

How to Be Both Rich and Happy: Combining Quantitative and Qualitative Strategic Reasoning about Multi-Player Games (Extended Abstract)

Nils Bulling

Clausthal University of Technology, Germany
bulling@in.tu-clausthal.de

Valentin Goranko

Technical University of Denmark, Denmark
vfgo@imm.dtu.dk

We propose a logical framework combining a game-theoretic study of abilities of agents to achieve *quantitative objectives* in multi-player games by optimizing payoffs or preferences on outcomes with a logical analysis of the abilities of players for achieving *qualitative objectives* of players, i.e., reaching or maintaining game states with desired properties. We enrich concurrent game models with payoffs for the normal form games associated with the states of the model and propose a quantitative extension of the logic ATL* enabling the combination of quantitative and qualitative reasoning.

1 Introduction

There are two rich traditions in studying strategic abilities of agents in multi-player games:

Game theory has been studying rational behavior of players, relevant for their achievement of *quantitative objectives*: optimizing payoffs (e.g., maximizing rewards or minimizing cost) or, more generally, preferences on outcomes. Usually, the types of games studied in game theory are one-shot normal form games, their (finitely or infinitely) repeated versions, and extensive form games.

Logic has been mostly dealing with strategic abilities of players for achieving *qualitative objectives*: reaching or maintaining outcome states with desired properties, e.g., winning states, or safe states, etc.

Among the most studied models in the logic tradition are *concurrent game models* [5, 21]. On the one hand they are richer than normal form games, as they incorporate a whole family of such games, each associated with a state of a transition system; but on the other hand, they are somewhat poorer because the outcomes of each of these normal form games, associated with a given state, are simply the successor states with their associated games, etc. whereas no payoffs, or even preferences on outcomes, are assigned. Thus, plays in concurrent game models involve a sequence of possibly different one-shot normal form games played in succession, and all that is taken into account in the purely logical framework are the properties – expressed by formulae of a logical language – of the states occurring in the play. Concurrent game models can also be viewed as generalization of (possibly infinite) extensive form games where cycles and simultaneous moves of different players are allowed, but no payoffs are assigned.

Put as a slogan, the game theory tradition is concerned with *how a player can become maximally rich, or how to pay as little cost as possible*, while the logic tradition – with *how a player can achieve a state of ‘happiness’, e.g. winning, or to avoid reaching a state of ‘unhappiness’ (losing) in the game*.

The most essential technical difference between qualitative and quantitative players’ objectives is that the former typically refer to (a temporal pattern over) Boolean properties of game states on a given play and can be monitored locally whereas the latter are determined by the entire history of the play (accumulated payoffs) or even the whole play (its value, being a limit of average payoffs, or of discounted

accumulated payoffs). It is therefore generally computationally more demanding and costly to design strategies satisfying quantitative objectives or to verify their satisfaction under a given strategy of a player or coalition.

These two traditions have followed rather separate developments, with generally quite different agendas, methods and results, including, inter alia:

- on the purely qualitative side, *logics of games and multiagent systems*, such as the Coalition logic CL [21], the Alternating time temporal logic ATL [5], and variations of it, see e.g. [15], [18], etc., formalizing and studying qualitative reasoning in concurrent game models;
- some *single-agent and multi-agent bounded resource logics* [9, 3, 19] extending or modifying concurrent game models with some quantitative aspects by considering cost of agents' actions and reasoning about what players with bounded resources can achieve.
- extensions of qualitative reasoning (e.g., reachability and Büchi objectives) in multi-player concurrent games with 'semi-quantitative' aspects by considering a preference preorder on the set of qualitative objectives, see e.g., [6], [7], thereby adding payoff-maximizing objectives and thus creating a setting where traditional game-theoretic issues such as game value problems and Nash equilibria become relevant.
- deterministic or stochastic *infinite games on graphs*, with qualitative objectives: typically, reachability, and more generally – specified as ω -regular languages over the set of plays, see e.g. [4], [10], [12].
- on the purely quantitative side, first to mention *repeated games*, extensively studied in game theory (see e.g., [20]), which can be naturally treated as simple, one-state concurrent game models with accumulating payoffs paid to each player after every round and no qualitative objectives;
- from a more computational perspective, stochastic games with quantitative objectives on discounted, mean or total payoffs, in particular energy objectives, see e.g. [11].
- the conceptually different but technically quite relevant study of *counter automata*, *Petri nets*, *vector addition systems*, etc. – essentially a study of the purely quantitative single-agent case of concurrent game models (see e.g. [14]), where only accumulated payoffs but no qualitative objectives are taken into account and a typical problem is to decide reachability of payoff configurations satisfying formally specified arithmetic constraints from a given initial payoff configuration.

A number of other relevant references discuss the interaction between qualitative and quantitative reasoning in multi-player games, e.g. [22], [16], which we cannot discuss here due to space limitations.

