Graded Alternating-Time Temporal Logic

Marco Faella[1], Margherita Napoli[2], and Mimmo Parente[2]

[1] Università di Napoli "Federico II", Via Cintia, 80126 - Napoli, Italy
[2] Università di Salerno, Via Ponte don Melillo, 84084 - Fisciano (SA), Italy

Abstract. Graded modalities enrich the universal and existential quantifiers with the capability to express the concept of *at least k* or *all but k*, for a non-negative integer k. Recently, temporal logics such as μ-calculus and Computational Tree Logic, CTL, augmented with graded modalities have received attention from the scientific community, both from a theoretical side and from an applicative perspective. Both μ-calculus and CTL naturally apply as specification languages for *closed* systems: in this paper, we add graded modalities to the Alternating-time Temporal Logic (ATL) introduced by Alur et al., to study how these modalities may affect specification languages for *open* systems. We present, and compare with each other, three different semantics. We first consider a natural interpretation which seems suitable to off-line synthesis applications and then we restrict it to the case where players can only employ memoryless strategies. Finally, we strengthen the logic by means of a different interpretation which may find application in the verification of fault-tolerant controllers. For all the interpretations, we efficiently solve the model-checking problem both in the concurrent and turn-based settings, proving its PTIME-completeness. To this aim we also exploit also a characterization of the maximum grading value of a given formula.

1 Introduction

Graded modalities are logical operators allowing to express quantitative bounds on the set of individuals satisfying a certain property [11]. They are well-known in the knowledge representation field, as well as in classical logic [12] and in description logics [13]. Such modalities have received renewed attention by the theoretical computer science community, especially in the formal verification field: in [15, 7] they are applied to the μ-calculus logic, while in [8, 10, 4] to CTL. Here, we add graded modalities to ATL, as a step from closed to open systems, and provide efficient model-checking algorithms for the resulting logic. At the best of our knowledge, this is the first time that such notions are applied in the game-theoretic setting.

ATL was introduced by Alur et al. [2] as a derivative of CTL that is interpreted on *games*, rather than transition systems. Since its inception, ATL has been quickly adopted in different areas of computer science dealing with multi-agent systems, and it has provided the basis for further extensions [1, 20, 5].

The temporal part of ATL coincides with the one of CTL, while the path quantifiers of CTL are replaced by *team* quantifiers, that quantify over the strategies of a given team. For instance, for a suitable subformula θ, the ATL formula $\langle\langle 1\rangle\rangle\theta$ expresses the fact that the team composed of Player 1 alone can ensure that θ holds. More in detail, said formula hides two classical quantifiers: there exists a strategy of Player 1, such that, whatever the other players do, θ holds in the resulting outcomes. (Standard CTL path quantifiers can be obtained as special cases of ATL quantifiers.

In this paper, we enrich the ATL quantifiers with an integral *grade*, and we interpret the resulting formulas using three alternative semantics. First, we consider a very natural extension of the semantics of ATL formulas: for a natural number k, the graded ATL formula $\langle\langle X\rangle\rangle^k\theta$ affirms that the players belonging to the team X have k *different* strategies to enforce θ, that is, the team has k different ways of winning, each satisfying θ, whatever the remaining players do. Intuitively, two distinct runs of the play are counted as different if they present a difference in the choice of the moves leading to satisfy the winning condition. We call this semantics *off-line*, as it seems suitable to off-line synthesis applications. In this context, a two-player game is a model of a control system, and the two players represent the controller and its environment, respectively. Verifying the property $\langle\langle 1\rangle\rangle^k\theta$, and possibly computing k witnessing strategies for Player 1, corresponds to synthesizing k different controllers, that may later (i.e., off-line) be compared w.r.t. some external criterion.

However, as shown in the following, some cases may exhibit infinitely many winning strategies, calling for a refined counting notion. We therefore introduce the *memoryless* semantics, that only counts the number of different memoryless winning strategies, i.e., strategies whose choices only depend on the current state in the game. This restriction makes perfect sense in the controller synthesis scenario, where memoryless controllers are highly desirable for their simplicity.

Then, we turn to the application of automatic verification of fault-tolerant controllers for open systems. In this case, we do not wish to restrict the moves of a player (i.e., synthesize a controller), but rather we assume that the controller may take any of the (redundant) actions that are present in the game, and we want to evaluate how many faults the controller can tolerate at most before violating its specification, where a fault is represented by the absence at run-time of a move that is present in the model. To this purpose, we introduce a new semantics to count the number of different winning paths that a team can follow, in the worst possible case w.r.t. the choices of the opposite team. We call this semantics "on-line" because it is related to the ability of the player to dynamically alter its behavior to overcome faults. In this sense, the graded ATL formula $\langle\langle X\rangle\rangle^k\theta$, interpreted in the on-line semantics, becomes a necessary condition to guarantee that team X can force θ even in the presence of $k-1$ faults.

To prove our results, we first consider the case of turn-based games, where states are partitioned among the players and at each state the player who owns it moves along one of the outgoing edges. We compare the semantics and prove that the on-line satisfaction of a negation-free graded ATL formula implies its off-line satisfaction, while the vice versa is not true. The on-line and the memoryless semantics turn out to be incomparable.

For all semantics we then solve the model-checking problem, computing the truth values of graded ATL formulas on the states of a given game. We provide algorithms that are executed in polynomial time w.r.t. the size of the input game and the number of logical operators in the input formula. A matching lower bound shows that these problems are in fact PTIME-complete. For the off-line and the on-line semantics, the time complexity does not even depend on the constants occurring in the graded team quantifiers. In particular, for the off-line semantics we retain the same complexity as in ATL.

Given an ATL formula, an extended form of model checking is to determine the value of the maximum grade for which that formula is true on a given state of a game. For the off-line and the on-line interpretations, we provide a fixpoint characterization for that value. A fixpoint characterization also suggests the simple Picard iteration method for computing such value. However, in our case, two issues prevent Picard iteration from being applied effectively. First, the maximum grade of a formula can be infinity. Second, even if grade infinity was to be treated separately, Picard iteration would still require a number of iterations proportional to the integer value being computed. For these reasons, the algorithms we give are ad-hoc, and compute the maximum grade of a formula avoiding the above-mentioned issues, while still exploiting the fixpoint characterization.

Finally, we consider concurrent game structures and show that the model-checking problem is PTIME-complete and can be solved with the same complexity as for turn based games.

ATL has been implemented in several tools for the analysis of open systems [3, 17], and graded modalities for CTL have been integrated in the NuSMV tool [9, 6]. We plan to extend this in the near future to consider graded ATL specifications.

The rest of the paper is organized as follows. Section 2 shows an example. Section 3 presents the basic definitions, including the three alternative semantics for graded ATL. Section 4 presents a fixpoint characterization of the off-line and on-line semantics. Section 5 performs a comparison between the semantics. Section 6 describes the model-checking algorithms, computing the truth values of graded ATL formulas on the states of a given game. In Section 7 we deal with concurrent games.

2 A Motivating Example

We give now an example, before introducing our logic. Consider the game in Figure 1, representing the steps required to open an attachment sent by an external agent in one of three possible file formats. The game is played by two players with a turn-based modality: each state belongs only to one of the players and at each turn the player who owns the current state chooses one of its outgoing edges. In the figure, states of Player 1 are circles and those of Player 2 are squares. In the initial state s_0, Player 2 picks a file type, among PostScript, Adobe PDF, and Microsoft Word. Then, Player 1 tries to open the file (i.e., reach state s_4) by using appropriate programs. The ATL formula $\langle\langle 1 \rangle\rangle \Diamond s_4$, meaning "Player 1 has a strategy to reach state s_4", is true at s_0, because, no matter what file type Player 2 chooses, Player 1 has a way (sometimes more than one) to reach s_4. Graded ATL provide the means to *count* how many different ways to win Player 1 has and this, clearly, cannot be achieved with classical ATL.

