# Combining Quantitative and Qualitative Reasoning in Concurrent Multi-player Games

JAAMAS Track

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

Valentin Goranko
Stockholm University
Stockholm, Sweden
valentin.goranko@philosophy.su.se

## ABSTRACT

We propose and study a general framework for modelling and formal reasoning about multi-agent systems and, in particular, multi-stage games where both *quantitative* and *qualitative* objectives and constraints are involved. Our models enrich concurrent game models with payoffs and guards on actions associated with each state of the model. We propose a quantitative extension of the logic ATL* that enables combination of quantitative and qualitative reasoning. We illustrate the framework with some examples and then consider the model-checking problems arising in it and establish some general undecidability and decidability results for them.

## KEYWORDS

Multi-stage Games; Quantitative and Qualitative Reasoning; Temporal Logic ATL; Model Checking; Decidability; Undecidability

**ACM Reference Format:**
Nils Bulling and Valentin Goranko. 2022. Combining Quantitative and Qualitative Reasoning in Concurrent Multi-player Games: JAAMAS Track. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

## 1 INTRODUCTION

Quantitative and qualitative reasoning about agents and multi-agent systems is pervasive in many areas of AI and game theory, including multi-agent planning and intelligent robotics. In particular, the studies of cooperative and non-cooperative multi-player games deal with both aspects of strategic abilities of agents, but usually separately. Quantitative reasoning concerns the abilities of agents to achieve *quantitative objectives*, such as optimizing payoffs (e.g., maximizing rewards or minimizing cost) or, more generally, ensuring most preferred outcomes. This tradition comes from game theory and usually studies one-shot normal form games, or their (finitely or infinitely) repeated versions, or extensive form games. On the other hand, qualitative reasoning, coming mainly from logic and computer science, is about strategic abilities of players for achieving *qualitative objectives*: reaching or maintaining states with desired properties, e.g., winning states or safe states, etc. Put as a slogan, quantitative reasoning is concerned with *how players can become maximally rich, or how to pay as little cost as possible*, while qualitative reasoning is about *how players can achieve a state of 'happiness', e.g. winning, or how to avoid reaching a state of 'unhappiness' (losing) in the game.* Often both types of reasoning about multi-agent systems are essential and must be explored interactively. For instance, in multi-agent planning and robotics it is important to achieve the agents' qualitative goals while satisfying various quantitative constraints on time and resource consumption. This motivates the development of a modelling framework for combining qualitative and quantitative reasoning, which is the main objective of this work, presented in detail in the full paper [4] (substantially extending the earlier version [3]), where a wide range of related works are also surveyed.

*Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.* 

## 2 CONCURRENT GAME MODELS WITH PAYOFFS AND GUARDS

A *concurrent game model* ([1]) (CGM) as a tuple
$\mathcal{S} = (\mathsf{Ag}, \mathsf{St}, \{\mathsf{Act}_\mathsf{a}\}_{\mathsf{a}\in\mathsf{Ag}}, \{\mathsf{act}_\mathsf{a}\}_{\mathsf{a}\in\mathsf{Ag}}, \mathsf{out}, \mathsf{Prop}, \mathsf{L})$ comprising:

⊳ a non-empty, fixed set of *players (agents)* $\mathsf{Ag} = \{1, \dots, k\}$ and a set of actions $\mathsf{Act}_\mathsf{a} \neq \emptyset$ for each $\mathsf{a} \in \mathsf{Ag}$. For any $A \subseteq \mathsf{Ag}$ we denote $\mathsf{Act}_A := \prod_{\mathsf{a}\in A} \mathsf{Act}_\mathsf{a}$ and will use $\alpha_A$ to denote a tuple from $\mathsf{Act}_A$. In particular, $\mathsf{Act}_\mathsf{Ag}$ is the set of all possible *action profiles* in $\mathcal{S}$.

⊳ a non-empty set of *game states* St.