This project purports to combine the two agendas in a common logical framework, by enriching concurrent game models with payoffs for the one-shot normal form games associated with the states, and thus enabling the combination of quantitative game-theoretic reasoning with the qualitative logical reasoning. Again, put as a slogan, our framework allows reasoning about whether/how a player can reach or maintain a state of 'happiness' while becoming, or remaining, as rich as (rationally) possible, or paying the least possible price on the way. The purpose of this extended abstract is to introduce and discuss a general framework of models and logics for combined quantitative and qualitative reasoning that would naturally cover each of the topics listed above, and to initiate a long term study on it.

2 Preliminaries

A **concurrent game model** [5] (CGM) $\mathcal{G} = (\text{Ag}, \text{St}, \{\text{Act}_a\}_{a \in \text{Ag}}, \{\text{act}_a\}_{a \in \text{Ag}}, \text{out}, \text{Prop}, \text{L})$ comprises:

- a non-empty, fixed set of players $\text{Ag} = \{1, \dots, k\}$ and a set of actions $\text{Act}_a \neq \emptyset$ for each $a \in \text{Ag}$.
For any $A \subseteq \text{Ag}$ we will denote $\text{Act}_A := \prod_{a \in A} \text{Act}_a$ and will use $\vec{\alpha}_A$ to denote a tuple from Act_A .
In particular, Act_{Ag} is the set of all possible action profiles in \mathcal{S} .
- a non-empty set of game states St .
- for each $a \in \text{Ag}$ a map $\text{act}_a : \text{St} \rightarrow \mathcal{P}(\text{Act}_a)$ setting for each state s the actions available to a at s .
- a transition function $\text{out} : \text{St} \times \text{Act}_{\text{Ag}} \rightarrow \text{St}$ that assigns the (deterministic) *successor (outcome) state* $\text{out}(q, \vec{\alpha}_{\text{Ag}})$ to every state q and action profile $\vec{\alpha}_{\text{Ag}} = \langle \alpha_1, \dots, \alpha_k \rangle$ such that $\alpha_a \in \text{act}_a(q)$ for every $a \in \text{Ag}$ (i.e., every α_a that can be executed by player a in state q).
- a set of atomic propositions Prop and a labelling function $L : \text{St} \rightarrow \mathcal{P}(\text{Prop})$.

Thus, all players in a CGM execute their actions synchronously and the combination of these actions, together with the current state, determines the transition to a (unique) successor state in the CGM.

The **logic of strategic abilities** ATL^* (*Alternating-Time Temporal Logic*), introduced and studied in [5], is a logical system, suitable for specifying and verifying qualitative objectives of players and coalitions in concurrent game models. The main syntactic construct of ATL^* is a formula of type $\langle\langle C \rangle\rangle \gamma$, intuitively meaning: “The coalition C has a collective strategy to guarantee the satisfaction of the objective γ on every play enabled by that strategy.” Formally, ATL^* is a multi-agent extension of the branching time logic CTL^* , i.e., multimodal logic extending the linear-time temporal logic LTL —comprising the temporal operators **X** (“at the next state”), **G** (“always from now on”) and **U** (“until”)—with *strategic path quantifiers* $\langle\langle C \rangle\rangle$ indexed with coalitions C of players. There are two types of formulae of ATL^* , *state formulae*, which constitute the logic and that are evaluated at game states, and *path formulae*, that are evaluated on game plays. These are defined by mutual recursion with the following grammars, where $C \subseteq \text{Ag}$, $p \in \text{Prop}$: state formulae are defined by $\varphi ::= p \mid \neg \varphi \mid \varphi \wedge \varphi \mid \langle\langle C \rangle\rangle \gamma$, and path formulae by $\gamma ::= \varphi \mid \neg \gamma \mid \gamma \wedge \gamma \mid \mathbf{X}\gamma \mid \mathbf{G}\gamma \mid \gamma \mathbf{U} \gamma$.

The logic ATL^* is very expressive and that comes at a high computational price: satisfiability and model checking are **2ExpTime**-complete. A computationally better behaved fragment is the logic ATL , which is the multi-agent analogue of CTL , only involving state formulae defined by the following grammar, for $C \subseteq \text{Ag}$, $p \in \text{Prop}$: $\varphi ::= p \mid \neg \varphi \mid \varphi \wedge \varphi \mid \langle\langle C \rangle\rangle \mathbf{X}\varphi \mid \langle\langle C \rangle\rangle \mathbf{G}\varphi \mid \langle\langle C \rangle\rangle (\varphi \mathbf{U} \varphi)$. For this logic satisfiability and model checking are **ExpTime**-complete and **P**-complete, respectively. We will, however, build our extended logical formalism on the richer ATL^* because we will essentially need the path-based semantics for it.

Arithmetic Constraints. We define a simple language of arithmetic constraints to express conditions about the accumulated payoffs of players on a given play. For this purpose, we use a set $V_{\text{Ag}} = \{v_a \mid a \in \text{Ag}\}$ of special variables to refer to the accumulated payoffs of the players at a given state and denote by V_A the restriction of V_{Ag} to any group $A \subseteq \text{Ag}$. The payoffs can be integers, rationals¹, or any reals. We denote the domain of possible values of the payoffs, assumed to be a subset of the reals \mathbb{R} , by \mathbb{D} and use a set of constants symbols X , with $0 \in X$, for names of special real values (see further) to which we want to refer in the logical language.