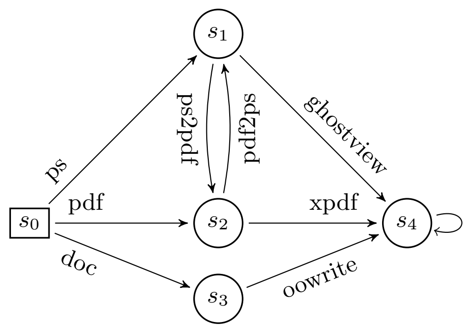

Fig. 1: An attachment-opening game.

Assume first that the aim for analyzing the game in Figure 1 is to automatically generate as many scripts as possible, each one of them able to open all types of attachment (i.e., win the game). If we apply the off-line semantics, it turns out that Player 1 has infinitely many winning strategies: When receiving a ps or pdf file, she can choose to run the converting programs ps2pdf and pdf2ps as many times as she likes, going back and forth between the two formats, before opening the file and reaching the target state s_4. In graded ATL terms, in the off-line semantics the formula $\langle\langle 1 \rangle\rangle^k \Diamond s_4$ holds at s_0 for all $k \geq 1$. Clearly, we are not interested in synthesizing infinitely many scripts that only differ in the amount of useless work they perform. Hence, we introduce the memoryless semantics, that only counts the number of memoryless winning strategies. In the current example, there are three memoryless winning strategies: the one that uses no converters, the one that uses only ps2pdf and the one that uses only pdf2ps. Using both converting programs leads to an infinite loop that does not reach state s_4. Formally, in the memoryless semantics the formula $\langle\langle 1 \rangle\rangle^k \Diamond s_4$ holds at s_0 for all $k \leq 3$. This suggests that we can synthesize three substantially different scripts for our problem.

On the other hand, assume that we want to know the degree of fault-tolerance of our configuration in the worst case (w.r.t. the choices of Player 2), where a fault is represented by the malfunction of one of the available programs. The on-line semantics tells us that if Player 2 inadvertently chooses the doc file format, Player 1 can only reach s_4 in one way. In graded ATL terms, $k = 1$ is the maximum integer such that $\langle\langle 1 \rangle\rangle^k \Diamond s_4$ holds at s_0 in the on-line semantics. Thus, the example under consideration shows no fault-tolerance in our sense, since a single fault (i.e., the absence of the "oowrite" program) can prevent Player 1 from opening the attachment.

3 Preliminaries

We consider games played by m players on a finite graph, whose set of states is partitioned in m subsets, each one corresponding to one of the players. The game starts in a state of the graph, and at each step the player who owns the current state chooses one of its outgoing edges. As a consequence, the game *moves* to the destination of that edge. The game continues in this fashion, until an infinite path is formed. Such games are called *turn-based*, as opposed to *concurrent*, since at each step only one player is responsible for the next move. Throughout the paper, we consider a fixed set Σ of *atomic propositions*. The following definitions make this framework formal.

Turn based games. A *Turn Based Game* (in the following, simply *game*) is a tuple $G = (m, S, pl, \delta, [\cdot])$ such that: $m > 0$ is the number of players; S is a finite set of states; $pl : S \to \{1, \dots, m\}$ is a function mapping each state s to the player who owns it; $\delta \subseteq S \times S$ is the *transition relation* which provides the moves of the players and $[\cdot] : S \to 2^{\Sigma}$ is the function assigning to each state s the set of atomic propositions that are true at s. In the following, unless otherwise noted, we consider a fixed game $G = (m, S, pl, \delta, [\cdot])$. We assume that games are non-blocking, i.e., each state has at least one successor in δ, to which it can move. The players can join to form a *team* which is a subset of $\{1, \dots, m\}$. For a team $X \subseteq \{1, \dots, m\}$, we denote by S_X the set of states belonging to team X, i.e. $S_X = \{s \in S \mid pl(s) \in X\}$, and we denote by $\neg X$ the opposite team, i.e., $\neg X = \{1, \dots, m\} \setminus X$. A (finite or infinite) path in G is a (finite or infinite) path in the directed graph (S, δ). Given a path ρ, we denote by $\rho(i)$ its i-th state, by $\mathit{first}(\rho)$ its first state, and by $\mathit{last}(\rho)$ its last state, when ρ is finite.

Strategies. A *strategy* in G is a pair (X, f), where $X \subseteq \{1, \dots, m\}$ is the *team* to which the strategy belongs, and $f : S^+ \to S$ is a function such that for all $\rho \in S^+$, $(\mathit{last}(\rho), f(\rho)) \in \delta$. Our strategies are deterministic, or, in game-theoretic terms, *pure*. A strategy $\sigma = (X, f)$ is *memoryless* if $f(\rho)$ only depends on the last state of ρ, that is, for all $\rho, \rho' \in S^+$, if $\mathit{last}(\rho) = \mathit{last}(\rho')$ then $f(\rho) = f(\rho')$. We say that an infinite path $s_0 s_1 \dots$ in G is *consistent* with a strategy $\sigma = (X, f)$ if, for all $i \geq 0$, if $s_i \in S_X$ then $s_{i+1} = f(s_0 s_1 \dots s_i)$. Observe that the infinite paths in G which start from a state and are consistent with a given strategy of a team X form a tree where the states of team X only have one child. We denote by $\mathrm{Outc}_G(s, \sigma)$ the set of all infinite paths in G which start from s and are consistent with σ (in the following, we omit the subscript G when it is obvious from the context). For two strategies $\sigma = (X, f)$ and $\tau = (\neg X, g)$, and a state s, we denote by $\mathrm{Outc}(s, \sigma, \tau)$ the unique infinite path which starts from s and is consistent with both σ and τ.

3.1 Graded ATL: Definitions

In this subsection we give the definition of graded ATL. We extend ATL, defined in [2], by adding grading capabilities to the team quantifiers.

Syntax. Consider the *path formulas* θ and *state formulas* ψ defined via the inductive clauses below.

$$\theta ::= \bigcirc\psi \mid \psi\, U \psi \mid \Box\psi;$$
$$\psi ::= q \mid \neg\psi \mid \psi \vee \psi \mid \langle\langle X \rangle\rangle^k \theta,$$

where $q \in \Sigma$ is an atomic proposition, $X \subseteq \{1, \ldots, m\}$ is a team, and k is a natural number. Graded ATL is the set of all state formulas.

The operators U (until), $\Box$ (globally) and $\bigcirc$ (next) are the temporal operators. As usual, also the operator $\Diamond$ (eventually) can be introduced using the equivalence $\Diamond\psi \equiv \mathit{true}\, U \psi$. The syntax of ATL is the same as the one of graded ATL, except that the team quantifier $\langle\langle\cdot\rangle\rangle$ exhibits no natural superscript.

Semantics. We present three alternative semantics for graded ATL, called *off-line semantics*, *memoryless semantics* and *on-line semantics* for reasons explained in the Introduction. Their satisfaction relations are denoted by $\models^{\text{off}}$, $\models^{\text{mless}}$ and $\models^{\text{on}}$, respectively, and they only differ in the interpretation of the team quantifier $\langle\langle . \rangle\rangle$. We start with the operators whose meaning is invariant in all semantics. Let ρ be an infinite path in the game, s be a state, and ψ_1, ψ_2 be state formulas. For $x \in \{\text{on}, \text{off}, \text{mless}\}$, the satisfaction relations are defined as follows.

$$\rho \models^x \bigcirc\psi_1 \quad \text{iff } \rho(1) \models^x \psi_1$$
$$\rho \models^x \Box\psi_1 \quad \text{iff } \forall i \in \mathbb{N} \,.\, \rho(i) \models^x \psi_1$$
$$\rho \models^x \psi_1\, U \psi_2 \quad \text{iff } \exists j \in \mathbb{N} \,.\, \rho(j) \models^x \psi_2 \text{ and } \forall 0 \le i < j \,.\, \rho(i) \models^x \psi_1 \qquad (\dagger)$$

$$s \models^x q \quad \text{iff } q \in [s]$$
$$s \models^x \neg\psi_1 \quad \text{iff } s \not\models^x \psi_1$$
$$s \models^x \psi_1 \vee \psi_2 \quad \text{iff } s \models^x \psi_1 \text{ or } s \models^x \psi_2.$$

As explained in the following, graded ATL formulas have the ability to count how many different paths (in the on-line semantics) or strategies (in the off-line semantics) satisfy a certain property. However, it is not obvious when two paths should be considered "different". For instance, consider the formula $p\, U q$, for some atomic propositions p and q, and two infinite paths that start in the same state s, where s satisfies q and not p. Both paths satisfy $p\, U q$, but only due to their initial state (i.e., $j = 0$ is the only witness for the definition ($\dagger$)). Thus, we claim that these two paths should not be counted as two different ways to satisfy $p\, U q$, because they only become different *after* they have satisfied $p\, U q$. The notion of dissimilar (sets of) paths captures this intuition.