⊳ for each $\mathsf{a} \in \mathsf{Ag}$, a map $\mathsf{act}_\mathsf{a} : \mathsf{St} \to \mathcal{P}(\mathsf{Act}_\mathsf{a})$ setting for each state $s$ the actions available to a at $s$.

⊳ a *transition function* $\mathsf{out} : \mathsf{St}\times\mathsf{Act}_\mathsf{Ag} \to \mathsf{St}$ that assigns to every state $q$ and action profile $\alpha_\mathsf{Ag} = \langle \alpha_1, \dots, \alpha_\mathsf{k}\rangle$, such that $\alpha_\mathsf{a} \in \mathsf{act}_\mathsf{a}(q)$ for every $\mathsf{a} \in \mathsf{Ag}$ (i.e., every $\alpha_\mathsf{a}$ that can be executed by player a in state $q$), the (deterministic) *successor (outcome) state* $\mathsf{out}(q, \alpha_\mathsf{Ag})$.

⊳ a set of atomic propositions Prop and a labelling function $\mathsf{L} : \mathsf{St} \to \mathcal{P}(\mathsf{Prop})$.

A CGM represents a multi-agent transition system where at any state all players in choose and execute their actions synchronously, and the combination of these actions determines the transition to a (unique) successor state in the CGM. A *play* in a CGM $\mathcal{M}$ is an infinite sequence of such subsequent states. Alternatively, concurrent game models can be viewed as multi-stage combinations of normal form games each associated with a state of the system. For further technical details we refer to [1], [5], [6, Ch.9],

We are now going to introduce an extension of concurrent game models that can be viewed as a *multi-stage game*, where at every stage the result of the simultaneous collective action of all players is two-fold: first, players receive individual payoffs – just like in the normal form games and repeated games traditionally studied in game theory – and second, a transition is effected to (possibly) another state, where (possibly) another such game is played, etc., infinitely. More precisely, we extend concurrent game models with *utility payoffs* for every action profile applied at every state. Thus, every action profile applied at a given state has now two effects:

i) it assigns a payoff to each player, and
ii) determines a transition to a new state, where the game associated with it is played at the next round of the play.

Besides, we also add individual *guards* that determine which actions are available to a player at a given *configuration* consisting of a state and the vector of current *accumulated utilities* for each player, i.e. the current sum of all payoffs the player has received in the course of the current play of the game.

To give the formal definition, we need some preliminaries. An *arithmetic constraint* over a set of constants $X$ in a fixed numerical domain $\mathbb{D}$ and a set of agents $A$ is any expression of the form $t_1 * t_2$ where $* \in \{<, \leq, =, \geq, >\}$ and $t_1, t_2$ are terms built over a set $X \cup V_A$ by applying addition, where $V_A = \{v_\mathsf{a} \mid \mathsf{a} \in A\}$ consists of special variables used to refer to the accumulated utilities of the players in $A$ at the current state of the game. The set of these arithmetic constraints is denoted by $\mathsf{AC}(X, A)$. For any $\mathsf{a} \in \mathsf{Ag}$, an *(individual) a-guard* is an arithmetic constraint formula $\mathsf{ac} \in \mathsf{ACF}(X, \{\mathsf{a}\})$.

*Definition 2.1.* A *guarded CGM with payoffs (GCGMP)* is a tuple $\mathcal{M} = (\mathcal{S}, \mathsf{payoff}, \mathsf{grd})$ consisting of:

▷ a CGM $\mathcal{S} = (\mathsf{Ag}, \mathsf{St}, \{\mathsf{Act_a}\}_{\mathsf{a}\in\mathsf{Ag}}, \{\mathsf{act_a}\}_{\mathsf{a}\in\mathsf{Ag}}, \mathsf{out}, \mathsf{Prop}, \mathsf{L})$,

▷ a *payoff function*, $\mathsf{payoff} : \mathsf{Ag} \times \mathsf{St} \times \mathsf{Act_{Ag}} \to \mathbb{D}$ assigning for every agent a, state $s$, and action profile $\alpha$ applied at $s$ a *payoff* $\mathsf{payoff_a}(s, \alpha)$ to that agent.