For fixed sets X and $A \subseteq \text{Ag}$ we build the set $T(X, A)$ of *terms over X and A* from $X \cup V_A$ by applying addition, e.g. $v_a + v_b$. An evaluation of a term $t \in T(X, A)$ is a mapping $\eta : X \cup V_A \rightarrow \mathbb{D}$. We write $\eta \models t$ to denote that t is *satisfied* under the evaluation η . Moreover, if some order of the elements $X \cup V_A$ is clear from context, we also represent an evaluation as a tuple from $\mathbb{D}^{|X|+|V_A|}$ and often assume that elements from X have their canonic interpretation. The set $\text{AC}(X, A)$ of *arithmetic constraints* over X and A consists of all expressions of the form $t_1 * t_2$ where $*$ $\in \{<, \leq, =, \geq, >\}$ and $t_1, t_2 \in T(X, A)$. We use $\text{ACF}(X, A)$ to refer to the set of Boolean formulae over $\text{AC}(X, A)$; e.g. $(t_1 < t_2) \wedge (t_2 \geq t_3) \in \text{ACF}(X, A)$.

¹Note that models with rational payoffs behave essentially like models with integer payoffs, after once-off initial re-scaling.

for $t_1, t_2, t_3 \in T(X, A)$. We note that the language $\text{ACF}(X, A)$ is strictly weaker than Presburger arithmetic, as it involves neither quantifiers nor congruence relations.

We also consider the set $\text{APC}(X, A)$ of *arithmetic path constraints* being expressions of the type $w_a * c$ where $a \in \text{Ag}$, $*$ $\in \{<, \leq, =, \geq, >\}$ and $c \in X$. The meaning of w_a is to represent the value of the current play for the player a . That value can be defined differently, typically by computing the accumulated payoff over the entire play, by using a future discounting factor, or by taking the limit – if it exists – of the mean (average) accumulated payoff (cf. [20]). We note that the discounted, accumulated, mean or limit payoffs may take real values beyond the original domain of payoffs \mathbb{D} ; so, we consider the domain for X to be a suitable closure of \mathbb{D} .

3 Concurrent Game Models with Payoffs and Guards

We now extend concurrent game models with utility values for every action profile applied at every state and with guards that determine which actions are available to a player at a given configuration, consisting of a state and a utility vector, in terms of arithmetic constraints on the utility of that player.

Definition 1 A *guarded CGM with payoffs (GCGMP)* is a tuple $\mathfrak{M} = (\mathcal{S}, \text{payoff}, \{g_a\}_{a \in \text{Ag}}, \{d_a\}_{a \in \text{Ag}})$ where $\mathcal{S} = (\text{Ag}, \text{St}, \{\text{Act}_a\}_{a \in \text{Ag}}, \{\text{act}_a\}_{a \in \text{Ag}}, \text{out}, \text{Prop}, L)$ is a CGM and:

- $\text{payoff} : \text{Ag} \times \text{St} \times \text{Act}_{\text{Ag}} \rightarrow \mathbb{D}$ is a payoff function assigning at every state s and action profile applied at s a payoff to every agent. We write $\text{payoff}_a(s, \vec{\alpha})$ for $\text{payoff}(a, s, \vec{\alpha})$.
- $g_a : \text{St} \times \text{Act}_a \rightarrow \text{ACF}(X, \{a\})$, for each player $a \in \text{Ag}$, is a guard function that assigns for each state $s \in \text{St}$ and action $\alpha \in \text{Act}_a$ an arithmetic constraint formula $g_a(s, \alpha)$ that determines whether α is available to a at the state s given the current value of a 's accumulated payoff. The guard must enable at least one action for a at s . Formally, for each state $s \in \text{St}$, the formula $\bigvee_{\alpha \in \text{Act}_a} g_a(s, \alpha)$ must be valid. Moreover, a guard $g_a(s, \alpha)$ is called *state-based* if $g_a(s, \alpha) \in \text{ACF}(X)$.
- $d_a \in [0, 1]$ is a discount factor, for each $a \in \text{Ag}$, used in order to define realistically values of infinite plays for players or to reason about the asymptotic behavior of players' accumulated payoffs.

The guard g_a refines the function act_a from the definition of a CGM, which can be regarded as a guard function assigning to every state and action a constant arithmetic constraint `true` or `false`. In our definition the guards assigned by g_a only depend on the current state and the current accumulated payoff of a . The idea is that when the payoffs are interpreted as costs, penalties or, more generally, consumption of resources the possible actions of a player would depend on her current availability of utility/resources.

Example 1 Consider the GCGMP shown in Figure 1 with 2 players, I and II, and 3 states, where in every state each player has 2 possible actions, C (cooperate) and D (defect). The transition function is depicted in the figure. The normal form games associated with the states are respectively versions of the Prisoners Dilemma at state s_1 , Battle of the Sexes at state s_2 and Coordination Game at state s_3 .