We say that two finite paths ρ and ρ' are *dissimilar* iff there exists $0 \le i \le \min\{|\rho|, |\rho'|\}$ such that $\rho(i) \neq \rho'(i)$. Observe that if ρ is a prefix of ρ', then ρ and ρ' are not dissimilar. For a path ρ and an integer i, we denote by $\rho_{\le i}$ the prefix of ρ comprising $i+1$ states, i.e. $\rho_{\le i} = \rho(0), \rho(1), \ldots, \rho(i)$. Given $\varphi = \langle\langle X \rangle\rangle\theta$, for a path formula θ and a team $X \subseteq \{1, \ldots, m\}$, and $x \in \{\text{on}, \text{off}, \text{mless}\}$, we say that two infinite paths ρ and ρ' are (φ, x)-*dissimilar* iff:

- $\theta = \bigcirc\psi$ and $\rho(1) \neq \rho'(1)$, or

– $\theta = \Box\psi$ and $\rho(i) \neq \rho'(i)$ for some i, or
– $\theta = \psi_1 \, U \psi_2$ and there are two integers j and j' such that:
  - $\rho(j) \models^x \psi_2$,
  - $\rho'(j') \models^x \psi_2$,
  - for all $0 \leq i < j$, $\rho(i) \models^x \psi_1$ and $\rho(i) \models^x \langle\langle X \rangle\rangle \psi_1 \, U \psi_2$, and
  - for all $0 \leq i' < j'$, $\rho'(i') \models^x \psi_1$, and $\rho'(i') \models^x \langle\langle X \rangle\rangle \psi_1 \, U \psi_2$, and
  - $\rho_{\leq j}$ and $\rho'_{\leq j'}$ are dissimilar.

Finally, two sets of infinite paths are (φ, x)-dissimilar iff one set contains a path which is (φ, x)-dissimilar to all the paths in the other set, and, given a state s, two strategies σ_1, σ_2 are (φ, x)-*dissimilar at* s if the sets $\mathrm{Outc}(s, \sigma_1)$ and $\mathrm{Outc}(s, \sigma_2)$ are (φ, x)-dissimilar.

Off-line semantics. The meaning of the team quantifier is defined as follows, for a state s and a path formula θ.

$s \models^{\mathrm{off}} \langle\langle X \rangle\rangle^k \theta$ iff there exist k strategies $\sigma_1 = (X, f_1), \ldots, \sigma_k = (X, f_k)$ s.t. for all i, j such that $i \neq j$, σ_i and σ_j are $(\langle\langle X \rangle\rangle \theta, \mathrm{off})$-dissimilar at s and for all $\rho \in \mathrm{Outc}(s, \sigma_i)$, we have $\rho \models^{\mathrm{off}} \theta$.

Memoryless semantics. In the memoryless semantics, the meaning of the team quantifier is defined as follows, for a state s and a path formula θ.

$s \models^{\mathrm{mless}} \langle\langle X \rangle\rangle^k \theta$ iff there exist k memoryless strategies $\sigma_1 = (X, f_1), \ldots, \sigma_k = (X, f_k)$ s.t. for all i, j such that $i \neq j$, σ_i and σ_j are $(\langle\langle X \rangle\rangle \theta, \mathrm{off})$-dissimilar at s and for all $\rho \in \mathrm{Outc}(s, \sigma_i)$, we have $\rho \models^{\mathrm{mless}} \theta$.

On-line semantics. The meaning of the team quantifier is defined as follows, for a state s and a path formula θ.

$s \models^{\mathrm{on}} \langle\langle X \rangle\rangle^k \theta$ iff for all strategies $\tau = (\neg X, f)$ there exist k pairwise $(\langle\langle X \rangle\rangle \theta, \mathrm{on})$-dissimilar paths $\rho \in \mathrm{Outc}(s, \tau)$ s.t. $\rho \models^{\mathrm{on}} \theta$.

In the following we omit the superscript k of a team quantifier when $k = 1$. If φ is a classical ATL formula, we simply say in the following that a state s *satisfies* φ or, equivalently, we say that φ holds in s. Moreover, we denote by $[\![\varphi]\!] = \{s \in S \mid s \models \varphi\}$ the set of states that satisfy φ. A *simple* formula has the form $\langle\langle X \rangle\rangle \theta$, for $\theta = \Box\psi$ or $\theta = \psi_1 \, U \psi_2$. For a simple formula $\varphi = \langle\langle X \rangle\rangle \theta$, we denote by δ_φ the restriction of the transition function δ to $[\![\varphi]\!] \times [\![\varphi]\!]$ such that if $(s, s') \in \delta_\varphi$ and $\theta = \psi_1 \, U \psi_2$ then ψ_1 holds in s. For a simple formula $\varphi = \langle\langle X \rangle\rangle \theta$, a tag $x \in \{\mathrm{on}, \mathrm{off}, \mathrm{mless}\}$, and a state s, we set $\mathit{grade}^x(s, \varphi)$ to be the greatest integer k such that $s \models^x \langle\langle X \rangle\rangle^k \theta$ holds. In particular, we set $\mathit{grade}^x(s, \varphi) = 0$ if $s \not\models^x \varphi$ and $\mathit{grade}^x(s, \varphi) = \infty$ if $s \models^x \langle\langle X \rangle\rangle^k \theta$ for all $k \geq 0$. Finally, we set $\hat{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$, where $\mathbb{N}$ is the set of non-negative integers.

4 Fixpoint Characterization

In this section we provide a fixpoint characterization of the functions $grade^x$, for $x \in \{\text{on}, \text{off}\}$. We start with the off-line semantics. Define the following operator $F_\varphi^{\text{off}} : (\llbracket\varphi\rrbracket \to \hat{\mathbb{N}}) \to (\llbracket\varphi\rrbracket \to \hat{\mathbb{N}})$.

$$F_\varphi^{\text{off}}(f)(s) = 1 \sqcup \begin{cases} \sum_{(s,s') \in \delta_\varphi} f(s') & \text{if } s \in S_X \\ \prod_{(s,s') \in \delta_\varphi} f(s') & \text{otherwise,} \end{cases} \tag{1}$$

where $x \sqcup y$ denotes $\max\{x, y\}$. The following result motivates the introduction of the F_φ^{off} operator. Observe that $grade^{\text{off}}(s, \varphi) = 0$ (and also $grade^{\text{on}}(s, \varphi) = 0$) for all $s \in S \setminus \llbracket\varphi\rrbracket$.

Lemma 1. *Let φ be a simple formula and $f : \llbracket\varphi\rrbracket \to \hat{\mathbb{N}}$ be such that $f(s) = grade^{\text{off}}(s, \varphi)$, for $s \in \llbracket\varphi\rrbracket$. The function f is the least fixpoint of F_φ^{off}.*

Proof. First, we prove that f is a fixpoint of F_φ^{off}. Let $\varphi = \langle\langle X\rangle\rangle\theta$ and $s \in \llbracket\varphi\rrbracket$, and, for all successors s_i of s such that $(s, s_i) \in \delta_\varphi$, let us set $k_i = grade^{\text{off}}(s_i, \varphi)$. That is, k_i strategies of team X exist which determine k_i (φ, off)-dissimilar sets of paths, consistent with the strategies and satisfying θ. If $s \in S_X$, then the total number of winning strategies for X from s is the sum of the k_i's. Indeed, each winning strategy starting from s_i remains winning if started from s (if $\theta = \psi_1 \, U \psi_2$, it is essential the hypothesis that $s \models \psi_1$ ensured by the definition of δ_φ). If $s \notin S_X$, then for each s_i, the players of the team X can choose one of the k_i dissimilar winning strategies. Each combination gives rise to a winning strategy from s, that is dissimilar to the one obtained by any other combination. Therefore, the total number of dissimilar winning strategies from s is the product of the k_i's.

