Fundamenta Informaticae XX (2011) 1–22
IOS Press

Graded Alternating-Time Temporal Logic

Marco Faella
Department of Physics
Università di Napoli "Federico II", Italy

Margherita Napoli
Dipartimento di Informatica ed Applicazioni "R.M. Capocelli"
Università di Salerno, Italy

Mimmo Parente
Dipartimento di Informatica ed Applicazioni "R.M. Capocelli"
Università di Salerno, Italy

Abstract. 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. In both these settings, 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. Both μ-calculus and CTL naturally apply as specification languages for *closed* systems: in this paper, we study how graded modalities may affect specification languages for *open* systems. We extend the Alternating-time Temporal Logic (ATL) introduced by Alur et al., that is a derivative of CTL interpreted on *game* structures, rather than transition systems. We solve the model-checking problem in the concurrent and turn-based settings, proving its PTIME-completeness. We present, and compare with each other, two different semantics: the first seems suitable to off-line synthesis applications while the second may find application in the verification of fault-tolerant controllers. We also study the case where players can only employ memoryless strategies, showing that also in this case the model-checking problem is in PTIME.

1. Introduction

Open systems are systems which interact with the environment and whose behavior is determined by this interaction. Concurrent systems may also be viewed as composed of interacting open components. In both these settings the problem of verifying and synthesizing open systems has become an important and challenging issue. *Games* turn out to be a useful formalism to model open systems. The players of a game, representing the individual components, aim at forcing the game into a specific goal. They move on a finite arena and their moves determine changes on the state of the game.

In [5], Alternating-time Temporal Logic (ATL) was introduced by Alur et al. as a formalism to specify requirements for open systems which involve multiple components. ATL is an extension of CTL interpreted on game structures. The temporal operators of ATL coincide with those of CTL, while the path quantifiers of CTL are replaced by *team quantifiers*. ATL allows coalitions of players, called *team*s, and the team quantifiers range over the strategies of a given team. For instance, the ATL formula $\langle\langle X \rangle\rangle\theta$ expresses that the team X can ensure that the path formula θ holds, i.e. it expresses that there exists a strategy of players belonging to X, such that, whatever the other players do, θ holds in the resulting outcomes. The standard CTL path quantifiers can be obtained as special cases of ATL quantifiers. In a game with set of players $\mathcal{P}$, the ATL formula $\langle\langle \mathcal{P} \rangle\rangle\theta$ is equivalent to the CTL formula $\exists\theta$ since it states that all the players together can cooperate to ensure θ. Equivalently, CTL can be seen as the fragment of ATL with only one player.

Graded modalities, especially well-known in the knowledge representation field, allow to express quantitative bounds on the set of individuals satisfying a certain property [9]. Similar notions are the counting quantifiers in classical logic [12] and the number restrictions in description logics [13]. Recently, graded modalities have received renewed attention in the field of formal verification and have been applied to formalisms used to specify requirements for closed systems, see e.g. [15, 7, 8]. In this paper, we make a progress along these lines by adding graded modalities to ATL, as a step from closed to open systems, and providing efficient model-checking algorithms for the resulting logic. At the best of our knowledge, this is the first time that such notions are applied to the game settings.

Our graded logic. We enrich the ATL quantifiers with an integral *grade*, and we interpret the resulting formulas using two alternative semantics. First, we consider a very natural extension of the semantics of ATL formulas which allows to count the number of winning strategies: for a natural number k, the graded ATL formula $\langle\langle X \rangle\rangle^k\theta$ affirms that the players belonging to a team X have k *different* strategies to enforce θ, that is, the team has k strategies whose outcoming runs are *different* and all satisfy θ, whatever the remaining players do. Intuitively, the notion of different runs used here implies that two distinct runs are counted as two different ways for a team to win if they differ for at least one move leading to satisfy the winning condition. We call this semantics *off-line* since 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. Observe that in this setting we might want to restrict some players to only employ *memoryless* strategies, i.e., strategies whose choices only depend on the current state in the game. Such strategies are indeed the easiest to implement in practice and then a memoryless controller is often desirable. Thus we have considered also the memoryless variation of the off-line semantics, that counts the number of different memoryless strategies of a given team.

The second semantics we introduce allows to count how many different winning paths a team can follow, in the worst possible case w.r.t. the choices of the opposite team. That is, it expresses properties such as "There are k *runs* winning for the players in a team, for each strategy of the opposite team". We call this semantics "on-line" because it is related to the ability of a player to dynamically alter its behavior. We believe that this semantics may find several applications, in particular in the verification of fault-tolerant controllers. 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 before violating its specification, where a fault is represented by the absence at run-time of a move that is present in the model. In this sense, the graded ATL formula $\langle\langle X \rangle\rangle^k \theta$, with this semantics, becomes a necessary condition to guarantee that team X can force θ even in presence of $k-1$ faults.

An example. 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 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. Suppose that we use the name of the states as propositions, 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.

The two semantics of 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.

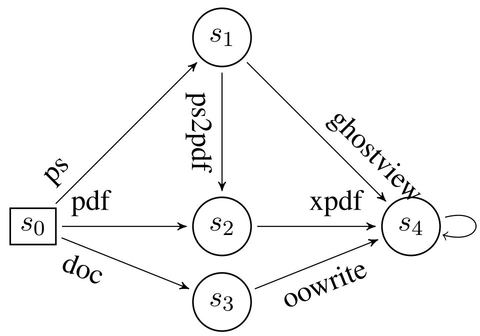

Figure 1: An attachment-opening game.

The off-line semantics counts the number of winning strategies. In the current example, it turns out that Player 1 has two winning strategies: When receiving a ps file, she can choose to immediately open it or to first convert it into a pdf file. Each one of such choices represents a different winning strategy. 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 $k \leq 2$. and, if Player 1 represents the controller, then it can be synthesized in two different ways. In this case the obtained strategies are memoryless. On the contrary, consider again the example in Figure 1, and suppose we add an edge from state s_2 to state s_1, labeled with "pdf2ps". Then, according to the off-line semantics, Player 1 has infinitely many strategies to reach s_4. In this case it could be useful to determine the memoryless strategies that are, indeed, only three: one does not use the converting programs "ps2pdf" and "pdf2ps", while the other two use only one of the converting programs. Using both converting programs leads to an infinite loop that does not reach state s_4.