The guards for both players are defined at each state so that the player can apply any action if she has a positive current accumulated payoff, may only apply action C if she has accumulated payoff 0; and must play an action maximizing her minimum payoff in the current game if she has a negative accumulated payoff. The discounting factors are 1 and the initial payoffs of both players are 0.

Configurations, plays, and histories. Let \mathfrak{M} be a GCGMP defined as above. A *configuration* (in \mathfrak{M}) is a pair (s, \vec{u}) consisting of a state s and a vector $\vec{u} = (u_1, \dots, u_k)$ of currently accumulated payoffs, one for each agent, at that state. Hereafter we refer to *accumulated payoffs* as *utility*, at a given state. We define the set of possible configurations as $\text{Con}(\mathfrak{M}) = \text{St} \times \mathbb{D}^{|\text{Ag}|}$. The *partial configuration transition function* is defined as $\widehat{\text{out}} : \text{Con}(\mathfrak{M}) \times \text{Act}_{\text{Ag}} \times \mathbb{N} \rightarrow \text{Con}(\mathfrak{M})$ such that $\widehat{\text{out}}((s, \vec{u}), \vec{\alpha}, l) = (s', \vec{u}')$ iff:

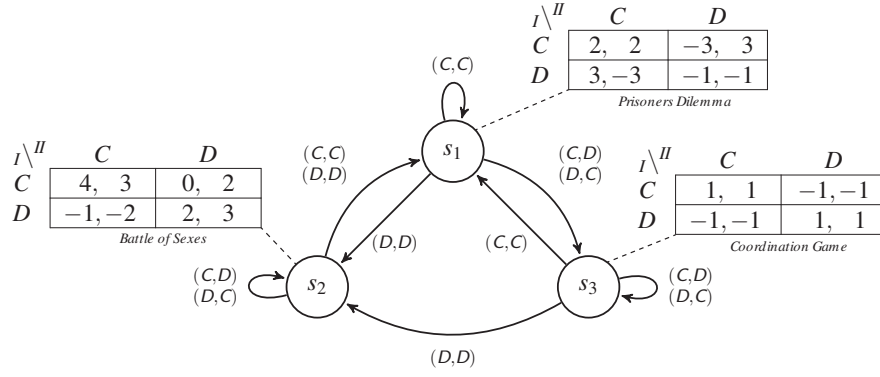


Figure 1: A simple GCGMP.

- (i) $\text{out}(s, \vec{\alpha}) = s'$ (s' is a successor of s if $\vec{\alpha}$ is executed).
- (ii) assigning the value u_a to v_a satisfies the guard $g_a(s, \alpha_a)$ for each $a \in \text{Ag}$, i.e. $u_a \models g_a(s, \alpha_a)$ (each agent's move α_a is enabled at s by the respective guard g_a applied to the current accumulated utility value u_a).
- (iii) $u'_a = u_a + d_a^l \cdot \text{payoff}_a(s, \vec{\alpha})$ for all $a \in \text{Ag}$ (i.e., the utility values change according to the utility function and the discounting rate where l denotes the number of steps that took place).

A GCGMP \mathfrak{M} with a designated initial configuration (s_0, \vec{u}_0) gives rise to a *configuration graph* on \mathfrak{M} consisting of all configurations in \mathfrak{M} reachable from (s_0, \vec{u}_0) by the configuration transition function. A *play* in a GCGMP \mathfrak{M} is an infinite sequence $\pi = c_0 \vec{\alpha}_0, c_1 \vec{\alpha}_1, \dots$ from $(\text{Con}(\mathfrak{M}) \times \text{Act})^\omega$ such that $c_n \in \widehat{\text{out}}(c_{n-1}, \vec{\alpha}_{n-1})$ for all $n > 0$. The set of all plays in \mathfrak{M} is denoted by $\text{Plays}_{\mathfrak{M}}$. Given a play π we use $\pi[i]$ and $\pi[i, \infty]$ to refer to the i th element and to the subplay starting in position i of π , respectively. A *history* is any finite initial sequence $h = c_0 \vec{\alpha}_0, c_1 \alpha_1, \dots, c_n \in (\text{Con}(\mathfrak{M}) \times \text{Act})^* \text{Con}(\mathfrak{M})$ of a play in $\text{Plays}_{\mathfrak{M}}$. The set of all histories is denoted by $\text{Hist}_{\mathfrak{M}}$. For any history h we also define $h[i]$ as for plays and additionally $h[\text{last}]$ and $h[i, j]$ to refer to the last state on h and to the sub-history between i and j , respectively. Finally, we introduce functions \cdot^c , \cdot^u , and \cdot^s which denote the projection of a given play or history to the sequence of its configurations, utility vectors, and states, respectively. For illustration, let us consider the play $\pi = c_0 \vec{\alpha}_0, c_1 \vec{\alpha}_1, \dots$. We have that $\pi[i, \infty] = c_i \vec{\alpha}_i, c_{i+1} \vec{\alpha}_{i+1}, \dots$; $\pi[i] = c_i \vec{\alpha}_i$; $\pi^c[i, \infty] = c_i, c_{i+1}, \dots$; $\pi^c[i] = c_i$; $\pi^u[i] = \vec{\alpha}_i$; $\pi^u[i] = v_i$; and $\pi^s[i] = s_i$ where $c_i = (s_i, \vec{u}_i)$.