Next, we prove that f is the *least* fixpoint of F_φ^{off}. Precisely, we prove by induction on n the following statement: Let g be a fixpoint of F_φ^{off} and let $s \in \llbracket\varphi\rrbracket$, if $g(s) \leq n$ then $f(s) \leq g(s)$. Assume that $\theta = \Box\psi$ (the other case is similar). If $n = 1$, by hypothesis $g(s) = 1$. Considering the definition of F_φ^{off}, there are the following three possibilities: *(i)* s has no successors according to δ_φ; *(ii)* s belongs to S_X and has only one successor in δ_φ; *(iii)* s does not belong to S_X and $g(t) = 1$, for all states t such that $(s, t) \in \delta_\varphi$. Option *(i)* can be discarded because $\llbracket\varphi\rrbracket$ is the set of states where $\langle\langle X\rangle\rangle\Box\psi$ holds, and thus each state in $\llbracket\varphi\rrbracket$ has at least one successor in δ_φ. Given the remaining two options, one can see that $\neg X$ can force the game in a loop where all states x have value $g(x) = 1$, and the players of X cannot exit this loop. Accordingly, we have $f(s) = 1$, as requested. If $n > 1$, by contradiction, let g be a fixpoint of F_φ^{off} which is smaller than f. I.e., there is a state $s \in \llbracket\varphi\rrbracket$ such that $g(s) < f(s)$. Clearly, it must be $f(s) > 1$. Assume w.l.o.g. that also $g(s) > 1$, otherwise proceed as in the case for $n = 1$. Starting from s, build a path in the game in the following way. Let t be the current last state of the path (at the beginning, $t = s$): if t has only one successor u according to δ_φ, pick u as the next state (notice that $g(u) = g(t)$ and $f(u) = f(t)$); if $t \notin S_X$ and t has more than one successor according to δ_φ, pick as the next state of the path a successor u such that $g(u) < f(u)$ (it is a simple matter of algebra to show that such a state exists); finally, if $t \in S_X$ and t has more than one successor according to δ_φ, stop. If the above process continues forever, it means that the adversaries (players not in X) can force the game in a loop from which players in X cannot exit. This means that $f(s) = 1$, which is a contradiction. Otherwise, the above process stops in a state $t \in S_X$, such that $g(t) \le g(s)$ and $g(t) < f(t)$. Since t has more than one successor, by (1), for all successors u of t we have $g(u) < g(t) \le g(s) \le n$ and thus $g(u) \le n-1$. Moreover, there is a successor u^* of t such that $g(u^*) < f(u^*)$. On the other hand, by inductive hypothesis $g(u^*) \ge f(u^*)$, which is a contradiction.

Now, we provide a similar characterization for the on-line semantics. Define the following operator $F^{\text{on}}_\varphi : (\llbracket\varphi\rrbracket \to \hat{\mathbb{N}}) \to (\llbracket\varphi\rrbracket \to \hat{\mathbb{N}})$.

$$F^{\text{on}}_\varphi(f)(s) = 1 \sqcup \begin{cases} \sum_{(s,s')\in\delta_\varphi} f(s') & \text{if } s \in S_X \\ \min_{(s,s')\in\delta_\varphi} f(s') & \text{otherwise.} \end{cases} \tag{2}$$

Lemma 2. *Let φ be a simple formula and $f : \llbracket\varphi\rrbracket \to \hat{\mathbb{N}}$ be such that $f(s) = grade^{\text{on}}(s,\varphi)$, for $s \in \llbracket\varphi\rrbracket$. The function f is the least fixpoint of F^{on}_φ.*

Proof. First, we prove that f is a fixpoint of F^{on}_φ. Let $\varphi = \langle\langle X\rangle\rangle\theta$ and $s \in \llbracket\varphi\rrbracket$. Suppose that there is at least one successor of s in δ_φ (otherwise $\theta = \psi_1 U \psi_2$, $s \models^{\text{on}} \psi_2$, and $F^{\text{on}}_\varphi(f)(s) = f(s) = 1$). For all successors s_i of s in δ_φ, let $k_i = grade^{\text{on}}(s_i, \langle\langle X\rangle\rangle\theta)$. For each strategy τ of $\neg X$, there are k_i dissimilar paths starting from s_i, consistent with τ, and satisfying θ. Therefore, if $s \in S_X$, by adding state s in front of each of these paths, we obtain $\sum_i k_i$ dissimilar paths starting from s, consistent with τ, and satisfying θ. In fact, if $\theta = \Box\psi$, since $s \models^{\text{on}} \langle\langle X\rangle\rangle\theta$, we have that $s \models^{\text{on}} \psi$, while, if $\theta = \psi_1 U \psi_2$, then $s \models^{\text{on}} \psi_1$ (if it is not the case, there are no successors s_i of s such that $(s, s_i) \in \delta_\varphi$). If instead $s \notin S_X$, let $i = \arg(\min_j k_j)$. Consider the memoryless strategy τ of $\neg X$ that picks s_i when the game is in s. Under τ, there are k_i dissimilar paths starting from s and satisfying θ. From the choice of i, it follows that all strategies of $\neg X$ have at least as many dissimilar paths from s.

Next, we prove that f is the *least* fixpoint of F^{on}_φ. Similarly to the proof of Lemma 1, we prove by induction on n the following statement: Let g be a fixpoint of F^{on}_φ and let $s \in \llbracket\varphi\rrbracket$, if $g(s) \le n$ then $f(s) \le g(s)$. Assume for simplicity that $\theta = \Box\psi$, as the other case can be proved along similar lines. The case for $n = 1$ can be proved similarly to the proof of Lemma 1. If $n > 1$, by contradiction, let g be a fixpoint of F^{on}_φ which is smaller than f. I.e., there is a state $s \in \llbracket\varphi\rrbracket$ such that $g(s) < f(s)$. Clearly, it must be $f(s) > 1$. Starting from s, build a path in the game in the following way. Let t be the current last state of the path (at the beginning, $t = s$): if $t \notin S_X$, pick as the next state of the path a successor u of t such that $g(u) = g(t)$ and $(t,u) \in \delta_\varphi$ (notice that $f(u) \ge f(t)$); if $t \in S_X$ and t has only one successor u with $(t,u) \in \delta_\varphi$, pick u as the next state (notice that $g(u) = g(t)$ and $f(u) = f(t)$); finally, if $t \in S_X$ and t has more than one such successor, stop. If the above process continues forever, team $\neg X$ can force the game in a loop from which team X cannot exit. This means that $f(s) = 1$, which is a contradiction. Otherwise, the above process stops in a state $t \in S_1$, such that $g(t) = g(s)$ and $f(t) \geq f(s)$. Therefore, $f(t) > g(t)$. Since t has more than one successor according to δ_φ, by (2), for all successors u of t we have $g(u) < g(t) = g(s) \leq n$ and thus $g(u) \leq n - 1$. Moreover, there is a successor u^* of t such that $g(u^*) < f(u^*)$. On the other hand, by inductive hypothesis $g(u^*) \geq f(u^*)$, which is a contradiction.

5 Comparing the Semantics

The example in Figure 1 shows that the three semantics are different in general. In this section we clarify the relationship between the semantics. In the following examples we consider two players, Player 1 and Player 2. As before, states of Player 1 are represented by circles and those of Player 2 by squares.

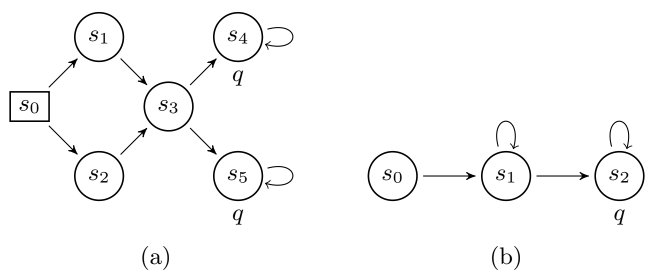

Fig. 2: Two games where the semantics differ.

The first example shows that the memoryless semantics can give a smaller grading value than the offline semantics, even when the latter produces a finite value.

