(\mathrm{Ag}) \setminus \bigcup_{i \in \mathrm{Ag}} \operatorname{int} V(\{i\})$ is non-empty and bounded.

We begin by giving a translation from mean-payoff games to NTU games. Let G be a mean-payoff game; then we define an NTU game, $G_{\mathrm{NTU}} = V : \mathcal{P}(\mathrm{Ag}) \to \mathbb{R}^{|\mathrm{Ag}|}$ as follows. If $C \subseteq \mathrm{Ag}$, then,

$$V(C) = \left\{ \vec{z} \mid \vec{z} \in \mathbb{R}^{|\mathrm{Ag}|}, \exists \vec{\sigma}_C \forall \vec{\sigma}_{\mathrm{Ag} \setminus C} \forall i \in C, \mathsf{pay}_i \left(\rho \left(\vec{\sigma}_C, \vec{\sigma}_{\mathrm{Ag} \setminus C} \right) \right) \geq z_i \right\}.$$

In words, $V(C)$ consists of the set of lower bounds that C can force. Note that for an outcome $x \in V(C)$, the components $x_i$for $i \in \mathrm{Ag} \setminus C$ do not matter — they can be arbitrary real numbers.

Lemma 5. *Let G be a game, and let G_{NTU} be the NTU game associated with G. Then G_{NTU} is well-defined.*

Proof. We need to show that the three conditions in the definition of an NTU game hold for G_{NTU}.

For condition (1), we see that $V(C)$ is always non-empty by noting a coalition can always force an outcome where they achieve at least their worst possible payoff each (the vector made up of each player's lowest weight in the game). The fact that $V(C)$ is closed follows from Theorem 4 of [39]. We also see that $V(C)$ is a proper subset of $\mathbb{R}^{|\text{Ag}|}$, as the members of C can do no better than achieve their maximum weights.

For condition (2), suppose we have $x \in V(C)$, and $y \in \mathbb{R}^{|\text{Ag}|}$ with $y_i \leq x_i$ for all $i \in C$. If $x \in V(C)$, then there exists some $\vec{\sigma}_C$, such that for all $\vec{\sigma}_{\text{Ag}\setminus C}$, we have $\mathsf{pay}_i(\rho(\vec{\sigma}_C, \vec{\sigma}_{\text{Ag}\setminus C})) \geq x_i$ for all $i \in C$. But this in turn implies that $\mathsf{pay}_i(\rho(\vec{\sigma}_C, \vec{\sigma}_{\text{Ag}\setminus C})) \geq y_i$ for all $i \in C$. Thus, by definition, we have $y \in V(C)$.

For condition (3), let p_j be the **punishment value** of the player j in the game G. Informally, the punishment value of a player j can be thought of as the worst payoff that the other players can inflict on that player. Alternatively, we can view the punishment value for player j as the best payoff they can guarantee themselves, no matter what the remaining players do — in this way, we can see that the punishment value is a maximal lower bound for a player.

Consider the vector $p \in \mathbb{R}^{|\text{Ag}|}$, where the j^{th} component of this vector is the punishment value for player j. Naturally, this vector lies in $V(\text{Ag})$. Additionally, we claim that is does not lie in $\text{int}\, V(\{i\})$ for any $i \in \text{Ag}$. For a contradiction, suppose there existed some $j \in \text{Ag}$ with $p \in \text{int}\, V(\{j\})$. So there exists some $\epsilon > 0$, such that for all $0 \leq r < \epsilon$, there exists some strategy, σ_j^r, such that for all counterstrategies, $\vec{\sigma}_{-j}$, we have $\mathsf{pay}_j(\rho(\sigma_j^r, \vec{\sigma}_{-j})) \geq p_j + r$. But this implies player j can achieve a better payoff than their punishment value — a contradiction. Thus, we see that the set $V(\text{Ag}) \setminus \bigcup_{i \in \text{Ag}} \text{int}\ V(\{i\})$ is non-empty.