On the other hand, the on-line semantics counts how many different winning paths Player 1 can follow, in the worst possible case. In the example in Figure 1, 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. In fact, when receiving a pdf or doc file Player 1 has no choice between different behaviors. Returning to the verification of fault-tolerant controllers, as said above, we consider faults represented by the absence at run-time of moves of the model. The example under consideration shows no fault-tolerance, since a single fault (i.e., the absence of the "oowrite" program) can prevent Player 1 from opening the attachment. This is witnessed by the fact that $\langle\langle 1\rangle\rangle^2 \Diamond s_4$ is false, under the on-line semantics interpretation.

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 two semantics and prove that the on-line satisfaction of a negation-free graded ATL formula implies its off-line satisfaction and that the viceversa is not true. For both semantics we then solve in polynomial time the model checking problem, computing the truth values of graded ATL formulas on the states of a given game. Let us remark that the time complexities do not depend on the constants occurring in the graded team quantifiers and that in particular for the off-line semantics we retain the same complexity as in ATL. Given an ATL formula, an interesting computational question is to determine the value of the maximum grade for which that formula is true on a given state of a game. For both 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 in polynomial time, avoiding the above-mentioned issues while still exploiting the fixpoint characterization.

We have then considered, as said above, also the problem of verifying the existence of a given number of memoryless strategies. We have given further examples to compare the semantics and proved that this new model checking problem is again in PTIME. However, in this case the time complexity also depends on the values of the constants occurring in the graded team quantifiers

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 turn based games.

The rest of the paper is organized as follows. Section 2 presents the basic definitions, including the two alternative semantics for graded ATL. Section 3 presents a fixpoint characterization of the two semantics. Section 4 performs a comparison between the two semantics. Section 5 describes the model-checking algorithms, computing the truth values of graded ATL formulas on the states of a given game. In Section 6 we deal with concurrent games and in Section 7 with memoryless strategies. Finally, in Section 8 we give some conclusions including related works and open problems.

2. Definitions

In this section we give the preliminary definitions. We start by defining *turn based* games, played by m players, on a finite graph where each state belongs to one of the players. At each turn, the player who owns the current state chooses one of its outgoing edges and the game *moves* to the destination of that edge. Throughout the paper, we consider a fixed set Σ of *atomic propositions*.

2.1. 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* and $[\cdot] : S \to 2^{\Sigma}$ is the function assigning to each state s the set of atomic propositions that are true at s. We assume that games are non-blocking, i.e. each state has at least one successor in δ. In the following, unless otherwise noted, we consider a fixed game $G = (m, S, pl, \delta, [\cdot])$. 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 *first*(ρ) its first state, and by $last(\rho)$ its last state, when ρ is finite.

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 $\rho \in S^+$, $(last(\rho), f(\rho)) \in \delta$. Our strategies are deterministic, or, in game-theoretic terms, *pure*. 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. $\neg X = \{1, \dots, m\} \setminus X$. 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 $s_{i+1} = f(s_0 s_1 \dots s_i)$. 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 (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.

2.2. Graded ATL: Syntax and Semantics

Syntax. Consider the *path formulas* θ and *state formulas* ψ defined via the inductive clauses below. Graded ATL is the set of the state formulas generated by the rules:

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

where $q \in \Sigma$ is an atomic proposition, $X \subseteq \{1, \dots, m\}$ is a team, and k is a natural number. The operators $\mathcal{U}$ (until), $\Box$ (globally) and $\bigcirc$ (next) are the temporal operators. As usual, the operator $\Diamond$ (eventually) can be introduced using the equivalence $\Diamond\psi \equiv true\,\mathcal{U}\psi$. 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 two alternative semantics for graded ATL, called *off-line semantics* and *on-line semantics* for reasons explained in the Introduction. Their satisfaction relations are denoted by $\models^{\mathrm{off}}$ and $\models^{\mathrm{on}}$, respectively, and they only differ in the interpretation of the team quantifier $\langle\!\langle \cdot \rangle\!\rangle$.

We start with the operators whose meaning is invariant in the two semantics. Let ρ be an infinite path in the game, s be a state, and ψ_1, ψ_2 be state formulas. Denote by $\mathbb{N}$ the set of non-negative integers. For $x \in \{\mathrm{on}, \mathrm{off}\}$, the satisfaction relations are defined as follows.

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

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

In order to state the meaning of the team quantifier, we need to introduce the following definitions. We say that two finite paths ρ and ρ' are *dissimilar* iff there exists $0 \leq i \leq \min\{|\rho|, |\rho'|\}$ 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 $\rho_{\leq i}$ the prefix of ρ comprising $i+1$ states, i.e. $\rho_{\leq i} = \rho(0), \rho(1), \ldots, \rho(i)$.

Given $\varphi = \langle\langle X \rangle\rangle \theta$, for a path formula θ and a team $X \subseteq \{1, \ldots, m\}$, and $x \in \{\text{on}, \text{off}\}$, 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
- $\theta = \psi_1 \mathcal{U} \psi_2$ 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 $\rho(i) \models^x \langle\langle X \rangle\rangle \psi_1 \mathcal{U} \psi_2$, and
  - for all $0 \leq i' < j'$, $\rho'(i') \models^x \psi_1$, and $\rho'(i') \models^x \langle\langle X \rangle\rangle \psi_1 \mathcal{U} \psi_2$, 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 $\text{Outc}(s, \sigma_1)$ and $\text{Outc}(s, \sigma_2)$ are (φ, x)-dissimilar

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 $p\mathcal{U}q$, 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 $p\mathcal{U}q$, but only due to their initial state (i.e., $j = 0$ is the only witness to definition (†)). Thus, we claim that these two paths should not be counted as two different ways to satisfy $p\mathcal{U}q$, because they only become different *after* they have satisfied $p\mathcal{U}q$. The notion of dissimilar (sets of) paths captures this intuition.

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

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

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

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

In the following we omit the superscript k of a the 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, without specifying whether the semantics which is referred to is the on-line or the off-line (see Theorem 4.1 in Section 4). Moreover, we denote by $[\![\varphi]\!] = \{s \in S \mid s \models \varphi\}$ the set of states that satisfy φ and by δ_φ the restriction of the transition function δ to $[\![\varphi]\!] \times [\![\varphi]\!]$ such that if $(s, s') \in \delta_\varphi$ and $\varphi = \langle\!\langle X \rangle\!\rangle \psi_1 \mathcal{U} \psi_2$ then ψ_1 holds in s. Moreover, for a graded ATL formula $\varphi = \langle\!\langle X \rangle\!\rangle \theta$, a tag $x \in \{\text{on}, \text{off}\}$, 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. If $s \models^x \langle\!\langle X \rangle\!\rangle^k \theta$ for all $k \geq 0$, we set $\mathit{grade}^x(s, \varphi) = \infty$. Finally, we denote the set of non-negative integers by $\mathbb{N}$ and set $\hat{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$.