Example 1. Consider the game in Figure 2a, where the goal for Player 1 is to reach the proposition q, which is true in states s_4 and s_5. According to the off-line semantics, there are 4 possible strategies to achieve that goal. Namely, for each choice of Player 2 in s_0, Player 1 has two options once the game is in s_3. Thus, we have $s_0 \models^{\text{off}} \langle\langle 1 \rangle\rangle^4 \Diamond q$ and $s_0 \not\models^{\text{off}} \langle\langle 1 \rangle\rangle^5 \Diamond q$. On the other hand, according to the memoryless semantics, there are only two memoryless strategies for Player 1, the one that leads to s_4 and the one leading to s_5. Thus, $s_0 \models^{\text{mless}} \langle\langle 1 \rangle\rangle^2 \Diamond q$ and $s_0 \not\models^{\text{mless}} \langle\langle 1 \rangle\rangle^3 \Diamond q$.

The example in the introduction shows that the memoryless semantics may attribute to a formula a higher grading value than the on-line semantics. The following example shows that the converse is also possible, hence proving that the two semantics are incomparable.

Example 2. Consider the game in Figure 2b, where the goal for Player 1 is again to reach the proposition q, which is true in s_2. According to the on-line semantics, there are infinitely many strategies to achieve that goal. For all $k > 0$, there is a strategy of Player 1 that makes k visits to s_1 before going to s_2. Thus, we have $s_0 \models^{\text{on}} \langle\langle 1\rangle\rangle^k \Diamond q$, for all $k > 0$. On the other hand, there is only one memoryless winning strategy, i.e., the one that goes directly from s_1 to s_2. Thus, $s_0 \models^{\text{mless}} \langle\langle 1\rangle\rangle^1 \Diamond q$ and $s_0 \not\models^{\text{mless}} \langle\langle 1\rangle\rangle^2 \Diamond q$.

When all quantifiers have grade 1, the semantics coincide. Indeed, the classical quantifiers embedded in each ATL team quantifier (there is a strategy of team X such that for all strategies of team $\neg X$, etc.) can be exchanged, due to the well-known result by Martin on the determinacy of games with Borel objectives [18]. Notice that the languages of infinite words defined by the linear part of ATL are trivially Borel languages. This leads to the following result.

Theorem 1. *For all states s and* ATL *state formulas φ, it holds that*

$$s \models^{\text{off}} \varphi \quad \textit{iff} \quad s \models^{mless} \varphi \quad \textit{iff} \quad s \models^{\text{on}} \varphi.$$

Now we prove that, if a graded ATL formula in which the negation does not occur is satisfied under the on-line semantics, then it is satisfied under the off-line semantics as well. Observe that the same result cannot hold for a general graded ATL formula, since we know that the off-line and the on-line semantics do not coincide. For example, for the game in Figure 1, we have that $s_0 \models^{\text{on}} \neg\langle\langle 1\rangle\rangle^2 \Diamond s_4$, but, on the contrary, it is false that $s_0 \models^{\text{off}} \neg\langle\langle 1\rangle\rangle^2 \Diamond s_4$. Let us consider the F_φ^{off} and F_φ^{on} operators, defined in the previous section, and their iteration, starting from the constant function 1: $F_\varphi^{x,0}(s) = 1$ and $F_\varphi^{x,i+1}(s) = F_\varphi^x(F_\varphi^{x,i})(s)$, for $x \in \{\text{on}, \text{off}\}$ and an ATL formula φ. It is easy to see that both the sequences $F_\varphi^{\text{on},i}(s)$ and $F_\varphi^{\text{off},i}(s)$ are nondecreasing, for every state s, and then, by Lemma 1 and Lemma 2, the following proposition follows.

Proposition 1. *Let φ be a simple formula and let $s \in [\![\varphi]\!]$.*

- *The value $grade^x(s, \varphi)$ is the least upper bound of the sequence $\{F_\varphi^{x,i}(s)\}_{i \geq 0}$, for $x \in \{\text{on}, \text{off}\}$.*
- *For every $i \geq 0$, $F_\varphi^{\text{on},i}(s) \leq F_\varphi^{\text{off},i}(s)$.*

Theorem 2. *For all states s and negation-free graded* ATL *formulas ψ,*

$$\textit{if} \quad s \models^{\text{on}} \psi \quad \textit{then} \quad s \models^{\text{off}} \psi.$$

Proof. Let $\psi = \langle\langle X\rangle\rangle^k \theta$, with either $\theta = \Box q$ or $\theta = p\, U q$, $p, q \in \Sigma$, and $\varphi = \langle\langle X\rangle\rangle \theta$. From Proposition 1, $grade^{\text{on}}(s, \varphi) \leq grade^{\text{off}}(s, \varphi)$, otherwise $grade^{\text{on}}(s, \varphi)$ would not be the least upper bound of $\{F_\varphi^{\text{on},i}(s)\}_{i \geq 0}$. Thus $s \models^{\text{on}} \psi$ only if $s \models^{\text{off}} \psi$. To complete the proof of our statement, we proceed by structural induction on a generic negation-free graded ATL formula. The proof is trivial for the atomic propositions and for the disjunction operator. Let ψ be a graded ATL formula for which we inductively suppose that if $r \models^{\text{on}} \psi$ then $r \models^{\text{off}} \psi$, for any state r of G. If $s \models^{\text{on}} \langle\langle X\rangle\rangle^k \bigcirc \psi$, the statement trivially follows. Suppose now that $s \models^{\text{on}} \langle\!\langle X \rangle\!\rangle^k \Box \psi$ and let $\hat{G}$ be a new game obtained from G by adding a new atomic proposition q_ψ, holding true in all the states r such that $r \models^{\text{on}} \psi$. Clearly, $s \models^{\text{on}}_{\hat{G}} \langle\!\langle X \rangle\!\rangle^k \Box q_\psi$ and, as shown above, $s \models^{\text{off}}_{\hat{G}} \langle\!\langle X \rangle\!\rangle^k \Box q_\psi$. This implies that $s \models^{\text{off}} \langle\!\langle X \rangle\!\rangle^k \Box \psi$ as well. The proof for the U operator is similar.

6 Model Checking

The model checking problem for the semantics $x \in \{\text{on}, \text{off}, \text{mless}\}$ takes as input a game G, a state s in G and a graded ATL formula ψ, and asks whether $s \models^x \psi$. In this section, we efficiently solve the model checking problem for all the considered semantics, in polynomial time w.r.t. the size of the input game and the number of logical operators in the input formula. Moreover, for the off-line and the on-line semantics, the time complexities do not depend on the constants occurring in the graded team quantifiers.

We say that a state is a *decision point for X* (or simply a *decision point*, when the team X is clear from the context) if it belongs to a player of team X and it has at least two successors. Moreover, a strongly connected component of a graph is a *sink* if there are no outgoing edges from it.

Off-line semantics. We first consider $\psi = \langle\!\langle X \rangle\!\rangle^k \theta$ with $\theta = \Box q$ or $\theta = p\,U q$, and provide algorithms for solving a stronger form of model checking, that is we compute $\mathit{grade}^x(s, \langle\!\langle X \rangle\!\rangle \theta)$.

Algorithm 1 The algorithm computing $\mathit{grade}^{\text{off}}(\cdot, \varphi)$, given $\varphi = \langle\!\langle X \rangle\!\rangle \theta$, with $\theta = \Box q$ or $\theta = p\,U q$.

---

1. Using standard ATL algorithms, compute the set of states $[\![\varphi]\!]$, and assign 0 to the states in $S \setminus [\![\varphi]\!]$. Then, compute the subgame with state-space $[\![\varphi]\!]$ and transition relation δ_φ.
2. On the sub-game, compute the strongly connected components.
3. Proceed backwards starting from the sink components, according to the following rules:
   (a) Sink components which do not contain decision points are assigned grade 1.
   (b) Sink and non-sink components having more than one state and containing a decision point (of the subgame) are assigned grade ∞.
   (c) Non-sink components having more than one state and do not fall in case 3b are assigned ∞ if they have a successor component with grade greater than 1; otherwise, they are assigned 1.
   (d) Non-sink components containing only one state: if this state belongs to S_X then it is assigned the sum of the grades of the successor components; while if the state does not belong to S_X, then it is assigned the product of the grades of the successor components.