Finally, to see that $V(\text{Ag}) \setminus \bigcup_{i \in \text{Ag}} \text{int}\, V(\{i\})$ is bounded, we claim that it is contained in a closed ball of radius M, where M is defined to be

$$M = \max_{\substack{i \in \mathrm{Ag} \\ s \in \mathrm{St}}} |w_i(s)| .$$

We show that if $x \in V(\mathrm{Ag})$, then we either have $x \in \bigcup_{i \in \mathrm{Ag}} \operatorname{int} V(\{i\})$ or $x \in \overline{B(0, M)}$, i.e., the closed ball of radius M, centred at the origin.

If $x \in V(\mathrm{Ag})$, then by definition, we must have $x_i \leq M$ for all $i \in \mathrm{Ag}$. Now, there are two possibilities: if we have $x_i \geq -M$ for all $i \in \mathrm{Ag}$, then we have $x \in \overline{B(0, M)}$. So instead suppose there exists some $i \in \mathrm{Ag}$ such that $x_i < -M$. In this case, letting ϵ be any positive number such that $x_i + \epsilon \leq -M$, any strategy σ_i has the property that for all counter-strategies $\vec{\sigma}_{-i}$, we have $\mathsf{pay}_i(\rho(\vec{\sigma}_{-i}, \sigma_i)) \geq x_i + \epsilon$. Thus, we have $x \in \operatorname{int} V(\{i\})$. This implies that,

$$V(\mathrm{Ag}) \subseteq \bigcup_{i \in \mathrm{Ag}} \operatorname{int} V(\{i\}) \cup \overline{B(0, M)},$$

which in turn implies,

$$V(\mathrm{Ag}) \setminus \bigcup_{i \in \mathrm{Ag}} \operatorname{int} V(\{i\}) \subseteq \overline{B(0, M)},$$

yielding the result. □

Given that we can translate mean-payoff games into well-defined NTU games, it is natural to ask whether we can use traditional cooperative game theory in order to understand the core in our setting. Thus, we introduce the (classic) definition of the core for NTU games. In an NTU game, we say that an element $x \in \mathbb{R}^{|\mathrm{Ag}|}$ is in the core if $x \in V(\mathrm{Ag})$, and there exists no $C \subseteq \mathrm{Ag}$ and no $y \in V(C)$ such that $x_i < y_i$ for all $i \in C$. In the following result, we show that the core of a mean-payoff game, and the core of its corresponding NTU game are intimately related:

Lemma 6. *Let G be a mean-payoff game. Let G_{NTU} be the NTU game associated with G. Then the core of G is non-empty if and only if the core of G_{NTU} is non-empty.*

In fact, we have a stronger result — we can define a bijection between the two sets. Should I state it in terms of this result instead?

Proof. First suppose that G has a non-empty core. Thus, there exists some strategy profile $\vec{\sigma}$ such that for all coalitions C and for all strategy vectors $\vec{\sigma}'_C$, there exists some $\vec{\sigma}'_{\mathrm{Ag}\setminus C}$ such that $\mathsf{pay}_i(\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\mathrm{Ag}\setminus C})) \leq \mathsf{pay}_i(\rho(\vec{\sigma}))$ for some $i \in C$. Let $x \in \mathbb{R}^{\mathrm{Ag}}$ be such that $x_i = \mathsf{pay}_i(\vec{\sigma})$ for all $i \in \mathrm{Ag}$. Then by definition, we have $x \in V(\mathrm{Ag})$. We claim that x is in the core of G_{NTU}. Suppose there is some $C \subseteq \mathrm{Ag}$ and a $y \in V(C)$ such that $x_i < y_i$ for all $i \in C$. Thus, there exists some $\vec{\sigma}'_C$ such that for all $\vec{\sigma}'_{\mathrm{Ag}\setminus C}$, such that $\mathsf{pay}_i(\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\mathrm{Ag}\setminus C})) \geq y_i > x_i = \mathsf{pay}_i(\vec{\sigma})$ for all $i \in C$. But this implies that $\vec{\sigma}$ is not in the core of G, which is a contradiction. Thus, x is in the core of G_{NTU}.