3. A Fixpoint Characterization

In this section we provide a fixpoint characterization of the functions grade^x, for $x \in \{\text{on}, \text{off}\}$, when the formula $\varphi = \langle\!\langle X \rangle\!\rangle \theta$ is given, with either $\theta = \Box\psi$ or $\theta = \psi_1 \mathcal{U} \psi_2$.

We start with the off-line semantics. Define the following operator $F^{\text{off}}_\varphi : ([\![\varphi]\!] \to \hat{\mathbb{N}}) \to ([\![\varphi]\!] \to \hat{\mathbb{N}})$.

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

where $x \sqcup y$ denotes $\max\{x, y\}$.

Observe that $\mathit{grade}^{\text{off}}(s, \varphi) = 0$ (and also $\mathit{grade}^{\text{on}}(s, \varphi) = 0$) for all $s \in S \setminus [\![\varphi]\!]$.

Lemma 3.1. Let $\varphi = \langle\!\langle X \rangle\!\rangle \theta$, with either $\theta = \Box\psi$ or $\theta = \psi_1 \mathcal{U} \psi_2$, for some graded ATL formulas ψ, ψ_1 and ψ_2. The function $f : [\![\varphi]\!] \to \hat{\mathbb{N}}$ s.t. $f(s) = \mathit{grade}^{\text{off}}(s, \varphi)$, for $s \in [\![\varphi]\!]$, is the least fixpoint of F^{off}_φ.

Proof:
First, we prove that f is a fixpoint of F^{off}_φ. Let $\varphi = \langle\!\langle X \rangle\!\rangle \theta$ and $s \in [\![\varphi]\!]$, and, for all successors s_i of s such that $(s, s_i) \in \delta_\varphi$, let us set $k_i = \mathit{grade}^{\text{off}}(s_i, \varphi)$. 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 $\theta = \psi_1 \mathcal{U} \psi_2$, 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^{off}_φ. Precisely, we prove by induction on n the following statement: Let g be a fixpoint of F^{off}_φ and let $s \in [\![\varphi]\!]$, 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^{off}_φ, 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 $[\![\varphi]\!]$ is the set of states where $\langle\!\langle X \rangle\!\rangle \Box\psi$ holds, and thus each state in $[\![\varphi]\!]$ 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^{off}_φ

which is smaller than f. I.e., there is a state $s \in [\![\varphi]\!]$ 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 $g(t) \leq g(s)$ and $g(t) < f(t)$. Since t has more than one successor, by (1), for all successors u of t we have $g(u) < g(t) \leq g(s) \leq n$ 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. □

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

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

Lemma 3.2. Let $\varphi = \langle\!\langle X \rangle\!\rangle \theta$, with either $\theta = \Box\psi$ or $\theta = \psi_1 \mathcal{U} \psi_2$, for some graded ATL formulas ψ, ψ_1 and ψ_2. The function $f : [\![\varphi]\!] \to \hat{\mathbb{N}}$ s.t. $f(s) = \mathit{grade}^{\text{on}}(s, \varphi)$, for $s \in [\![\varphi]\!]$, 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 $s \in [\![\varphi]\!]$. Suppose that there is at least one successor of s in δ_φ (otherwise $\theta = \psi_1 \mathcal{U} \psi_2$, $s \models^{\text{on}} \psi_2$, and $F^{\text{on}}_\varphi(f)(s) = f(s) = 1$). For all successors s_i of s in δ_φ, let $k_i = \mathit{grade}^{\text{on}}(s_i, \langle\!\langle X \rangle\!\rangle \theta)$. 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 $\theta = \psi_1 \mathcal{U} \psi_2$, 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 3.1, we prove by induction on n the following statement: Let g be a fixpoint of F^{on}_φ and let $s \in [\![\varphi]\!]$, if $g(s) \leq n$ then $f(s) \leq g(s)$. 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 3.1. 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 $s \in [\![\varphi]\!]$ 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 $f(u) \geq f(t)$); 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. □

4. Comparing the Two Semantics

The example illustrated in the Introduction for the game in Figure 1 shows that the two semantics are different in general. Let us give some other examples to remark the differences between the two semantics. In these examples we consider two players, Player 1 and Player 2. In the figure, states of Player 1 are circles and those of Player 2 are squares.

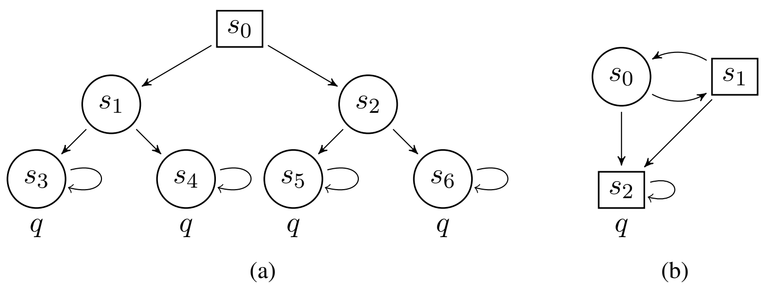

Figure 2: Two games where the two semantics differ.

Example 4.1. Consider the game in Figure 2a, where the goal for Player 1 is to reach the proposition q, which is true in the leaves of the tree. According to the off-line semantics, there are 4 possible strategies to achieve that goal. Namely, there are two choices from s_1 and two choices from s_2. The total number of strategies is then given by multiplying the two. Thus, we have $s_0 \models^{\text{off}} \langle\!\langle 1 \rangle\!\rangle^k \Diamond q$, for all $k \leq 4$ and $s_0 \not\models^{\text{off}} \langle\!\langle 1 \rangle\!\rangle^5 \Diamond q$. On the other hand, according to the on-line semantics, for all strategies of Player 2, there are only two paths satisfying $\Diamond q$. Thus, $s_0 \models^{\text{on}} \langle\!\langle 1 \rangle\!\rangle^2 \Diamond q$ and $s_0 \not\models^{\text{on}} \langle\!\langle 1 \rangle\!\rangle^3 \Diamond q$.