Lemma 3. *For each state s, Algorithm 1 computes $\mathit{grade}^{\text{off}}(s, \varphi)$, for $\varphi = \langle\!\langle X \rangle\!\rangle \theta$, with $\theta = \Box q$ or $\theta = p\,U q$. The algorithm runs in linear time.*

Proof. The algorithm first computes the states satisfying the ATL formula $\langle\langle X\rangle\rangle\theta$, and removes all other states s, for which it indeed holds $grade^{\text{off}}(s, \langle\langle X\rangle\rangle\theta) = 0$. Then, it computes in the new game the strongly connected components (we assume that there exists at least one such component, otherwise the statement trivially holds). Observe that all the states belonging to the same strongly connected component have the same grade. The algorithm then looks for sink components not containing a decision point, assigning the value 1 to them. Components having more than one state and containing a decision point, get the value ∞. Let us prove that this is correct. Let r be a decision point and suppose that r belongs to Player j, and call r_1, r_2 two of its successors. For all $h > 0$ there is a strategy of team X such that Player j, in state r, chooses to visit r_1 h times, before visiting r_2. Clearly, for each $h > 0$, the strategies σ_i, $i \leq h$, determine pairwise $(\langle\langle X\rangle\rangle\theta, \text{off})$-dissimilar $\text{Outc}(r_2, \sigma_i)$ and thus, $r_2 \models^{\text{off}} \langle\langle X\rangle\rangle^k\theta$, for all $k > 0$. The same reasoning holds for non-sink components and, thus, steps 3a and 3b are correct. Consider now a non-sink component C having more than one state and not containing a decision point (step 3c). Edges outgoing from C are moves of players not belonging to X and thus, if the algorithm has assigned 1 to all the successor components of C, there is only one strategy for the team X. Otherwise, suppose that there is a state r in C having a successor r' in another component and that there exist two strategies of X starting from r'. Then, for any way of alternating between these two strategies, whenever the state r' is entered, there is a strategy of X from r, and thus the algorithm correctly assigns grade infinite. The correctness of case 3d follows from Lemma 1. Finally, observe that the algorithm is complete as all cases have been examined and assuming an adjacency list representation for the game, the above algorithm runs in linear time.

To solve the model checking problem for graded ATL we can use Lemma 3 thus, the following theorem holds. The complexity result assumes that each basic operation on integers is performed in constant time.

Theorem 3. *Given a state s and a graded* ATL *formula ψ, the graded model checking problem, $s \models^{\text{off}} \psi$, can be solved in time $\mathcal{O}(|\delta| \cdot |\psi|)$, where $|\psi|$ is the number of operators occurring in ψ.*

Memoryless semantics. For $\theta \in \{\Box q, p\,U q\}$, in order to model-check a graded ATL formula $\langle\langle X\rangle\rangle^k\theta$ on a state s in the memoryless semantics, we call the function count_mless(G, φ, k, s) (Algorithm 2), with $\varphi = \langle\langle X\rangle\rangle\theta$. We have that s satisfies $\langle\langle X\rangle\rangle^k\theta$ if and only if the result of this call is k. To describe Algorithm 2, we need some extra definitions. Given an ATL formula, we say that a state s is *winning* w.r.t. φ if there exists a strategy σ of team X such that for all $\rho \in \text{Outc}(s, \sigma)$, we have $\rho \models \theta$ (i.e., $s \models \varphi$). In that case we also say that σ is *winning from s*. We say that a strategy is *uniformly winning* if it is winning from all winning states. For two states s and u, and a strategy σ, we say that the *distance of u from s according to σ* is the shortest distance between s and u considering only paths consistent with σ.

In general, count_mless$(G, \varphi = \langle\langle X\rangle\rangle\theta, k, s)$ computes the minimum between k and the number of memoryless strategies of team X, (φ, off)-dissimilar at s, that are winning from s. The idea of the algorithm is the following. We start by computing the set of states W where the ATL formula φ holds, using standard ATL algorithms (line 1). If s does not belong to W, it does not satisfy $\langle\langle X \rangle\rangle^k \theta$, for any $k > 0$. If s belongs to W, we analyze the subgame with state-space W and transition relation δ_φ (see Section 3.1). On this subgame, we compute an arbitrary memoryless uniformly winning strategy. After removing the edge $(u, \pi(u))$ on line 8, every strategy π' in the residual game must be distinct from π, because it cannot use that edge. When $\theta = p\,U q$, two distinct infinite paths that satisfy θ need not be dissimilar. It is necessary that the paths become distinct before an occurrence of q. This property is ensured by the subgame only containing winning states and by all the computed strategies being uniformly winning.

We then put back the edge $(u, \pi(u))$ and on line 12 we remove all other edges leaving u. This ensures that the strategies computed in the following iterations are dissimilar from the ones computed so far. This structure is inspired by the algorithms for computing the k shortest simple paths in a graph [21, 16].

The following result establishes an upper bound on the value returned by count_mless, showing that it cannot return a value higher than the number of mutually dissimilar memoryless winning strategies present in the game. Due to space constraints, we omit the proof and refer the reader to the extended version of this paper.

Lemma 4. *All strategies computed on line 4 by Algorithm 2 are mutually* (φ, off)*-dissimilar at* s*.*

The following result privides a lower bound on the value returned by count_mless. Together with Lemma 4, this result implies the correctness of the algorithm.

Lemma 5. *Given a state* s*, if there are* n *memoryless strategies that are mutually* (φ, off)*-dissimilar at* s*, and that are winning from* s*, then* count_mless(G, φ, k, s) *returns at least* n*, for all* $k \geq n$*.*

The following result characterizes the time complexity of Algorithm 2, in terms of calls to the procedures get_winning_set and get_uniformly_winning_strategy.

Lemma 6. *A call to* count_mless(G, φ, k, s) *which returns value* $n > 0$ *makes at most* $1 + n \cdot |S|$ *calls to get_winning_set and at most* n *calls to get_uniformly_winning_strategy.*

Proof. We proceed by induction on n. For $n = 0$, the statement is trivially true, because the value zero can only be returned on line 2, after one call to get_winning_set.

For $n > 0$, if the algorithm returns on line 5, the statement is trivially true. Otherwise, the algorithm enters the "for all" loop after one call to get_winning_set and one call to get_uniformly_winning_strategy. Let n_i be the value returned by the i-th recursive call on line 9. We have that $n = 1 + \sum_i n_i$ and the number of iterations of the loop is at most $|S|$. By inductive hypothesis, the i-th recursive call is responsible for at most $1 + n_i \cdot |S|$ calls to get_winning_set and at most n_i calls to get_uniformly_winning_strategy. Hence, the total number of calls to get_winning_set is

$$1 + \sum_i (1 + n_i \cdot |S|) = 1 + \sum_i 1 + |S| \sum_i n_i \leq 1 + |S| + |S| \cdot (n-1) = 1 + n \cdot |S|.$$

The total number of calls to get_uniformly_winning_strategy is instead $1 + \sum_i n_i = n$, as required.

Considering that Atl model checking can be performed in linear time w.r.t. the adjacency list representation of the game, from Lemma 6 we obtain the following.

Corollary 1. *The time complexity of count_mless*(G, φ, k, s) *is* $\mathcal{O}(k \cdot |S| \cdot (|S| + |\delta|)) = \mathcal{O}(k \cdot |S| \cdot |\delta|)$.

Algorithm 2 The procedure count_mless(G, φ, k, s).

Require: $G = (m, S, pl, \delta, [\cdot])$: game, $\varphi \in \{\langle\langle X \rangle\rangle \Box q, \langle\langle X \rangle\rangle p\, U q\}$, k: natural number, s: state of G

$W :=$ get_winning_set(G, φ)
if $s \notin W$ **then return** 0
$G' := (m, S, pl, \delta_\varphi, [\cdot])$
$\pi :=$ get_uniformly_winning_strategy(G', φ)
if $k = 1$ **then return** 1
$n := 1$
for all decision points u of team X, reachable from s according to π, in non-decreasing order of distance from s according to π **do**
  remove_edge$(G', (u, \pi(u)))$
  $n := n+$ count_mless$(G', \varphi, k - n, s)$
  add_edge$(G', (u, \pi(u)))$
  if $n = k$ **then return** n
  remove_edges$(G', \{(u, x) \mid x \neq \pi(u)\})$
end for
return n

From the previous lemmas and by using standard arguments, we obtain a solution to the model-checking problem for a graded Atl formula, under the memoryless semantics.