Example 2 Some possible plays starting from s_1 in Example 1 are given in the following where we assume that the initial accumulated payoff is 0 for both agents. We note that this implies that the first action taken by any agent is always C.

1. Both players cooperate forever: $(s_1, 0, 0), (s_1, 2, 2), (s_1, 4, 4), \dots$
2. After the first round both players defect and the play moves to s_2 , where player I chooses to defect whereas II cooperates. Then I must cooperate while II must defect but at the next round can choose any action, so a possible play is: $(s_1, 0, 0), (s_1, 2, 2), (s_2, 1, 1), (s_2, 0, -1), (s_2, 0, 1), (s_2, 0, 3), (s_2, 0, 5), \dots$
3. After the first round player I defects while II cooperates and the play moves to s_3 , where they can get stuck indefinitely, until – if ever – they happen to coordinate, so a possible play is: $(s_1, 0, 0), (s_1, 2, 2), (s_3, 5, -2), (s_3, 4, -3), (s_3, 3, -4), \dots (s_3, 0, -7), (s_3, -1, -8), \dots$

Note, however, that once player I reaches accumulated payoff 0 he may only apply C at that round, so if player II has enough memory or can observe the accumulated payoffs of I he can use the

opportunity to coordinate with I at that round by cooperating, thus escaping the trap at s_3 and making a sure transition to s_2 .

4. If, however, the guards did not force the players to play C when reaching accumulated payoffs 0, then both players could plunge into an endless misery if the play reaches s_3 .

Strategies. A strategy of a player a is a function $s_a : \text{Hist} \rightarrow \text{Act}$ such that if $s_a(h) = \alpha$ then $h^u[\text{last}]_a \models g_a(h^s[\text{last}], \alpha)$; that is, actions prescribed by a strategy must be enabled by the guard. Our definition of strategy is based on histories of configurations and actions, so it extends the notion of strategy from [5] where it is defined on histories of states, and includes strategies, typically considered e.g. in the study of repeated games, where often strategies prescribe to the player an action dependent on the previous action, or history of actions, of the other player(s). Such are, for instance, TIT-FOR-TAT or GRIM-TRIGGER in repeated Prisoners Dilemma; likewise for various card games, etc. Since our notion of strategy is very general, it easily leads to undecidable model checking problems. So, we also consider some natural restrictions, such as: *state-based*, *action-based* or *configuration-based*, *memoryless*, *bounded memory*, of *perfect recall* strategies². Here we adopt a generic approach and assume that two classes of strategies \mathcal{S}^P and \mathcal{S}^O are fixed as parameters, with respect to which the proponents and opponents select their strategies, respectively. The proponent coalition A selects a \mathcal{S}^P -strategy s_A (i.e. one agreeing with the class \mathcal{S}^P) while the opponent coalition $\text{Ag} \setminus A$ selects a \mathcal{S}^O -strategy $s_{\text{Ag} \setminus A}$. The outcome play $\text{outcome_play}_{\mathfrak{M}}(c, (s_A, s_{\text{Ag} \setminus A}), l)$ in a given GCGMP \mathfrak{M} determines the play emerging from the execution of the (complete) strategy profile $(s_A, s_{\text{Ag} \setminus A})$ from configuration c in \mathfrak{M} .

4 The Logic: Quantitative ATL*

We now extend the logic ATL* to the logic QATL* with atomic quantitative objectives being state or path arithmetic constraints over the players' accumulated payoffs. The semantics of QATL* naturally extends the semantics of ATL* over GCGMPs, but parameterised with the two classes of strategies \mathcal{S}^P and \mathcal{S}^O .

Definition 2 (The logic QATL*) *The language of QATL* consists of state formulae φ , which constitute the logic, and path formulae γ , generated as follows, where $A \subseteq \text{Ag}$, $\text{ac} \in \text{AC}$, $\text{apc} \in \text{APC}$, and $p \in \text{Prop}$:*

$$\varphi ::= p \mid \text{ac} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle\gamma \text{ and } \gamma ::= \varphi \mid \text{apc} \mid \neg\gamma \mid \gamma \wedge \gamma \mid \mathbf{X}\gamma \mid \mathbf{G}\gamma \mid \gamma\mathbf{U}\gamma.$$