For a more extreme case, consider also the following.

Example 4.2. 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 off-line semantics, there are infinitely many strategies to achieve that goal. One strategy goes directly from s_0 to s_2. Another one goes first from s_0 to s_1 and then from s_0 to s_2, if the Player 2 moves back to s_0. Essentially, for all $k > 0$, there is a strategy of Player 1 that tries k visits to s_1 before going to s_2. Thus, we have $s_0 \models^{\text{off}} \langle\!\langle 1 \rangle\!\rangle^k \Diamond q$, for all $k > 0$. On the other hand, according to the on-line semantics, for all strategies of Player 2, there are only two paths leading to victory. Thus, $s_0 \models^{\text{on}} \langle\!\langle 1 \rangle\!\rangle^2 \Diamond q$ and $s_0 \not\models^{\text{on}} \langle\!\langle 1 \rangle\!\rangle^3 \Diamond q$.

The following theorem states that the two semantics coincide when all quantifiers have grade 1.

Theorem 4.1. For all games G, states s in G, and ATL state formulas φ, it holds that

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

Proof:
When all team quantifiers have grade 1, the classical quantifiers embedded in the ATL formula φ can be exchanged due to the well-known result by Martin on the determinacy of games with Borel objectives [17]. Exchanging the classical quantifiers leads from one semantics to the other. □

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. The same result does not hold for a general graded ATL formula, since the we know that off-line semantics and on-line semantics do not coincide. For example, for the Kripke structure 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 states s, and then, by Lemma 3.1 and Lemma 3.2, the following proposition follows.

Proposition 4.1. Let $\varphi = \langle\langle X\rangle\rangle\theta$, where either $\theta = \Box q$ or $\theta = p\,\mathcal{U} q$, and $s \in [\![\varphi]\!]$.

- 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 $i \geq 0$, $F_\varphi^{\text{on},i}(s) \leq F_\varphi^{\text{off},i}(s)$.

Now we can state the comparison between the two semantics.

Theorem 4.2. For a game G, a state s in G, and a negation-free graded ATL formula ψ

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

Proof:
Let $\psi = \langle\langle X\rangle\rangle^k\theta$, with either $\theta = \Box q$ or $\theta = p\,\mathcal{U} q$, $p, q \in \Sigma$, and $\varphi = \langle\langle X\rangle\rangle\theta$. From Proposition 4.1, $grade^{\text{on}}(s, \varphi) \leq grade^{\text{off}}(s, \varphi)$, 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 $\mathcal{U}$ operator is similar. □

5. Model Checking

Given $x \in \{\text{on}, \text{off}\}$, a state s in G and a graded ATL formula ψ, the model checking problem asks whether $s \models^x \psi$. In this section, we first consider $\psi = \langle\langle X\rangle\rangle^k\theta$ with only one team quantifier, and provide algorithms for solving a stronger form of model checking, that is we compute $grade^x(s, \langle\langle X\rangle\rangle\theta)$. Then we generalize the result and solve in polynomial time the model checking problem for a graded

ATL formula with any number of nested team quantifiers. Let us remark that 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 unambiguous) if it belongs to a player of team X and it has at least two successors.

5.1. Off-line Semantics

Given a path formula $\theta = \Box q$ or $\theta = p\mathcal{U}q$, we first describe an algorithm for computing $grade^{\text{off}}(s, \langle\langle X\rangle\rangle\theta)$ for all states $s \in S$ (the $\bigcirc$ operator is a simple case). Then, we show how to solve the model checking problem for a generic graded ATL formula using said algorithm. In the following, we say that a strongly connected component of a graph is a *sink* if there are no outgoing edges from it.

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

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 which have 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 5.1. For each state s, Algorithm 1 computes $grade^{\text{off}}(s, \varphi)$, for $\varphi = \langle\langle X\rangle\rangle\theta$, with $\theta = \Box q$ or $\theta = p\mathcal{U}q$. The algorithm runs in linear time.

Proof:
The algorithm first computes, as a base step, the states satisfying the ATL formula $\langle\langle X\rangle\rangle\theta$, and removes states and moves which do not contribute to winning this game. A state s receives 0 in the base step if it does not belong to $[\![\varphi]\!]$ and in fact $grade^{\text{off}}(s, \langle\langle X\rangle\rangle\theta) = 0$. 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). It is immediate to observe that all the states belonging to the same strongly connected component have the same grade, as they all have the same number of strategies.

The algorithm proceeds as follows: first it examines all sink components, that is components from which there are no edges outgoing to other components. If such a component does not contains a decision point, then the value 1 is assigned to it: if there is no decision point, there is only one strategy in the connected component. On the contrary, if the component has more than one state and contains a decision point, the value ∞ is assigned. Let us prove that this is correct. Let r be a decision point and suppose that r belongs to Player j, that is $\sigma(r) = j \in X$, and call r_1, r_2 two of its successors. Informally speaking, 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. More precisely, let $\alpha = r_1 \ldots r$ be a finite path in G, define the strategy $\sigma_h = (X, f_h)$ where $f_h(\alpha^h) = r_2$ and for $j < h$, $f_h(\alpha^j) = r_1$. Clearly, for each $h > 0$, the strategies σ_i, $i \leq h$, determine pairwise $(\langle\langle X\rangle\rangle\theta, \text{off})$-dissimilar $\mathrm{Outc}(r_2, \sigma_i)$ and thus, we have $r_2 \models^{\text{off}} \langle\langle X\rangle\rangle^k\theta$, 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.

Case 3d refers to singleton connected components. The algorithm correctly assigns the values of $grade^{\text{off}}(\cdot, \langle\langle X\rangle\rangle\theta)$ from Lemma 3.1.

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.

$\square$

To solve the model checking problem for graded ATL, we can follow the standard approach for the non-graded operators and design a trivial algorithm for the $\bigcirc$ operator. For the other temporal graded operators we can use Lemma 5.1 as follows. Suppose that G has been model-checked against a given graded ATL formula ψ. Then, to check whether $s \models^{\text{off}} \langle\langle X\rangle\rangle^{\hat{k}}\Box\psi$, for a given grade $\hat{k}$, Algorithm 1 can determine the greatest grade k such that $s \models \langle\langle X\rangle\rangle^k\Box q_\psi$ holds, for a new atomic proposition q_ψ, holding true in each state r such that $r \models^{\text{off}} \psi$. Similarly for the $\mathcal{U}$ operator. Thus, the following theorem holds.