Conversely, suppose that G_{NTU} has an empty core. Thus, there exists some $x \in \mathbb{R}^{|\mathrm{Ag}|}$ such that $x \in V(\mathrm{Ag})$, such that there exists no $C \subset \mathrm{Ag}$ and no $y \in V(C)$ with $x_i < y_i$ for all $i \in C$. Since $x \in V(\mathrm{Ag})$, there exists some strategy $\vec{\sigma}$ such that $\mathsf{pay}_i(\rho(\vec{\sigma})) \geq x_i$ for all $i \in \mathrm{Ag}$. We claim that $\vec{\sigma}$ is in the core of G. If it were not, then there would exist some coalition C, and some strategy vector $\vec{\sigma}'_C$ such that for all strategy vectors $\vec{\sigma}'_{\mathrm{Ag}\setminus C}$, we have $\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\mathrm{Ag}\setminus C}) \succ_i \rho(\vec{\sigma})$ for all $i \in C$. We then define $y \in \mathbb{R}^{\mathrm{Ag}}$ by setting $y_i = \min_{\vec{\sigma}'_{\mathrm{Ag}\setminus C}} \mathsf{pay}_i(\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\mathrm{Ag}\setminus C}))$ for $i \in C$ and setting $y_i = 0$ for $i \in \mathrm{Ag} \setminus C$. Then we have that $y \in V(C)$ by definition. Since for all $\vec{\sigma}'_{\mathrm{Ag}\setminus C}$, we have that $\rho(\vec{\sigma}'_C, \vec{\sigma}'_{\mathrm{Ag}\setminus C})$ is strictly preferred to $\rho(\vec{\sigma})$ by all players in C, we must have that $y_i > x_i$ for all $i \in C$. But this contradicts the fact that x is in the core of G_{NTU}. Thus, we must have that $\vec{\sigma}$ is in the core of G. □

As stated previously, we have been unable to determine the complexity of E-CORE in the setting of mean-payoff games. However, given the above result, we suggest a route which may bear fruits in the future. In [40–44], the authors reason about the core of cooperative games (in both the transferable utility and non-transferable utility settings) by appealing to the notion of a **balanced** set. In [41], the authors generalise this by introducing the notion of π-balancedness. Let $\pi = \{\pi_C \in \mathbb{R}^{|\mathrm{Ag}|} \mid C \subseteq \mathrm{Ag}\}$ be a collection of vectors such that,

1. For all $C \subseteq \mathrm{Ag}$, we have $\pi_C \neq \mathbf{0}$;
2. For all $C \subseteq \mathrm{Ag}$, and for all $i \notin C$, we have $\pi_{C,i} = 0$;
3. For all $C \subseteq \mathrm{Ag}$, and for all $i \in C$, we have $\pi_{C,i} \geq 0$,

and let $\mathcal{C} \subseteq \mathcal{P}(\mathrm{Ag})$ be a collection of coalitions. We say that $\mathcal{C}$ is π-balanced if there exist balancing weights, $\lambda_C > 0$, for each $C \subseteq \mathrm{Ag}$ such that,

$$\sum_{C \in \mathcal{C}} \lambda_C \pi_C = \pi_{\mathrm{Ag}}.$$

We then say that an NTU game, V, is π-balanced if whenever $\mathcal{C}$ is a π-balanced collection, we have,

$$\bigcap_{C \in \mathcal{C}} V(C) \subseteq V(\mathrm{Ag}).$$

In [41], the authors show that if there exists some π such that V is π-balanced, then V has a non-empty core. The condition of π-balancedness translates readily over to the setting of mean-payoff games, and so we see that if such a game is π-balanced, then it has a non-empty core. This suggest a a (sound, but not complete) algorithm for detecting if a mean-payoff game has a non-empty core; somehow guess a polynomial-sized π, use a linear program to calculate the corresponding balancing weights, and then use an co-NP oracle to verify there exists no π-balanced collection such that $\bigcap_{C \in \mathcal{C}} V(C) \subseteq V(\mathrm{Ag})$. Obviously, this is not a rigorous argument, but is suggestive of what a possible solution may look like.