Let \mathfrak{M} be a GCGMP, c a configuration, $\varphi, \varphi_1, \varphi_2$ state-formulae, $\gamma, \gamma_1, \gamma_2$ path formulae, and $l \in \mathbb{N}$. Further, let \mathcal{S}^P and \mathcal{S}^O be two classes of strategies as described above. The semantics of the path constraints is specified according to the limit-averaging or discounting mechanism adopted for computing the value of a play for a player. Then the truth of a QATL formula at a position of a configuration in \mathfrak{M} is defined by mutual recursion on state and path formulae as follows:*

$$\mathfrak{M}, c, l \models p \text{ for } p \in \text{Prop} \text{ iff } p \in L(c^s); \quad \mathfrak{M}, c, l \models \text{ac} \text{ for } \text{ac} \in \text{AC} \text{ iff } c^u \models \text{ac},$$

$$\mathfrak{M}, c, l \models \langle\langle A \rangle\rangle\gamma \text{ iff there is a collective } \mathcal{S}^P\text{-strategy } s_A \text{ for } A \text{ such that for all collective } \mathcal{S}^O\text{-strategies } s_{\text{Ag} \setminus A} \text{ for } \text{Ag} \setminus A \text{ we have that } \mathfrak{M}, \text{outcome_play}_{\mathfrak{M}}(c, (s_A, s_{\text{Ag} \setminus A}), l), l \models \gamma.$$

$$\mathfrak{M}, \pi, l \models \varphi \text{ iff } \mathfrak{M}, \pi[0], l \models \varphi; \quad \mathfrak{M}, \pi, l \models \text{apc} \text{ iff } \pi^u, l \models \text{apc} \text{ for } \text{apc} \in \text{APC}.$$

$$\mathfrak{M}, \pi, l \models \mathbf{G}\gamma \text{ iff } \mathfrak{M}, \pi[i], l+i \models \gamma \text{ for all } i \in \mathbb{N}_0,$$

$$\mathfrak{M}, \pi, l \models \mathbf{X}\gamma \text{ iff } \mathfrak{M}, \pi[1], l+1 \models \gamma,$$

$$\mathfrak{M}, \pi, l \models \gamma_1 \mathbf{U} \gamma_2 \text{ iff there is } j \in \mathbb{N}_0 \text{ such that } \mathfrak{M}, \pi[j], l+j \models \gamma_2 \text{ and } \mathfrak{M}, \pi[i], l+i \models \gamma_1 \text{ for all } 0 \leq i < j.$$

Ultimately, we define $\mathfrak{M}, c \models \varphi$ as $\mathfrak{M}, c, 1 \models \varphi$. Moreover, if not clear from context, we also write $\models_{(\mathcal{S}^P, \mathcal{S}^O)} \text{ for } \models$.

²We note that all strategies need to be consistent with the guards, so state-based strategies are only applicable in models where the guards only take into account the current state, but not the accumulated payoffs.

The semantics presented above extends the standard semantics for ATL^* and is amenable to various refinements and restrictions, to be studied further. For instance, if appropriate, an alternative semantics can be adopted, based on irrevocable strategies [1] or, more generally, on strategy contexts [8] or other mechanisms for strategy commitment and release [2]. Also, the nested operators as defined here access the accumulated utility values and require plays to be infinite. Similarly to [9], one can consider variants of these settings which may yield decidable model checking and better complexity results.

As the logic $QATL^*$ extends ATL^* , it allows expressing all purely qualitative ATL^* properties. It can also express purely quantitative properties, e.g.: $\langle\langle\{a\}\rangle\rangle G(v_a > 0)$ meaning “Player a has a strategy to maintain his accumulated payoff to be always positive”, or $\langle\langle A \rangle\rangle (w_a \geq 3)$ meaning “The coalition A has a strategy that guarantees the value of the play for player a to be at least 3”. Moreover, $QATL^*$ can naturally express combined qualitative and quantitative properties, e.g. $\langle\langle\{a, b\}\rangle\rangle ((v_a + v_b \geq 1)Up)$, etc.

Example 3 *The following $QATL^*$ state formulae are true at state s_1 of the GCGMP in Example 1, where p_i is an atomic proposition true only at state s_i , for each $i = 1, 2, 3$:*

- (i) $\langle\langle\{I, II\}\rangle\rangle F(p_1 \wedge v_I > 100 \wedge v_{II} > 100) \wedge \langle\langle\{I, II\}\rangle\rangle XX\langle\langle\{II\}\rangle\rangle (G(p_2 \wedge v_I = 0) \wedge F v_{II} > 100)$.
- (ii) $\neg\langle\langle\{I\}\rangle\rangle G(p_1 \vee v_I > 0) \wedge \neg\langle\langle\{I, II\}\rangle\rangle F(p_3 \wedge G(p_3 \wedge (v_I + v_{II} > 0)))$.

5 (Un)Decidability: Related Work and Some Preliminary Results

Generally, the GCGMP models are too rich and the language of $QATL^*$ is too expressive to expect computational efficiency, or even decidability, of either model checking or satisfiability testing. Some preliminary results and related work show that model checking of $QATL^*$ in GCGMPs is undecidable under rather weak assumptions, e.g. if the proponents or the opponents can use memory-based strategies. These undecidability results are not surprising as GCGMPs are closely related to Petri nets and vector addition systems and it is known that model checking over them is generally undecidable. In [13], for example, this is shown for fragments of CTL and (state-based) LTL over Petri nets. Essentially, the reason is that the logics allow to encode a “test for zero”; for Petri nets this means to check whether a place contains a token or not. In our setting undecidability follows for the same reason, and we will sketch some results below.