Theorem 4. *Given a state* s *and a graded* Atl *formula* ψ*, the graded model checking problem, in the memoryless semantics, can be solved in time* $\mathcal{O}(\hat{k} \cdot |S| \cdot |\delta| \cdot |\psi|)$*, where* $\hat{k}$ *is the maximum value of a constant appearing in* ψ.

On-line semantics. Similarly to the case of off-line semantics, we describe an algorithm for computing $grade^{\mathrm{on}}(s, \langle\langle X \rangle\rangle \theta)$ for $\theta = \Box q$ or $\theta = p\,U q$, and for all states $s \in S$. Given a path formula $\theta = \Box q$ or $\theta = p\,U q$, Algorithm 3 computes $grade^{\mathrm{on}}(s, \langle\langle X \rangle\rangle \theta)$, for all states $s \in S$. The complexity of the algorithm is dominated by step 3, which involves the solution of a Büchi game [19]. This task can be performed in time $\mathcal{O}(|S| \cdot |\delta|)$, i.e., quadratic in the size of the adjacency-list representation of the game.

Algorithm 3 The algorithm computing $grade^{\mathrm{on}}(\cdot, \varphi)$, given $\varphi = \langle\langle X \rangle\rangle \theta$, with $\theta = \Box q$ or $\theta = p\,U q$.

---

1. Using standard ATL algorithms, compute the set of states $[\![\varphi]\!]$. The following steps are performed in the subgame with state-space $[\![\varphi]\!]$ and transition relation δ_φ. Assign grade 0 to the states in $S \setminus [\![\varphi]\!]$.
2. Let d be a new atomic proposition which holds in the decision points (of the subgame). Find the states where $\langle\langle \neg X \rangle\rangle \Box \neg d$ holds, and assign grade 1 to them.
3. Find the states from which team X can enforce infinitely many visits to a decision point (i.e., states where the ATL* formula $\langle\langle X \rangle\rangle \Box \Diamond d$ holds), and assign grade ∞ to them.
4. For the remaining states, compute their value by inductively applying equation (2) to those states whose successors have already been assigned a value.

It is not obvious that the algorithm assigns a value to each state in the game. Indeed, step 4 assigns a value to a state only if all of its successors have already received a value. If, at some point, each state that does not have a value has a successor that in turn does not have a value, the algorithm stops. For the above situation to arise, there must be a loop of states with no value. The following lemma states that the above situation cannot arise, and therefore that the algorithm ultimately assigns a value to each state.

Lemma 7. *At the end of step 3 of Algorithm 3, there is no loop of states with no value.*

We can now state the correctness and complexity of Algorithm 3.

Lemma 8. *Given a path formula $\theta = \Box q$ or $\theta = p\,U q$, at the end of Algorithm 3, each state s has value $grade^{\mathrm{on}}(s, \langle\langle X \rangle\rangle \theta)$. The algorithm runs in quadratic time.*

Proof. We proceed by examining the four steps of the algorithm. If s receives its value (zero) during step 1, it means that $s \not\models^{\mathrm{on}} \langle\langle X \rangle\rangle \theta$. Therefore, zero is the largest integer k such that $s \models^{\mathrm{on}} \langle\langle X \rangle\rangle^k \theta$ holds.

If s receives its value (one) during step 2, it means that $s \models^{\mathrm{on}} \langle\langle \neg X \rangle\rangle \Box \neg d$. Consider a strategy of team $\neg X$ ensuring the truth of $\Box \neg d$. According to this strategy, a player of the team X can never choose between two different successors. Therefore, there is a unique infinite path consistent with this strategy of $\neg X$. This implies that 1 is the greatest integer k such that $s \models^{\mathrm{on}} \langle\langle X \rangle\rangle^k \theta$ holds.

If s receives its value (infinity) during step 3, it means that $s \models^{\text{on}} \langle\langle X \rangle\rangle \Box\Diamond d$. Consider any strategy τ of $\neg X$, and a strategy σ of X ensuring $\Box\Diamond d$. The resulting infinite path ρ contains infinitely many decision points for X. For each decision point $\rho(i)$, let σ_i be a strategy of X with the following properties: *(i)* σ_i coincides with σ until the prefix $\rho_{\leq i}$ is formed, *(ii)* after $\rho_{\leq i}$, σ_i picks a different successor than σ, and then keeps ensuring θ. It is possible to find such a σ_i because $\rho(i)$ is a decision point in the subgame. For all $i \neq j$ such that $\rho(i)$ and $\rho(j)$ are decision points, the outcome of τ and σ_i is dissimilar from the outcome of τ and σ_j. Therefore, $s \models^{\text{on}} \langle\langle X \rangle\rangle^k \theta$ holds for all $k > 0$.

Finally, if s receives its value during step 4, the correctness of the value is a consequence of Lemma 2. The complexity of the algorithm is discussed previously in this section.

Due to the above complexity result, and the discussion already made for the off-line semantics, we obtain the following conclusion.

Theorem 5. *Given a state s and a graded* ATL *formula ψ, the graded model checking problem, $s \models^{\text{on}} \psi$, can be solved in time $\mathcal{O}(|S| \cdot |\delta| \cdot |\psi|)$, where $|\psi|$ is the number of operators occurring in ψ.*

As before, under the constant-time assumption for basic integer operations, the above complexity is independent of the integer constants appearing in the formula. Finally, from the PTIME hardness of the reachability problem for AND-OR graphs [14], this corollary follows.

Corollary 2. *The graded* ATL *model checking problem is PTIME-complete, under the off-line, memoryless and on-line semantics, with respect to the size of the game.*

7 Model Checking Concurrent Games

In this section we consider graded ATL when interpreted over concurrent games, i.e., games where at each step all players contribute to the choice of the next move. We show that the model-checking problem for concurrent games can be reduced to the turn-based setting.

Concurrent game structures. A *concurrent* game structure is a tuple $G = (m, S, d, \delta, [\cdot])$ where: $m > 0$ is the number of players; S is a finite set of states; for each player $i \in \{1, \ldots, m\}$ and state $s \in S$, $d_i(s) \geq 1$ is an integer representing the number of moves of player i at state s; and $[\cdot] : S \rightarrow 2^{\Sigma}$ is the function assigning to each state s the set of atomic propositions that are true at s. In the following, we use integers from 1 to $d_i(s)$ for the moves of player i in state s. In a state s, the vector $\langle j_1, \ldots, j_m \rangle$ is the *move vector* such that $j_i \leq d_i(s)$. For a state s, we define the *move function* $D(s) = \{1, \ldots, d_1(s)\} \times \ldots \times \{1, \ldots, d_m(s)\}$ that lists all the joint moves available to the players. The transition function assigns to each $s \in S$ and $j \in D(s)$ the state $\delta(s, j)$.

Strategies can be defined naturally as in the case of the turn-based setting. In particular, a strategy of player i assigns a move in the range $1, \ldots, d_i(s)$ to each run ending in state s. As far as the dissimilarity is concerned, we have to decide whether moves are sufficient for distinguishing two strategies (or paths). In the following, we answer the above question in the positive: two strategies (or paths) that only differ in the moves chosen by a player, and not in the sequences of states, are considered different, and hence potentially dissimilar. This corresponds to the assumption that different moves in the game represent different real-world actions, that we are interested in counting. The satisfaction relations $\models^{\text{off}}$ and $\models^{\text{on}}$ are defined accordingly.