Theorem 5.1. Given a game G, a state s in G 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.

The above complexity result assumes that each basic operation on integers is performed in constant time. Under this assumption, notice that the complexity of the model checking problem is independent of the integer constants appearing in the formula.

5.2. On-line Semantics

Similarly to the previous section, we describe an algorithm for computing $grade^{\text{on}}(s, \langle\langle X\rangle\rangle\theta)$ for $\theta = \Box q$ or $\theta = p\mathcal{U}q$, and for all states $s \in S$.
Given a path formula $\theta = \Box q$ or $\theta = p\mathcal{U}q$, Algorithm 2 computes $grade^{\text{on}}(s, \langle\langle X\rangle\rangle\theta)$, for all states $s \in S$. The complexity of the algorithm is dominated by step 3, which involves the solution of a Büchi game [18]. Standard algorithms perform this task in time $\mathcal{O}(|S| \cdot |\delta|)$, i.e., quadratic in the size of the adjacency-list representation of the game.

Algorithm 2 The algorithm computing $grade^{\text{on}}(\cdot, \varphi)$, given $\varphi = \langle\langle X \rangle\rangle \theta$, with $\theta = \Box q$ or $\theta = p \mathcal{U} q$.

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 decision points (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 shows that the above situation cannot arise, and therefore that the algorithm ultimately assigns a value to each state.

Lemma 5.2. At the end of step 3 of Algorithm 2, there is no loop of states with no value.

Proof:
By contradiction, assume that the thesis is false. We proceed by proving three intermediate claims, finally leading to a contradiction. For sake of simplicity, we prove our claims under the assumption that there are only two players and that $X = \{1\}$. The arguments can be easily generalized.

(a) First, we prove that all loops with no value contain a state which is labeled with d. If no state in the loop is labeled with d, each state in the loop either belongs to Player 2 or has one successor only. Therefore, all states in the loop satisfy $\langle\langle 2 \rangle\rangle \Box \neg d$ and receive value 1 during step 2, which is a contradiction. Thus, at least one state in the loop is labeled with d.

(b) A state of Player 2 with no value cannot have a successor with value 1, otherwise by step 2 it would have value 1 too, since $\neg d$ holds in all states of Player 2.

(c) Consider the set A of all states belonging to a loop of states with no value. Each state in A has a successor in A, with no value. If Player 1 always chooses to remain in A, by *(b)* we obtain an infinite path that either *(i)* remains forever in A, or *(ii)* eventually reaches a state with value ∞. In the first case, by *(a)* the infinite path contains infinitely many occurrences of d. In the second case, again Player 1 can enforce infinitely many visits to d. This proves that each state in A satisfies $\langle\langle 1 \rangle\rangle \Box \Diamond d$, which is a contradiction. Therefore, A must be empty and we obtain the thesis. □

Lemma 5.3. Given a path formula $\theta = \Box q$ or $\theta = p \mathcal{U} q$, at the end of Algorithm 2, each state s has value $grade^{\text{on}}(s, \langle\langle X \rangle\rangle \theta)$. The algorithm runs in quadratic time.

Proof:
We proceed by examining the four steps of the algorithm. If state s receives its value (zero) during step 1, it means that $s \not\models^{\text{on}} \langle\langle X \rangle\rangle \theta$. Therefore, zero is indeed the largest integer k such that $s \models^{\text{on}} \langle\langle X \rangle\rangle^k \theta$ 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 $s \models^{\mathrm{on}} \langle\!\langle X \rangle\!\rangle \Box \Diamond d$. 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, $s \models^{\mathrm{on}} \langle\!\langle X \rangle\!\rangle^k \theta$ holds for all $k > 0$.

Finally, if s receives its value during step 4, the correctness of the value is a consequence of Lemma 3.2. 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 5.2. Given a game G, a state s in G and a graded ATL formula ψ, the graded model checking problem, $s \models^{\mathrm{on}} \psi$, 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 for both off-line and on-line semantics.

Corollary 5.1. The graded ATL model checking problem is PTIME-complete.

6. Concurrent Games

In this section we investigate the model checking problems for the concurrent version of graded games. For a next-free graded ATL formula, we reduce its model checking problem to a 2-player game in the turn-based setting, for which we have already shown a solution in the previous section. The "next" operator can be easily handled by a specific algorithm.

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 $[\cdot] : S \to 2^{\Sigma}$ is the function assigning to each state s the set of atomic propositions that are true at s. In the following we will 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.

Notice that the comparison between the two semantics leads to different results in the case of concurrent games. Since concurrent games are not determined, Theorem 4.1 and Theorem 4.2 do not hold for concurrent games. In other words there are games, such as the classical Penny Matching game, where a simple ATL formula like $\langle\langle 1\rangle\rangle\Diamond p$ is true in the on-line semantics and false in the off-line semantics.

Model-checking complexity. The construction of a 2-player turn-based game G_A from a concurrent game G played by the team A is that of [5]; we briefly report it here. Consider a concurrent game structure $G = (m, S, pl, \delta, [\cdot])$ and a team A of players in $\{1, \ldots, m\}$. For a state $s \in S$, an A-*move* is a possible combination of moves of the players in A when the game is in state s. We denote by $C(A, s)$ the set of A-moves in s, and by $C(A) = \cup_{s\in S} C(A, s)$ the set of all A-moves. For an A-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 $a \in A$, is determined by the A-move c.

The 2-player turn-based game structure $G_A = (2, S', pl', \delta', [\cdot]')$ is defined as follows: the set of atomic propositions is augmented with a special proposition *aux*, the set of states is $S' = S \cup C(A)$, that is it contains S and, for each A-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 $c \in C(A, s)$, and there is an edge from $c \in C(A, s)$, for some s, to $s' \in S$ if s' is a c-successor of s.

Clearly G_A has $\mathcal{O}(|\delta|)$ states and edges. Moreover it is easy to see that for each strategy σ of team A in G there exists a corresponding strategy σ' in G_A such that every path π' in $\text{Outc}_{G_A}(s, \sigma')$ 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 $C(A)$, 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 $\theta = p\mathcal{U}q$, for atomic propositions p and q. The considerations above imply that the number of $(\langle\langle A\rangle\rangle\theta, \text{off})$-dissimilar strategies of A in G, is equal to the number of $(\langle\langle 1\rangle\rangle\theta, \text{off})$-dissimilar strategies of Player 1 in G_A.