Additionally, whilst π-balancedness is a sufficient condition for core non-emptiness, it is not necessary. However, in [41], the authors strengthen the condition of π-balancedness in the setting of convex-valued NTU games, to obtain a necessary and sufficient result. Given in that mean-payoff games, the outcomes that a coalition can achieve can be expressed as a union of convex sets, this approach seems promising. However, we have been unable to yield any results via this route.

7. Discussion and Related Work

In this section, we present some concluding remarks and related work, discuss some issues related to the core, and indicate how to produce an implementation using model checking techniques.

Coalition formation in cooperative games. Coalition formation with externalities has been studied in the cooperative game-theory literature [19, 26, 27]. They considered several concepts of the core. For instance, α-core takes the pessimistic approach that requires that all members of a deviating coalition, S, will benefit from the deviation regardless of the behaviour of the other coalitions that may be formed. Our first definition of the core follows this approach. Contrarily, β-core takes an optimistic approach and requires that the members of a deviating coalition S will benefit from at least one possibility of coalition formation of the rest of the players. In addition, γ-core [45, 46] assumes that the coalition structure that will be created after a deviation will include the deviating coalition S and the rest of the coalition structure will consist of all singletons. The "worth" of S is now defined as equal to its payoff in the Nash equilibrium between S and the other players acting individually, in which the members of S play their joint best response strategy against the individually best response strategies of the remaining players. It is well-known that α- and β-characteristic functions lead to large cores [47], which is consistent with our observation that, with respect to our first definition, the core is never empty. Coalition formation is important in multi-agent system [48]. However, even though Coalition formation with externalities is very common in multi-agent systems, not much work has studied the concept of stability in multi-agent coalition formation with externalities. Instead, in artificial intelligence and multi-agent systems, most research has focused on the structure formation itself [49].

Rational verification of concurrent games. The formal verification of temporal logic properties of multi-agent systems, while assuming rational behaviour of the agents in such a system, has been studied for almost a decade now; see, for instance, [3–5, 32, 50]. However, to the best of our knowledge, all these studies have considered a non-cooperative setting, even if coalitional power is allowed, for instance, as in a strong Nash equilibrium. Nonetheless, also in such non-cooperative settings, the complexity of checking whether a temporal logic property is satisfied in a stable outcome of the game is a 2EXPTIME-complete problem, even for two-player zero-sum games where only trivial coalitions can be formed. On the positive side, cooperative games seem to have better model theoretic properties in the rational verification framework: with respect to our first definition of the core (which corresponds to the concept of α-core in the literature of cooperative games), a witness in the core is always guaranteed (since the core is never empty), preserved across bisimilar systems, and easily checked in practice in an efficient way using ATL* model checking techniques, which are supported by a number of verification tools, *e.g.*, MCMAS [9], an automated formal verification tool that supports ATL* model checking and specifications of concurrent game structures in ISPL.

Acknowledgment. This research was partly supported by the Israel Ministry of Science and Technology and the Japan Science and Technology Agency (JST).

References

[1] J. Gutierrez, S. Kraus, M. J. Wooldridge, Cooperative concurrent games, in: AAMAS, IFAAMAS, 2019, pp. 1198–1206.

[2] R. Alur, T. A. Henzinger, O. Kupferman, Alternating-time temporal logic, Journal of the ACM 49 (5) (2002) 672–713.

[3] J. Gutierrez, P. Harrenstein, M. Wooldridge, Iterated boolean games, Information and Computation 242 (2015) 53–79.

[4] J. Gutierrez, P. Harrenstein, M. Wooldridge, From model checking to equilibrium checking: Reactive modules for rational verification, Artificial Intelligence 248 (2017) 123–157.

[5] M. Wooldridge, J. Gutierrez, P. Harrenstein, E. Marchioni, G. Perelli, A. Toumi, Rational verification: From model checking to equilibrium checking, in: Proc. of AAAI, AAAI Press, 2016, pp. 4184–4191.

[6] E. A. Emerson, Temporal and modal logic, in: Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B), MIT Press, 1990, pp. 995–1072.

[7] E. M. Clarke, O. Grumberg, D. A. Peled, Model Checking, MIT Press, 2002.

