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

June 5, 2009

Abstract. Recently, graded modalities have been added to the semantics of two of the logics most commonly used by the computer science community: μ-calculus, in [KSV02], and Ctl in [FNP08]. 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. To this aim, we consider the Alternating-time Temporal Logic (Atl), a derivative of Ctl that is interpreted on *games*, rather than transition systems.
We present 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.
For both these semantics, we efficiently solve the model checking problem, computing the truth values of graded Atl formulas on the states of a given game.

1 Introduction

Graded modalities are logical operators, especially well-known in the knowledge representation field, allowing to express quantitative bounds on the set of individuals satisfying a certain property [Fin72]. Similar notions are the counting quantifiers in classical logic [GOR97] and the number restrictions in description logics [HB91]. Recently, such notions have received renewed attention by the theoretical computer science community [CDL99,KSV02,FNP08,FMP08]. In these works, the authors have introduced graded modalities into the logics commonly used in computer science, such as the temporal logic Ctl and the μ-calculus. In this paper, we make progress along these lines by adding graded modalities to the so-called Alternating-time Temporal Logic, Atl, and providing efficient model-checking algorithms for the resulting logic. Atl was introduced by Alur et al. [AHK02] 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, it has been implemented in several tools for the analysis of such systems [AHM+98,LR06], and it has provided the basis for further extensions [ÅGJ07,vdHW03].

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 over strategies: there exists a strategy of Player 1, such that, for all strategies of Player 2, θ holds in the resulting outcome. Notice that, by a well-known result by Martin [Mar75] on the determinacy of games with Borel objectives, the two quantifiers above can be exchanged with no effect on the semantics.

Standard CTL path quantifiers can be obtained as special cases of ATL quantifiers. In a game with two players [1], the ATL formula $\langle\langle 1, 2\rangle\rangle\theta$ states that the two players together can cooperate to ensure θ. Since this formula puts both players in the same team, it is equivalent to the CTL formula $\exists\theta$.

In this paper, we enrich the ATL quantifiers with an integral *grade*, as follows. For a natural number k, the graded ATL formula $\langle\langle 1\rangle\rangle^k\theta$ affirms that Player 1 has k different strategies for ensuring θ. This corresponds to applying the grade k to the classical existential quantifier hidden in the team quantifier $\langle\langle 1\rangle\rangle$. Now, simple examples show that the two classical quantifiers *cannot* be exchanged if one of them (i.e., the existential one) has been enriched with a grade. Therefore, we end up with two alternative semantics for the graded quantifier $\langle\langle 1\rangle\rangle^k$. In the first semantics, that we already presented and that we call the *off-line* semantics, the existential quantifier comes before the universal one, leading to the following interpretation: "Player 1 has k different strategies such that for all Player 2's strategies...". In the second semantics, that we call the *on-line* semantics, the universal quantifiers comes first, leading to "For all strategies of Player 2, Player 1 has k different strategies such that...".

The first semantics seems suitable to off-line synthesis applications. In this context, the 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.

On the other hand, the second semantics may find application in the verification of fault-tolerant controllers. In this case, we do not wish to restrict the moves of Player 1 (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 just 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 semantics, if the formula $\langle\langle 1\rangle\rangle^k\theta$ holds true, then, for all behaviors of the environment, the controller has k different ways to achieve its goal. This can be interpreted as a rough estimate of the amount of fault-tolerance of the controller, in the following sense: The controller tolerates *some* sets of k faults, even if a single fault may invalidate all

[1] The original definition of ATL allows for multiple-player games. In this paper, for simplicity we treat two-player games only.

k ways to achieve the goal. We call this semantics "on-line" because it is related to the ability of the player to dynamically alter its behavior to overcome such faults.

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 ATL formulas on the states of a given game. Finally, in Section 6 we give some conclusions.

2 Definitions

We treat games that are played by two players on a finite graph, where each state belongs to one of the players. The game starts in a state of the graph, and at each turn the player who owns the current state chooses one of its outgoing edges. As a consequence, the game *moves* to the destination state of that edge. The game continues in this fashion, until an infinite path in the game is formed. For an introduction to the general setting of game theory, see [OR94], while for a more specific introduction to games played on graphs, see [Tho95]. Throughout the paper, we consider a fixed set Σ of *atomic propositions*. The following definitions make this framework formal.

Games. A *game* is a tuple $G = (S_1, S_2, \delta, [\cdot])$ such that: S_1 and S_2 are disjoint finite sets of states; let $S = S_1 \cup S_2$, we have that $\delta \subseteq S \times S$ 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 $G = (S_1, S_2, \delta, [\cdot])$.