▷ a *guard function* $\mathsf{grd} : \mathsf{Ag}\times\mathsf{St}\times\mathsf{Act_{Ag}} \to \mathsf{ACF}(X, \mathsf{Ag})$, such that for each $\mathsf{a} \in \mathsf{Ag}$, state $s \in \mathsf{St}$ and action $\alpha$, the guard $\mathsf{grd}(\mathsf{a}, s, \alpha)$, also denoted $\mathsf{grd_a}(s, \alpha)$, is an arithmetic constraint formula in $\mathsf{ACF}(X, \{\mathsf{a}\})$ that determines whether $\alpha$ is enabled for a at state $s$, given the value of a's current utility. It is required that grd is a total function.

*Example 2.2.* The GCGMP shown in Figure 1 involves 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 full details of the example, incl. the the guards, are in [4].

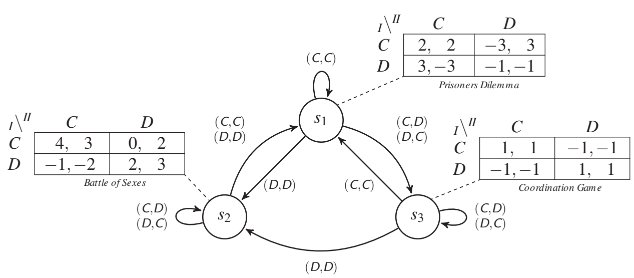


**Figure 1: A simple GCGMP combining 3 games.**

## 3 THE LOGIC FOR QUANTITATIVE AND QUALITATIVE REASONING QATL*

### 3.1 Syntax, Semantics and Expressiveness

QATL* extends the Alternating time Temporal Logic ATL* [1] with atomic quantitative objectives expressed by arithmetic constraints from a set AC over the players' currently accumulated utilities. The language of QATL* consists of *state formulae* $\varphi$, and *path formulae* $\gamma$, generated as follows, where $A \subseteq \mathsf{Ag}$ $\mathsf{ac} \in \mathsf{AC}$, and $p \in \mathsf{Prop}$:

$\varphi ::= p \mid \mathsf{ac} \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle A \rangle\rangle \gamma;$

$\gamma ::= \varphi \mid \neg\gamma \mid (\gamma \wedge \gamma) \mid \mathbf{X}\,\gamma \mid \mathbf{G}\,\gamma \mid (\gamma\,\mathbf{U}\,\gamma).$

The semantics of QATL* naturally extends the semantics of ATL* over GCGMP, based on the notion of *truth of a state formula $\varphi$ at a configuration $c$ in a GCGMP $\mathcal{M}$*, denoted $\mathcal{M}, c \models \varphi$, and possibly parameterised with two classes of admissible strategies, $\mathcal{S}^p$ for the proponents and $\mathcal{S}^o$ for the opponents. See the full details of the formal semantics in [4].

The logic QATL* is quite expressive. Besides capturing all purely qualitative, ATL*-definable properties, the logic QATL* can also express purely quantitative properties like $\langle\langle\{\mathsf{a}\}\rangle\rangle \mathbf{G}\,(v_\mathsf{a} > 0)$, meaning "*Player* a *has a strategy to maintain his accumulated utility to be always positive*". QATL* can also express combined qualitative and quantitative properties, e.g. $\langle\langle\{\mathsf{a}\}\rangle\rangle((\mathtt{a\ is\ happy})\ \mathbf{U}\,(v_\mathsf{a} \geq 10^6))$, saying "*Player* a *has a strategy to stay happy until* a *becomes a millionaire*", or $\langle\langle\{\mathsf{a},\mathsf{b}\}\rangle\rangle((v_\mathsf{a} + v_\mathsf{b} > v_\mathsf{c})\ \mathbf{U}\,\mathbf{G}\,(\mathtt{safe}))$, saying "*Players* a *and* b *have a joint strategy to keep their joint accumulated utility greater than the one of* c *until a the system enters a safe region and remains there forever.*" More such examples, based on Example 2.2 and on another one, can be found in [4].