[8] J. Gutierrez, M. Najib, G. Perelli, M. Wooldridge, EVE: A tool for temporal equilibrium analysis, in: ATVA, Vol. 11138 of Lecture Notes in Computer Science, Springer, 2018, pp. 551–557.

[9] A. Lomuscio, H. Qu, F. Raimondi, MCMAS: an open-source model checker for the verification of multi-agent systems, STTT 19 (1) (2017) 9–30.

[10] R. Brenguier, PRALINE: A tool for computing nash equilibria in concurrent games, in: CAV, Vol. 8044 of Lecture Notes in Computer Science, Springer, 2013, pp. 890–895.

[11] M. Kwiatkowska, G. Norman, D. Parker, G. Santos, Prism-games 3.0: Stochastic game verification with concurrency, equilibria and time, in: S. K. Lahiri, C. Wang (Eds.), Computer Aided Verification - 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part II, Vol. 12225 of Lecture Notes in Computer Science, Springer, 2020, pp. 475–487. doi:10.1007/978-3-030-53291-8_25.
URL https://doi.org/10.1007/978-3-030-53291-8_25

[12] M. J. Osborne, A. Rubinstein, A Course in Game Theory, MIT Press, 1994.

[13] M. Maschler, E. Solan, S. Zamir, Game Theory, Cambridge University Press, 2013.

[14] G. Chalkiadakis, E. Elkind, M. Wooldridge, Computational Aspects of Cooperative Game Theory, Morgan-Claypool, 2011.

[15] A. Pnueli, The temporal logic of programs, in: Proc. of FOCS, IEEE Computer Society, 1977, pp. 46–57.

[16] C. Baier, J.-P. Katoen, Principles of model checking, MIT Press.

[17] R. M. Keller, Formal verification of parallel programs 19 (7) 371–384, place: New York, NY, USA Publisher: Association for Computing Machinery. doi:10.1145/360248.360251.
URL https://doi.org/10.1145/360248.360251

[18] S. A. Kripke, Semantical analysis of modal logic i normal modal propositional calculi 9 (5) 67–96, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/malq.19630090502. doi:10.1002/malq.19630090502.

URL https://onlinelibrary.wiley.com/doi/abs/10.1002/malq.19630090502

[19] M. Uyanık, On the nonemptiness of the α-core of discontinuous games: Transferable and nontransferable utilities, Journal of Economic Theory 158 (2015) 213–231.

[20] M. Hennessy, R. Milner, Algebraic laws for nondeterminism and concurrency, Journal of the ACM 32 (1) (1985) 137–161.

[21] J. Gutierrez, P. Harrenstein, G. Perelli, M. Wooldridge, Nash equilibrium and bisimulation invariance, in: CONCUR, Vol. 85 of LIPIcs, Schloss Dagstuhl, 2017, pp. 17:1–17:16.

[22] E. A. Emerson, J. Y. Halpern, "sometimes" and "not never" revisited: on branching versus linear time temporal logic, Journal of the ACM 33 (1) (1986) 151–178.

[23] A. P. Sistla, E. M. Clarke, The complexity of propositional linear temporal logics, Journal of the ACM 32 (3) (1985) 733–749.

[24] R. Alur, S. La Torre, Deterministic generators and games for LTL fragments, ACM Transactions on Computational Logic 5 (1) (2004) 1–25.

[25] O. Kupferman, M. Y. Vardi, P. Wolper, An automata-theoretic approach to branching-time model checking, Journal of the ACM 47 (2) (2000) 312–360.

[26] S.-S. Yi, Stable coalition structures with externalities, Games and economic behavior 20 (2) (1997) 201–237.

[27] M. Finus, B. Rundshagen, A non-cooperative foundation of core-stability in

positive externality ntu-coalition games, Nota Di Lavoro 31.2003. Economics Energy Environment (2003).

[28] F. Mogavero, A. Murano, G. Perelli, M. Y. Vardi, Reasoning about strategies: On the model-checking problem, ACM Transactions on Computational Logic 15 (4) (2014) 34:1–34:47.

[29] F. Mogavero, A. Murano, G. Perelli, M. Y. Vardi, Reasoning about strategies: on the satisfiability problem, Logical Methods in Computer Science 13 (1) (2017).