A (finite or infinite) path in G is a (finite or infinite) path in the directed graph $(S_1 \cup S_2, \delta)$. Given a path ρ, we denote by $\rho(i)$ its i-th state, by $\mathit{first}(\rho)$ the first state, and by $\mathit{last}(\rho)$ the last state, when ρ is finite.

Strategies. A *strategy* in G is a pair (X, f), where $X \subseteq \{1, 2\}$ 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*. For a team $X \subseteq \{1, 2\}$, we denote by S_X the set of states belonging to team X, i.e. $S_X = \bigcup_{i \in X} S_i$, and we denote by $\neg X$ the opposite team, i.e. $\neg X = \{1, 2\} \setminus X$.

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. For two strategies $\sigma = (X, f)$ and $\tau = (\neg X, g)$, and a state s, we denote by $\mathrm{Outc}_G(s, \sigma, \tau)$ the unique infinite path which starts from s and is consistent with both σ and τ.

2.1 Logic

The logic ATL is defined in [AHK02]. We extend it with graded quantifiers.

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

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

where $q \in \Sigma$ is an atomic proposition, $X \subseteq \{1, 2\}$ 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, also the operator $\Diamond$ (eventually) can be introduced using the equivalence $\Diamond p \equiv \mathit{true}\,\mathcal{U} p$. 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 Section 4. Their satisfaction relations are denoted by $\models^{\text{off}}$ 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 the two semantics. Let ρ be an infinite path in the game, s be a state, and φ_1, φ_2 be state formulas. Denote by $\mathbb{N}$ the set of non-negative integers. For $x \in \{\text{on}, \text{off}\}$, the satisfaction relations are defined as follows.

$\rho \models^x \bigcirc\varphi_1$	iff $\rho(1) \models^x \varphi_1$	
$\rho \models^x \Box\varphi_1$	iff $\forall i \in \mathbb{N} (i) \models^x \varphi_1$	
$\rho \models^x \varphi_1 \mathcal{U} \varphi_2$	iff $\exists j \in \mathbb{N} (j) \models^x \varphi_2$ and $\forall 0 \le i < j\, (i) \models^x \varphi_1$	(†)
$s \models^x q$	iff $q \in [s]$	
$s \models^x \neg\varphi_1$	iff $s \not\models^x \varphi_1$	
$s \models^x \varphi_1 \vee \varphi_2$	iff $s \models^x \varphi_1$ or $s \models^x \varphi_2$.	

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 $0 \le i \le \min\{|\rho|, |\rho'|\}$ 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 a path formula ψ and $x \in \{\text{on}, \text{off}\}$, we say that two infinite paths ρ and ρ' are (ψ, x)*-dissimilar* iff:

- $\psi = \bigcirc\varphi$ and $\rho(1) \neq \rho'(1)$, or
- $\psi = \Box\varphi$ and $\rho(i) \neq \rho'(i)$ for some i, or
- $\psi = \varphi_1 \mathcal{U} \varphi_2$ and there are two integers j and j' such that:
  - $\rho(j) \models^x \varphi_2$,
  - $\rho'(j') \models^x \varphi_2$,
  - for all $0 \le i < j$, $\rho(i) \models^x \varphi_1$,
  - for all $0 \le i' < j'$, $\rho'(i') \models^x \varphi_1$, and
  - $\rho_{\le j}$ and $\rho'_{\le j'}$ are dissimilar.

Finally, two sets of infinite paths are (ψ, x)-dissimilar iff one set contains a path which is (ψ, x)-dissimilar to all the paths in the other set.

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 $p\mathcal{U}q$, 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 for the 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 ψ.

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

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

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

For an ATL formula φ, a tag $x \in \{\text{on}, \text{off}\}$, and a state s, we set $grd^x(s, \varphi)$ to be the greatest integer k such that $s \models^x \varphi^k$ holds, where φ^k is obtained from φ by assigning grade k to the outmost team quantifier. If such integer does not exist, we set $grd^x(s, \varphi) = \infty$. We say that a state is a *decision point* if it belongs to Player 1 and it has at least two successors.

3 Fixpoint Characterization

In this section we provide a fixpoint characterization of the function $grd(s, \varphi)$ when the state s is given.