### 3.2 Some Results on Model Checking in QATL*

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. Indeed, in [4] we show that model checking of QATL*– and even of QATL – in a GCGMP is undecidable under rather weak assumptions, e.g., in the 2-agent case, with state-based guards, where the proponent or the opponent can use finite memory strategies, effectively definable by means of finite transducers. These undecidability results are not surprising, as GCGMPs are technically closely related to Petri nets and vector addition systems with states (VASS) and it is known that logic-based model checking over them is generally undecidable. For example, in [7] this is shown for fragments of CTL and (state-based) LTL over Petri nets. Essentially, the reason is that these logics allow encoding a "test for zero" over such models; 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 have obtained such results in [4]. The reduction is done by applying ideas from [2] to simulate a *two-counter machine* (TCM) (aka *two-counter automaton*, or *2-register Minsky machine* [9]).

Despite the wide-ranging undecidability results, there are some natural semantic and syntactic restrictions of QATL* where decidability of the model checking problem may be restored, by making the configuration space and the strategy search space finite. Such restrictions include: the enabling of only memoryless strategies, imposing non-negative payoffs, constraints on the transition graph of the model, restrictions of the arithmetical constraints and guards ensuring bounded players' accumulated utilities, etc. In [4] we outline one such non-trivial case, using a result by Karp and Miller [8] for the coverability problem for vector addition systems, and discuss some other ideas and conjectures. Identifying sufficiently expressive fragments of QATL* and restrictions on GCGMP models with decidable model checking is a major direction for future work.

## REFERENCES


[1] R. Alur, T. A. Henzinger, and O. Kupferman. 2002. Alternating-Time Temporal Logic. *J. ACM* 49 (2002), 672–713. https://doi.org/10.1145/585265.585270

[2] Nils Bulling and Berndt Farwer. 2010. On the (Un-)Decidability of Model-Checking Resource-Bounded Agents. In *Proc. of ECAI 2010*, H. Coelho and M. Wooldridge (Eds.). IOS Press, Amsterdam, 567–572. https://doi.org/10.3233/978-1-60750-606-5-567

[3] Nils Bulling and Valentin Goranko. 2013. How to Be Both Rich and Happy: Combining Quantitative and Qualitative Strategic Reasoning about Multi-Player Games (Extended Abstract). In *Proceedings of the 1st International Workshop on Strategic Reasoning (Electronic Proceedings in Theoretical Computer Science)*. Rome, Italy, 33–41. https://doi.org/10.4204/EPTCS.112.8 arXiv:arXiv/1303.0789

[4] Nils Bulling and Valentin Goranko. 2022. Combining quantitative and qualitative reasoning in concurrent multi-player games. *Auton. Agents Multi Agent Syst.* 36, 1 (2022), 2. https://doi.org/10.1007/s10458-021-09531-9

[5] Nils Bulling, Valentin Goranko, and Wojciech Jamroga. 2015. Logics for Reasoning About Strategic Abilities in Multi-player Games. In *Models of Strategic Reasoning - Logics, Games, and Communities*, Johan van Benthem, Sujata Ghosh, and Rineke Verbrugge (Eds.). Lecture Notes in Computer Science, Vol. 8972. Springer, 93–136. https://doi.org/10.1007/978-3-662-48540-8

[6] Stéphane Demri, Valentin Goranko, and Martin Lange. 2016. *Temporal Logics in Computer Science*. Cambridge University Press.

[7] Javier Esparza. 1997. Decidability of Model Checking for Infinite-State Concurrent Systems. *Acta Informatica* 34 (1997), 85–107. https://doi.org/10.1007/s002360050074

[8] R. M. Karp and R. E. Miller. 1969. Parallel Program Schemata. *J. Comput. System Sci.* 3, 2 (1969), 147–195.

[9] M. Minsky. 1967. *Computation, Finite and Infinite Machines*. Prentice Hall.