Model-checking complexity. We briefly report the construction of a 2-player turn-based game G_X from a concurrent game G played by the team X [2]. Consider a concurrent game structure $G = (m, S, d, \delta, [\cdot])$ and a team X of players in $\{1, \ldots, m\}$. For a state $s \in S$, an X-*move* is a possible combination of moves of the players in X when the game is in state s. We denote by $C(X, s)$ the set of X-moves in s, and by $C(X) = \cup_{s \in S} C(X, s)$ the set of all X-moves. For an X-move c in s, a state s' is said to be a c-*successor* of s if $s' = \delta(s, j)$, where $j = \langle j_1, \ldots, j_m \rangle$ and each j_a, for $a \in X$, is determined by the X-move c.

The 2-player turn-based game structure $G_X = (2, S', pl, \delta', [\cdot]')$ is defined as follows: the set of atomic propositions is augmented with a special proposition *aux*, the set of states is $S' = S \cup C(X)$, that is it contains S and, for each X-move, a new state which is now labeled with *aux* by the labeling function (the states in S have the same label as in G). Player 1 owns the states $s \in S$, while Player 2 owns the new states, and the behaviour is the following: there is an edge from a state $s \in S$ to each state $c \in C(X, s)$, and there is an edge from $c \in C(X, s)$, for some s, to $s' \in S$ if s' is a c-successor of s. Clearly G_X has $\mathcal{O}(|\delta|)$ states and edges. Moreover it is easy to see that for each strategy σ of team X in G there exists a corresponding strategy σ' in G_X such that every path π' in $\text{Outc}_{G_X}(s, \sigma')$ is of the type $\ldots, s_i, a_i, s_{i+1}, a_{i+1}, \ldots$, where the s states are in S, and the a states are in $C(X)$, and π' uniquely corresponds to a path $(\ldots, s_i, s_{i+1}, \ldots) \in \text{Outc}_G(s, \sigma)$. Consider now a path formula $\theta = \Box p$ or $\theta = p\, U q$, for atomic propositions p and q. The considerations above imply that the number of $(\langle\!\langle X \rangle\!\rangle \theta, \text{off})$-dissimilar strategies of X in G is equal to the number of $(\langle\!\langle 1 \rangle\!\rangle \theta, \text{off})$-dissimilar strategies of Player 1 in G_X.

Proposition 2. *Let G be a concurrent game, $s \in S$, $p, q \in \Sigma$, and $X \subseteq \{1, \ldots, m\}$. Then the following hold for all $k > 0$:*

$$s \models^{\text{off}}_G \langle\!\langle X \rangle\!\rangle^k \Box p \quad \textit{iff} \quad s \models^{\text{off}}_{G_X} \langle\!\langle 1 \rangle\!\rangle^k \Box (p \vee \text{aux})$$
$$s \models^{\text{off}}_G \langle\!\langle X \rangle\!\rangle^k p\, U q \quad \textit{iff} \quad s \models^{\text{off}}_{G_X} \langle\!\langle 1 \rangle\!\rangle^k (p \vee \text{aux})\, U q.$$

For the on-line case, we consider the 2-player turn-based game $G_{\neg X}$ in which Player 1 plays the role of the team $\neg X$ of the concurrent game G. Thus the number of $(\langle\!\langle \neg X \rangle\!\rangle \theta, \text{on})$-dissimilar paths in each $\text{Outc}_G(s, \sigma)$, for a strategy σ of team $\neg X$ is equal to the number of $(\langle\!\langle 1 \rangle\!\rangle \theta, \text{on})$-dissimilar paths in $\text{Outc}_{G_{\neg X}}(s, \sigma')$ for the corresponding strategy σ' of player 1 in $G_{\neg X}$.

Proposition 3. *Let G be a concurrent game, $s \in S$, $p, q \in \Sigma$ and $X \subseteq \{1, \ldots, m\}$. Then the following hold for all $k > 0$:*

$$s \models^{\text{on}}_{G} \langle\langle X \rangle\rangle^{k} \Box p \quad \textit{iff} \quad s \models^{\text{on}}_{G_{\neg X}} \langle\langle 2 \rangle\rangle^{k} \Box (p \vee \text{ aux})$$
$$s \models^{\text{on}}_{G} \langle\langle X \rangle\rangle^{k} p\, U q \quad \textit{iff} \quad s \models^{\text{on}}_{G_{\neg X}} \langle\langle 2 \rangle\rangle^{k} (p \vee \text{aux})\, U q.$$

Let us remark that the above propositions allow us to model check a formula ψ with one team quantifier $\langle\langle X \rangle\rangle$ by using the algorithms for the turn-based games given in the previuos sections. By applying standard techniques one can model-check formulas with any number of nested team quantifiers. A *PTIME*-completeness result thus follows from Corollary 2 and the construction of the 2-player turn-based game described above.

Theorem 6. *The model-checking problem for graded* ATL *on concurrent games is PTIME-complete with respect to the size of the game. Moreover, it can be solved in time $\mathcal{O}(|\delta| \cdot |\psi|)$ in the off-line semantics and in time $\mathcal{O}(|\delta|^2 \cdot |\psi|)$ in the on-line semantics, for a concurrent game with transition function δ and for a graded* ATL *formula ψ.*

References

1. T. Ågotnes, V. Goranko, and W. Jamroga. Alternating-time temporal logics with irrevocable strategies. In *Proc. of TARK '07*, pages 15–24. ACM, 2007.
2. R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49:672–713, 2002.
3. R. Alur, T.A. Henzinger, F.Y.C. Mang, S. Qadeer, S.K. Rajamani, and S. Tasiran. Mocha: modularity in model checking. In *Proc. of CAV 98*, volume 1427 of *Lect. Notes in Comp. Sci.*, pages 521–525. Springer-Verlag, 1998.
4. A. Bianco, F. Mogavero, and A. Murano. Graded computation tree logic. In *Proc. of LICS'09*, 2009.
5. T. Brihaye, A. Da Costa Lopes, F. Laroussinie, and N. Markey. Atl with strategy contexts and bounded memory. In *LFCS*, pages 92–106, 2009.
6. A. Ferrante, M. Memoli, M. Napoli, M. Parente, and F. Sorrentino. A NuSMV extension for graded-CTL model checking. In *Proc. of CAV'10*, 2010.
7. A. Ferrante, A. Murano, and M. Parente. Enriched μ-calculi module checking. *Logical Methods in Computer Science*, 4(3), 2008.
8. A. Ferrante, M. Napoli, and M. Parente. CTL model-checking with graded quantifiers. In *Proc. of ATVA '08*, volume 5311 of *Lect. Notes in Comp. Sci.*, pages 18–32, 2008.
9. A. Ferrante, M. Napoli, and M. Parente. Graded-CTL:satisfiability and symbolic model-checking. In *Proc. of ICFEM '09*, volume 5885 of *Lect. Notes in Comp. Sci.*, pages 306–325, 2009.
10. A. Ferrante, M. Napoli, and M. Parente. Model-checking for graded CTL. *Fundamenta Informaticae*, 96(3):323–339, 2009.
11. K. Fine. In so many possible worlds. *Notre Dame journal of Formal Logic*, 13(4):516–520, 1972.
12. E. Gradel, M. Otto, and E. Rosen. Two-variable logic with counting is decidable. In *Proc. of LICS97*, 1997.

13. B. Hollunder and F. Baader. Qualifying number restrictions in concept languages. In *2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 335–346, 1991.
14. N. Immerman. Number of quantifiers is better than number of tape cells. *J. Comput. Syst. Sci.*, 22(3):384–406, 1981.
15. O. Kupferman, U. Sattler, and M.Y. Vardi. The complexity of the graded μ-calculus. In *Proc. 18th Conference on Automated Deduction*, volume 2392 of *Lect. Notes in Comp. Sci.*, 2002.
16. E. L. Lawler. A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem. *Management Science*, 18(7):401–405, 1972.
17. A. Lomuscio and F. Raimondi. Mcmas: A model checker for multi-agent systems. In *Proc. of TACAS 06*, volume 3920 of *Lect. Notes in Comp. Sci.*, pages 450–454. Springer-Verlag, 2006.
18. D.A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975.
19. W. Thomas. On the synthesis of strategies in infinite games. In *Proc. of STACS95*, volume 900 of *Lect. Notes in Comp. Sci.*, pages 1–13. Springer-Verlag, 1995.
20. W. van der Hoek and M. Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.
21. Jin Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.