Proposition 6.1. Let G be a concurrent game, $s \in S$, $p, q \in \Sigma$, $A \subseteq \{1, \ldots, m\}$, and $k > 0$. Then the following hold:

$$s \models^{\text{off}}_G \langle\langle A\rangle\rangle^k \Box p \text{ iff } s \models^{\text{off}}_{G_A} \langle\langle 1\rangle\rangle^k \Box(p \vee \textit{aux})$$

$$s \models^{\text{off}}_G \langle\langle A\rangle\rangle^k p\mathcal{U}q \text{ iff } s \models^{\text{off}}_{G_A} \langle\langle 1\rangle\rangle^k (p \vee \textit{aux})\mathcal{U}q.$$

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

Proposition 6.2. Let G be a concurrent game, $s \in S$, $p, q \in \Sigma$, $A \subseteq \{1, \ldots, m\}$, and $k > 0$. Then the following hold:

$$s \models^{\text{on}}_G \langle\langle A\rangle\rangle^k \Box p \text{ iff } s \models^{\text{on}}_{G_{\neg A}} \langle\langle 2\rangle\rangle^k \Box(p \vee \textit{aux})$$

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

We get a *PTIME*-completeness result for the graded concurrent game model-checking problem. The result comes from Corollary 5.1 and the construction of the 2-player turn-based game structure described above.

Theorem 6.1. The model-checking problem for graded ATL on concurrent games is PTIME-complete, and 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 game with set of states S and transition function δ and for a formula ψ.

To conclude this section, let us remark that a fixpoint characterization of the functions $grade^x$, for $x \in \{\text{on}, \text{off}\}$, can be given also in the concurrent game setting. For the off-line semantics, define the following operator $F^{\text{off},conc}_{\varphi} : ([\![\varphi]\!] \to \hat{\mathbb{N}}) \to ([\![\varphi]\!] \to \hat{\mathbb{N}})$.

$$F^{\text{off},conc}_{\varphi}(f)(s) = 1 \sqcup (\sum_{(s,s') \in \delta_\varphi} f(\prod_{(s',s'') \in \delta_\varphi} f(s'')) \tag{3}$$

where $x \sqcup y$ denotes $\max\{x, y\}$. For the on-line semantics, define the following operator $F^{\text{on},conc}_{\varphi}$: $([\![\varphi]\!] \to \hat{\mathbb{N}}) \to ([\![\varphi]\!] \to \hat{\mathbb{N}})$.

$$F^{\text{on},conc}_{\varphi}(f)(s) = 1 \sqcup (\min_{(s,s') \in \delta_\varphi} f(\sum_{(s',s'') \in \delta_\varphi} f(s'')) \tag{4}$$

The following proposition can be proved, extending the arguments used in Lemmas 1 and 2.

Proposition 6.3. Let G be a concurrent game and $\varphi = \langle\langle X \rangle\rangle \theta$ be an ATL formula, where either $\theta = \Box\psi$ or $\theta = \psi_1 \mathcal{U} \psi_2$, for some ψ, ψ_1 and ψ_2. Let $x \in \{\text{on}, \text{off}\}$ and $f_x : [\![\varphi]\!] \to \hat{\mathbb{N}}$ be a mapping such that $f_x(s) = grade^x(s, \varphi)$, for any state s of G belonging to $[\![\varphi]\!]$. Then, the function f_x is the least fixpoint of $F^{x,conc}_{\varphi}$.

7. Memoryless Semantics

In this section, we develop a semantics that only considers memoryless strategies. We consider turn-based games, except at the end of section, where we briefly extend our analysis to concurrent games. In the memoryless semantics, the meaning of the team quantifier is defined as follows, for a state s and a path formula θ.

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

Comparison with the other semantics. First, consider again the example in Figure 1, and suppose we add an edge from state s_2 to state s_1, labeled with “pdf2ps”. According to the off-line semantics, Player 1 has infinitely many strategies to reach s_4. On the other hand, there are only three memoryless strategies: one does not use the converting programs “ps2pdf” and “pdf2ps”, while the other two only use one of the converting programs. Using both converting programs leads to an infinite loop that does not reach state

s_4. The on-line semantics still assigns grade 1 to the property of reaching s_4. Hence, the memoryless semantics can give a grading value that is smaller than the off-line semantics and greater than the on-line semantics. The following examples complete the picture, showing that the memoryless semantics can be: *(i)* smaller than the off-line even when the latter produces a finite value, and *(ii)* greater than the on-line semantics.

In the first example, in Figure 3a, assume that the goal for Player 1 is to reach 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. On the other hand, according to the memoryless semantics, there are only two memoryless strategies for Player 1, one leading to s_4 and one leading to s_5.

Consider the game in Figure 3b, 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. On the other hand, there is only one memoryless winning strategy, i.e., the one going directly from s_1 to s_2.

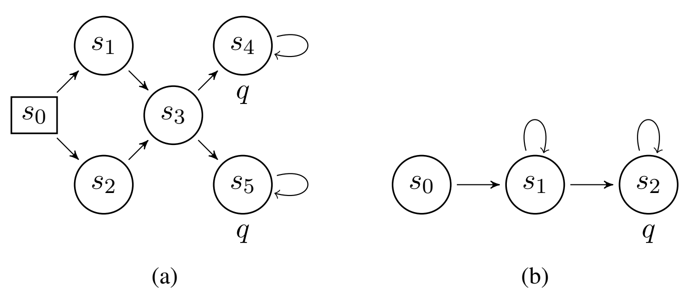

Figure 3: Two games refining the relationship between the three semantics.

Model checking. For $\theta \in \{\Box q, p\mathcal{U}q\}$, 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 3), 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 3, 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 $\rho \in \mathrm{Outc}(s, \sigma)$, 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.

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 2.2). 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 $\theta = p\mathcal{U}q$, 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 [20, 16].

Throughout this section, we consider a fixed game G and an ATL formula $\varphi \in \{\langle\langle X\rangle\rangle\Box q, \langle\langle X\rangle\rangle p\mathcal{U}q\}$. 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.

Moreover, whenever discussing about the "for all" loop of Algorithm 3, we denote by u_i the state picked on line 7 of the i-th iteration and by G_i the status of the game G when the recursive call on line 9 is made.