Undecidability results. The logic $QATL$ restricts $QATL^*$ in the same way as ATL restricts ATL^* , due to lack of space we skip the formal definition. As a first result we show that model checking $QATL$ is undecidable even if only the proponents are permitted to use perfect recall strategies and the opponents are bound to memoryless strategies. More formally, let S^{pr} denote the class of perfect recall state-based strategies and S^m the class of memoryless state-based strategies. That is, strategies of the former class are functions of type $St^* \rightarrow Act$ and of the latter class functions of type $St \rightarrow Act$.

Undecidability can be shown using ideas from e.g. [9, 13]. Here, we make use of the construction of [9] to illustrate the undecidability by simulating a two-counter machine (TCM). A TCM [17] can be considered as a transition system equipped with two integer counters that enable/disable transitions. Each step of the machine depends on the current state, symbol on the tape, and the counters, whether they are zero or not. After each step the counters can be incremented (+1), or decremented (−1), the latter only if the respective counter is not zero. A TCM is essentially a (nondeterministic) push-down automaton with two stacks and exactly two stack symbols (one of them is the initial stack symbol) and has the same computation power as a Turing machine (cf. [17]). A *configuration* is a triple (s, w_1, w_2) describing the current state (s), the value of counter 1 (w_1) and of counter 2 (w_2). A *computation* δ is a sequence of subsequent configurations effected by transitions.

For the simulation, we associate each counter with a player. The player's accumulated payoff encodes the counter value; actions model the increment/decrement of the counters; guards ensure that the actions respect the state of the counters. The accepting states of the two-counter machine are encoded by a special proposition *halt*. Now, the following lemma stating the soundness of the simulation can be proved:

Lemma 1 (Reduction) *For any two-counter machine A we can construct a finite GCGMP \mathfrak{M}^A with two players and proposition *halt* such that the following holds: A halts on the empty input iff \mathfrak{M}^A contains a play π with $\pi^c = (s^0, (v_1^0, v_2^0))(s^1, (v_1^1, v_2^1)) \dots$ such that there exists $j \in \mathbb{N}$ with *halt* $\in L(s^j)$.*

The next theorem gives two cases for which the model checking problem is undecidable. By the previous Lemma we have to ensure that the halting state is reached which can be expressed by $\langle\langle 1 \rangle\rangle \mathbf{F} \text{halt}$. We can also use purely state-based guards and encode the consistency checks in the formula as follows: $\langle\langle 1 \rangle\rangle (v_1 \geq 0 \wedge v_2 \geq 0 \wedge e_1 \rightarrow v_a = 0 \wedge e_2 \rightarrow v_2 = 0) \mathbf{U} \text{halt}$ where the proposition e_i is added to the model to indicate that the value of counter i is zero. Note that this information is static and obtained from the transition relation of the automaton.

Proposition 1 *Model checking the logic QATL is undecidable, even for the 2 agent case and no nested cooperation modalities, where $\mathcal{S}^p = S^{pr}$ and $\mathcal{S}^o = S^m$. This does even hold either for formulae not involving arithmetic constraints, or for state-based guards.*

Restoring decidability. There are some natural semantic and syntactic restrictions of QATL* where decidability may be restored; these include for instance, the enabling of only memoryless strategies, imposing non-negative payoffs, constraints on the transition graph of the model, bounds on players utilities etc. For instance, the main reason for the undecidability result above is the possibility for negative payoffs that allow for decrementing the accumulated payoffs and thus simulating the TCM operations. Therefore, a natural restriction in the quest for restoring decidability is to consider only GCGMP models with non-negative payoffs. In this case the accumulated payoffs increase monotonically over every play of the game, and therefore the truth values of every arithmetic constraint occurring in the guards and in the formula eventually stabilize in a computable way, which in the long run reduces the model checking of any QATL-formula in an GCGMP to a model checking of an ATL-formula in a CGM. One can thus obtain decidability of the model checking of the logic QATL in finite GCGMP with non-negative payoffs and perfect information. We will discuss these and other decidability results in a future work, where we will also consider restrictions similar to [9].

6 Concluding Remarks

This paper proposes a long-term research agenda bringing together issues, techniques and results from several research fields. It aims at bridging the two important aspects of reasoning about objectives and abilities of players in multi-player games: quantitative and qualitative, and eventually providing a uniform framework for strategic reasoning in multi-agent systems.

Acknowledgements: We thank the anonymous referees for detailed and helpful comments and additional references.

References

- [1] T. Ågotnes, V. Goranko & W. Jamroga (2007): *Alternating-time Temporal Logics with Irrevocable Strategies*. In D. Samet, editor: *Proceedings of TARK XI*, pp. 15–24, doi:10.1145/1324249.1324256.
- [2] T. Ågotnes, V. Goranko & W. Jamroga (2008): *Strategic Commitment and Release in Logics for Multi-Agent Systems (Extended abstract)*. Technical Report IfI-08-01, Clausthal University of Technology.