[30] M. Mavronicolas, B. Monien, K. W. Wagner, Weighted boolean formula games, in: X. Deng, F. C. Graham (Eds.), Internet and Network Economics, Third International Workshop, WINE 2007, San Diego, CA, USA, December 12-14, 2007, Proceedings, Vol. 4858 of Lecture Notes in Computer Science, Springer, pp. 469–481. doi:10.1007/978-3-540-77105-0_51.
URL https://doi.org/10.1007/978-3-540-77105-0_51

[31] S. Almagor, O. Kupferman, G. Perelli, Synthesis of controllable nash equilibria in quantitative objective games, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI'18, AAAI Press, pp. 35–41, event-place: Stockholm, Sweden.

[32] O. Kupferman, G. Perelli, M. Y. Vardi, Synthesis with rational environments, Annals of Mathematics and Artificial Intelligence 78 (1) (2016) 3–20.

[33] A. Ehrenfeucht, J. Mycielski, Positional strategies for mean-payoff games, International Journal of Game Theory 8 (1979) 109–113.

[34] U. Zwick, M. Paterson, The complexity of mean payoff games on graphs 158 (1) 343–359.

[35] M. Ummels, D. Wojtczak, The complexity of nash equilibria in limit-average games abs/1109.6220.
URL http://arxiv.org/abs/1109.6220

[36] R. M. Karp, A characterization of the minimum cycle mean in a digraph 23 (3) 309–311.

[37] Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. M. Rabinovich, J.-F. Raskin, The complexity of multi-mean-payoff and multi-energy games 241 177–196. doi:10.1016/j.ic.2015.03.001.

[38] E. Kopczynski, Half-positional determinacy of infinite games, in: M. Bugliesi, B. Preneel, V. Sassone, I. Wegener (Eds.), Automata, Languages and Programming, 33\textsuperscriptrd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II, Vol. 4052 of Lecture Notes in Computer Science, Springer, pp. 336–347. doi:10.1007/11787006_29.
URL https://doi.org/10.1007/11787006_29

[39] R. Brenguier, J.-F. Raskin, Pareto curves of multidimensional mean-payoff games, in: D. Kroening, C. S. Pasareanu (Eds.), Computer Aided Verification - 27\textsuperscriptth International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part II, Vol. 9207 of Lecture Notes in Computer Science, Springer, pp. 251–267. doi:10.1007/978-3-319-21668-3_15.
URL https://doi.org/10.1007/978-3-319-21668-3_15

[40] H. E. Scarf, The core of an n person game 50–69.

[41] L. J. Billera, Some theorems on the core of an n-game without side-payments 18 (3) 567–579.

[42] L. S. Shapley, On balanced games without side payments, in: Mathematical Programming, Elsevier, pp. 261–290. doi:10.1016/b978-0-12-358350-5.50012-9.

[43] O. N. Bondareva, Some applications of linear programming methods to the theory of cooperative games, Problemy kibernetiki 10 (119) (1963) 139.

[44] L. S. Shapley, On balanced sets and cores 14 (4) 453–460. doi:10.1002/nav.3800140404.

[45] P. Chander, et al., Cores of games with positive externalities, CORE DP 2010/4 (2010).

[46] P. Chander, The gamma-core and coalition formation, International Journal of Game Theory 35 (4) (2007) 539–556.

[47] D. Ray, R. Vohra, Equilibrium binding agreements, Journal of Economic theory 73 (1) (1997) 30–78.

[48] O. Shehory, S. Kraus, Methods for task allocation via agent coalition formation, Artificial intelligence 101 (1) (1998) 165–200.

[49] T. Rahwan, T. Michalak, M. Wooldridge, N. R. Jennings, Anytime coalition structure generation in multi-agent systems with positive or negative externalities, Artificial Intelligence 186 (2012) 95–122.

[50] D. Fisman, O. Kupferman, Y. Lustig, Rational synthesis, in: TACAS, Vol. 6015 of Lecture Notes in Computer Science, Springer, 2010, pp. 190–204.