Lemma 7.1. For all iterations i of the "for all" loop in Algorithm 3, team X cannot avoid state u_i from s in all the games G_j, for $j \geq i$.

Proof:
Consider a generic invocation to count_mless. Denote by π the strategy computed on line 4 of the algorithm, we proceed by induction on the distance of u_i from s according to π. If the distance is 0, we have that $u_i = s$, and u_i is trivially unavoidable from s.

Assume instead that the distance is greater than zero. Let x be the last decision point of team X in a shortest path from s to u_i consistent with π (if there is no such state, u_i is trivially unavoidable). The state x has a distance from s according to π smaller than u_i. We distinguish two cases: *(i)* there is an index $h < i$ such that $x = u_h$, or *(ii)* there is no such index, because x is not winning. In case *(i)*, by inductive hypothesis x is unavoidable from s in the games G_j with $j \geq h$; moreover, there is only one edge left from x in the games G_j, with $j > h$. Hence, u_i is also unavoidable from s in the games G_j, with $j \geq i > h$. Case *(ii)*, instead, leads to the following contradiction: since s is winning in G and there is a path from s to x consistent with π, x must be winning as well (see line 3). □

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.

Lemma 7.2. All strategies computed on line 4 by Algorithm 3 are mutually (φ, off)-dissimilar at s.

Proof:
Consider a generic invocation to count_mless. Let π be the strategy computed on line 4. We prove that, for all i, the strategies computed during the i-th iteration of the "for all" loop are (φ, off)-dissimilar at s (in the following, *dissimilar*) from π and from the strategies computed during the following iterations. This property inductively ensures that all computed strategies are mutually dissimilar.

Each strategy π' computed during the i-th iteration cannot use a certain edge (u, v), removed on line 8. By Lemma 7.1, u is unavoidable from s in G_i. Hence, if s is winning in G_i, so is u. If instead s is not winning in G_i, there are no strategies computed in the i-th iteration. Moreover, $\pi'(u) \neq v = \pi(u)$. Therefore, considering that π' is uniformly winning in G_i, there is a path in $\mathrm{Outc}(s, \pi')$ that is dissimilar from all paths in $\mathrm{Outc}(s, \pi)$, i.e., π and π' are dissimilar.

Consider now a strategy π' computed in one of the following iterations $j > i$. Again, by Lemma 7.1, u is unavoidable from s in G_j. As before, if s is winning in G_j, so is u. The strategy π' contains the edge (u, v), because line 12 removes all other edges leaving u. Since all the strategies computed at the i-th iteration do not use the edge (u, v), and since π' is uniformly winning, and in particular winning from u, we obtain that π' is dissimilar from the strategies computed during the i-th iteration.

□

Given two strategies that are dissimilar at s, the following definition identifies the first state (i.e., the closest to s) that justifies this dissimilarity.

Definition 7.1. Given a state s and two strategies π and π' that are winning from s, a state u is called a *closest distinction* for (π, π', φ, s) iff it satisfies: *(i)* u is a decision point of team X that is reachable from s according to both π and π', *(ii)* $\pi(u) \neq \pi'(u)$, *(iii)* both π and π' are winning from u, and *(iv)* the distance between s and u according to π is minimal among all states satisfying *(i)-(iii)*.

Clearly, the above condition *(iii)* is non-trivial only if φ is an until formula. The following property is an immediate consequence of the definition of dissimilar (sets of) paths.

Lemma 7.3. If two memoryless winning strategies π and π' are (φ, off)-dissimilar at s then there exists a closest distinction for (π, π', s, φ).

We now prove a lower bound on the value returned by count_mless. Together with Lemma 7.2, this result implies the correctness of the algorithm.

Lemma 7.4. 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 $k \geq n$.

Proof:
Let $P = \{\pi_1, \ldots, \pi_n\}$ be a set of mutually dissimilar strategies that are winning from s w.r.t. φ. We proceed by induction on n. If $n = 0$, the result is trivially true.

Otherwise, consider the call count_mless(G, φ, k, s). Since there is at least a strategy that is winning from s, we have that $s \in W$, and the algorithm reaches line 5. Let π be the strategy computed on line 4. Assume that π is *not* dissimilar from one of the strategies in P, say π_1. The opposite case can be treated similarly. If $k = 1$, the algorithm returns 1 and we are done. Otherwise, the algorithm enters the "for all" loop.

We show that each strategy in $P \setminus \{\pi_1\}$ is a valid strategy in one of the games G_i. Let $\pi' \in P \setminus \{\pi_1\}$, by Lemma 7.3 there is a closest distinction u for (π, π', s, φ). Since u respects all the conditions of line 7, there is an iteration i such that $u = u_i$. During that iteration, a recursive call is made on the subgame G_i. Compared to G, for all $j < i$, in G_i all edges leaving u_j have been removed from the game, except $(u_j, \pi(u_j))$. Moreover, the edge $(u_i, \pi(u_i))$ has been removed. By definition of closest distinction, we have that for all $j < i$, $\pi'(u_j) = \pi(u_j)$, and $\pi'(u_i) \neq \pi(u_i)$. Therefore, π' is a valid strategy in G_i.

By inductive hypothesis, since there are $n - 1$ strategies in P that are dissimilar from π, there are integers n_i such that the recursive call during the i-th iteration returns the value n_i and $\sum_i n_i \geq n - 1$.

□

The following result characterizes the time complexity of Algorithm 3, in terms of calls to the procedures get_winning_set and get_uniformly_winning_strategy.

Lemma 7.5. 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 7.5 we obtain the following.

Corollary 7.1. 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 3 The procedure count_mless(G, φ, k, s).

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

1: $W :=$ get_winning_set(G, φ)
2: **if** $s \notin W$ **then return** 0
3: $G' := (m, S, pl, \delta_\varphi, [\cdot])$
4: $\pi :=$ get_uniformly_winning_strategy(G', φ)
5: **if** $k = 1$ **then return** 1
6: $n := 1$
7: **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**
8: remove_edge$(G', (u, \pi(u)))$
9: $n := n +$ count_mless$(G', \varphi, k - n, s)$
10: add_edge$(G', (u, \pi(u)))$
11: **if** $n = k$ **then return** n
12: remove_edges$(G', \{(u, x) \mid x \neq \pi(u)\})$
13: **end for**
14: **return** n

From previous lemmas and by using the standard argument for dealing with nested formulas, we can get a solution of the model checking problem for a graded ATL formula, under the memoryless semantics.