- [3] Natasha Alechina, Brian Logan, Nguyen Hoang Nga & Abdur Rakib (2011): *Logic for coalitions with bounded resources*. *J. Log. Comput.* 21(6), pp. 907–937, doi:10.1093/logcom/exq032.
- [4] Luca de Alfaro, Thomas A. Henzinger & Orna Kupferman (2007): *Concurrent reachability games*. *Theor. Comput. Sci.* 386(3), pp. 188–217, doi:10.1016/j.tcs.2007.07.008.
- [5] R. Alur, T. A. Henzinger & O. Kupferman (2002): *Alternating-Time Temporal Logic*. *Journal of the ACM* 49, pp. 672–713, doi:10.1145/585265.585270.
- [6] Patricia Bouyer, Romain Brenguier, Nicolas Markey & Michael Ummels (2011): *Nash Equilibria in Concurrent Games with Büchi Objectives*. In S. Chakraborty & A. Kumar, editors: *FSTTCS'2011 LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 375–386, doi:10.4230/LIPIcs.FSTTCS.2011.375.
- [7] Patricia Bouyer, Romain Brenguier, Nicolas Markey & Michael Ummels (2012): *Concurrent Games with Ordered Objectives*. In L. Birkedal, editor: *Proc. of FoSSaCS'2012*, Springer LNCS, vol. 7213, pp. 301–315, doi:10.1007/978-3-642-28729-9_20.
- [8] T. Brihaye, A. Da Costa, F. Laroussinie & N. Markey (2008): *ATL with Strategy Contexts and Bounded Memory*. Technical Report LSV-08-14, ENS Cachan, doi:10.1007/978-3-540-92687-0_7.
- [9] Nils Bulling & Berndt Farwer (2010): *On the (Un-)Decidability of Model-Checking Resource-Bounded Agents*. In H. Coelho & M. Wooldridge, editors: *Proc. of ECAI 2010*, IOS Press, Amsterdam, pp. 567–572, doi:10.3233/978-1-60750-606-5-567.
- [10] Krishnendu Chatterjee, Luca de Alfaro & Thomas A. Henzinger (2011): *Qualitative concurrent parity games*. *ACM Trans. Comput. Log.* 12(4), p. 28, doi:10.1145/1970398.1970404.
- [11] Krishnendu Chatterjee & Laurent Doyen (2012): *Energy parity games*. *Theor. Comput. Sci.* 458, pp. 49–60, doi:10.1016/j.tcs.2012.07.038.
- [12] Krishnendu Chatterjee & Thomas A. Henzinger (2012): *A survey of stochastic ω -regular games*. *J. Comput. Syst. Sci.* 78(2), pp. 394–413. Available at <http://dx.doi.org/10.1016/j.jcss.2011.05.002>.
- [13] Javier Esparza: *Decidability of Model Checking for Infinite-State Concurrent Systems*. *Acta Informatica* 34, pp. 85–107, doi:10.1007/s002360050074.
- [14] Javier Esparza (1998): *Decidability and complexity of Petri net problems - an Introduction*. In: *In Lectures on Petri Nets I: Basic Models*, Springer-Verlag, pp. 374–428, doi:10.1007/3-540-65306-6_20.
- [15] Valentin Goranko & Wojciech Jamroga (2004): *Comparing Semantics of Logics for Multi-agent Systems*. *Synthese* 139(2), pp. 241–280, doi:10.1023/B:SYNT.0000024915.66183.d1.
- [16] Erich Grädel & Michael Ummels (2008): *Solution Concepts and Algorithms for Infinite Multiplayer Games*. In Krzysztof Apt & Robert van Rooij, editors: *New Perspectives on Games and Interaction*, Texts in Logic and Games 4, Amsterdam University Press, pp. 151–178. Available at <http://www.logic.rwth-aachen.de/~ummels/knaw07.pdf>.
- [17] JE Hopcroft & JD Ullman (1979): *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, Massachusetts, doi:10.1145/568438.568455.
- [18] W. Jamroga & T. Ågotnes (2007): *Constructive Knowledge: What Agents Can Achieve under Incomplete Information*. *Journal of Applied Non-Classical Logics* 17(4), pp. 423–475, doi:10.3166/jancl.17.423-475.
- [19] Dario Della Monica, Margherita Napoli & Mimmo Parente (2011): *On a Logic for Coalitional Games with Priced-Resource Agents*. *Electr. Notes Theor. Comput. Sci.* 278, pp. 215–228, doi:10.1016/j.entcs.2011.10.017.
- [20] M. Osborne & A. Rubinstein (1994): *A Course in Game Theory*. MIT Press.
- [21] M. Pauly (2002): *A Modal Logic for Coalitional Power in Games*. *J. of Logic and Computation* 12(1), pp. 149–166, doi:10.1093/logcom/12.1.149.
- [22] Sophie Pinchinat (2007): *A Generic Constructive Solution for Concurrent Games with Expressive Constraints on Strategies*. In K. Namjoshi et al, editor: *Proc. of ATVA'2007*, Springer LNCS, vol. 4762, pp. 253–267, doi:10.1007/978-3-540-75596-8_19.