Theorem 7.1. Given a game G, a state s in G 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.

Following the approach outlined in Section 6, we can model check a graded ATL formula on a *concurrent* game in the memoryless semantics. For a concurrent game G and a graded ATL formula $\psi = \langle\langle A \rangle\rangle^k \theta$, we construct the corresponding 2-player turn-based game G_A. By applying the algorithm presented in this section to G_A, we can model-check ψ in time $\mathcal{O}(\hat{k} \cdot |\delta|^2 \cdot |\psi|)$. Nesting and boolean operators can be handled recursively, in the usual way.

8. Conclusions

In this paper, we explored the consequences of adding counting capabilities to the team quantifiers of the game logic ATL. Such capability naturally leads to two different interpretations, called the “off-line” and the “on-line” semantics.

Memoryless. The semantics introduced in Section 7 is essentially the memoryless version of the off-line semantics. A similar technique leads to the definition of a memoryless version of the on-line semantics, which we plan to investigate in the future. In such a semantics, formula $\langle\langle X \rangle\rangle^k \theta$ is true if, for all strategies of team $\neg X$, there are k memoryless paths (i.e., simple paths, in the graph theoretic meaning) that satisfy θ. Preliminary results show that the model checking problem for this semantics may be substantially harder than the semantics studied in this paper.

Related work. Several extensions of ATL have been proposed that enrich team quantifiers with integer grades. However, these proposals interpret the grades differently. In [3], integral grades are used to bound the amount of memory that each strategy can employ. On the other hand, [4] provides an abstract framework where moves in the game are associated with a cost vector (one cost for each resource in a given set) and the logic constrains strategies w.r.t. the amount of resources they can consume.

A concept similar to our fault tolerance is that of robustness of [10], though, as the authors observe, “the Robustly operator is a path quantifier that is not a state formula” which may vanish when applied to pure CTL-like operators. Moreover, the model-checking complexity of the RoCTL* is nonelementary. Close enough to these arguments are also those of robustness of [1, 2] intended as “deviations from a norm” and interpreted as missing actions (missing edges in the structure).

Expressive power. Our comparisons only show the relationship between the value achieved by *the same formula*, when interpreted according to different semantics. The relationship between the expressive power of the semantics is instead an open question. To make progress on this issue, it would be beneficial to study the equivalence relations induced by graded ATL, possibly along the lines of [6].

Future work. The satisfiability problem for ATL has been analyzed in various forms, which vary according to the assumptions made on the set of possible agents. In all cases, it was shown to be EXPTIME-complete [19]. One interesting future direction of research is the complexity of the satisfiability problem for graded ATL.

Another appealing field is that of the axiomatization of graded-ATL. As far as we know, no axiomatization has been given for temporal graded logics. On the contrary, the axiomatization of ATL is investigated in [11], that also provides an automata-theoretic algorithm for the satisfiability problem.

References

[1] Ågotnes, T., van der Hoek, W., Wooldridge, M.: Robust normative systems, *AAMAS (2)*, 2008.

[2] Ågotnes, T., van der Hoek, W., Wooldridge, M.: Robust Normative Systems and a Logic of Norm Compliance, *Journal of Logic, Language and Information*, **18**(1), 2010, 4–30.

[3] Ågotnes, T., Walther, D.: A Logic of Strategic Ability Under Bounded Memory, *J. of Logic, Lang. and Inf.*, **18**(1), 2009, 55–77, ISSN 0925-8531.

[4] Alechina, N., Logan, B., Nga, N. H., Rakib, A.: A logic for coalitions with bounded resources, *IJCAI'09: Proceedings of the 21st international jont conference on Artifical intelligence*, Morgan Kaufmann Publishers Inc., 2009.

[5] Alur, R., Henzinger, T., Kupferman, O.: Alternating-Time Temporal Logic, *J. ACM*, **49**, 2002, 672–713.

[6] Corradini, F., De Nicola, R., Labella, A.: Graded Modalities and Resource Bisimulation, *Proceedings of the 19th Conference on Foundations of Software Technology and Theoretical Computer Science*, Springer-Verlag, 1999, ISBN 3-540-66836-5.

[7] Ferrante, A., Murano, A., Parente, M.: Enriched μ-Calculi Module Checking, *Logical Methods in Computer Science*, **4**(3), 2008.

[8] Ferrante, A., Napoli, M., Parente, M.: CTL Model-Checking with Graded Quantifiers, *Proc. of ATVA '08*, 5311, 2008.

[9] Fine, K.: In so many possible worlds, *Notre Dame journal of Formal Logic*, **13**(4), 1972, 516–520.

[10] French, T., McCabe-Dansted, J. C., Reynolds, M.: A Temporal Logic of Robustness, *LNCS 4720, FroCos*, 2007.

[11] Goranko, V., van Drimmelen, G.: Complete axiomatization and decidability of Alternating-time temporal logic, *Theoretical Computer Science*, **353**(1-3), 2006, 93–117.

[12] Gradel, E., Otto, M., Rosen, E.: Two-variable logic with counting is decidable, *Proc. of LICS97*, 1997.

[13] Hollunder, B., Baader, F.: Qualifying Number Restrictions in Concept Languages, *2nd International Conference on Principles of Knowledge Representation and Reasoning*, 1991.

[14] Immerman, N.: Number of Quantifiers is Better Than Number of Tape Cells, *J. Comput. Syst. Sci.*, **22**(3), 1981, 384–406.

[15] Kupferman, O., Sattler, U., Vardi, M.: The Complexity of the Graded μ-Calculus, *Proc. 18th Conference on Automated Deduction*, 2392, 2002.

[16] Lawler, E. L.: A Procedure for Computing the K Best Solutions to Discrete Optimization Problems and Its Application to the Shortest Path Problem, *Management Science*, **18**(7), 1972, 401–405.

[17] Martin, D.: Borel determinacy, *Annals of Mathematics*, **102**(2), 1975, 363–371.

[18] Thomas, W.: On the Synthesis of Strategies in Infinite Games, *Proc. of STACS95*, 900, Springer-Verlag, 1995.

[19] Walther, D., Lutz, C., Wolter, F., Wooldridge, M.: ATL Satisfiability is Indeed EXPTIME-complete, *J. of Logic and Computation*, **16**(6), 2006.

[20] Yen, J. Y.: Finding the K Shortest Loopless Paths in a Network, *Management Science*, **17**(11), 1971, 712–716.