Let $T \subseteq S$ be the set of states of G where the ATL formula $\langle\langle 1\rangle\rangle\theta$ holds, for $\theta = \Box q$ or $\theta = p\mathcal{U}q$, and let $\hat{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$. Consider the following functor $F^{\text{on}} : (T \to \hat{\mathbb{N}}) \to (T \to \hat{\mathbb{N}})$.

$$F^{\text{on}}(f)(s) = 1 \sqcup \begin{cases} \sum_{(s,s') \in \delta \text{ and } s' \in T} f(s') & \text{if } s \in S_1 \\ \min_{(s,s') \in \delta \text{ and } s' \in T} f(s') & \text{if } s \in S_2, \end{cases} \tag{1}$$

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

Lemma 1. *Given $\theta = \Box q$ or $\theta = p\mathcal{U}q$, let T be the set of states where $\langle\langle 1\rangle\rangle\theta$ holds. The function $f : s \in T \to grd^{\text{on}}(s, \langle\langle 1\rangle\rangle\theta) \in \mathbb{N}$ is the least fixpoint of F^{on}.*

Proof. First, we prove that f is a fixpoint of F^{on}. Let $s \in T$. If $\theta = \Box q$, since $s \models^{\text{on}} \langle\langle 1\rangle\rangle\theta$, we have that $s \models^{\text{on}} q$. If instead $\theta = p\mathcal{U}q$, either $s \models^{\text{on}} p$ or $s \models^{\text{on}} q$.

For all successors s_i of s, let $k_i = grd^{\text{on}}(s_i, \langle\langle 1\rangle\rangle\theta)$. For each strategy τ of Player 2, there are k_i dissimilar paths starting from s_i, consistent with τ, and satisfying θ. Therefore, if $s \in S_1$, 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. If instead $s \in S_2$, let $i = \arg(\min_j k_j)$. Consider the memoryless strategy τ of Player 2 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 Player 2 have at least as many dissimilar paths from s.

Next, we prove that f is the *least* fixpoint of F^{on}. Precisely, we prove by induction on n the following statement: Let g be a fixpoint of F^{on} and let $s \in T$, if $g(s) \leq n$ then $f(s) \leq g(s)$. Assume for simplicity that $\theta = \Box q$, as the other case can be proved along similar lines.

If $n = 1$, by hypothesis $g(s) = 1$. Considering the definition of operator (1), there are the following three possibilities: *(i)* s has no successors in T; *(ii)* s belongs to S_1 and has only one successor in T; *(iii)* s belongs to S_2 and has a successor t in T such that $g(t) = 1$. Option *(i)* can be discarded because T is the set of states where $\langle\langle 1\rangle\rangle\Box q$ holds, and thus each state in T has at least one successor in T. Given the remaining two options, one can see that Player 2 can force the game in a loop where all states x have value $g(x) = 1$, and Player 1 cannot exit this loop. Accordingly, we have $f(s) = 1$, as requested.

If $n > 1$, by contradiction, let g be a fixpoint of F^{on} which is smaller than f. I.e., there is a state $s \in T$ 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 $t \in S_2$, pick as the next state of the path a successor $u \in T$ of t such that $g(u) = g(t)$ (notice that $f(u) \geq f(t)$); if $t \in S_1$ and t has only one successor u in T, pick u as the next state (notice that $g(u) = g(t)$ and $f(u) = f(t)$); finally, if $t \in S_1$ and t has more than one successor in T, stop. If the above process continues forever, it means that Player 2 can force the game in a loop from which Player 1 cannot exit. As before, 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 in T, by (1), 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 induction $g(u^*) \geq f(u^*)$, which is a contradiction. ∎

Let us observe the following proposition that immediately follows from the Lemma and that will be used in the following.

Proposition 1. *Let $G = (S_1, S_2, \delta, [\cdot])$ be a game and let $\varphi = \langle\langle 1\rangle\rangle^k\theta$, for $k > 1$ be a graded formula where $\theta = \Box q$ or $\theta = p\mathcal{U}q$. Given $s \in S$, if $s \models^{\text{on}} \varphi$, then:*

– *if* $s \in S_1$ *then there exist* t *successors* $s_1, \ldots, s_t$ *of* s *in* δ*, and* t *integers* $k_1, \ldots, k_t$*, such that for all* $i = 1, \ldots, t$*, we have* $s_i \models^{\text{on}} \langle\langle 1 \rangle\rangle^{k_i} \theta$*, and* $\sum_{i=1}^{t} k_i = k$*.*
– *if* $s \in S_2$ *then all successors* s' *of* s *in* δ *are such that* $s' \models^{\text{on}} \langle\langle 1 \rangle\rangle^{k} \theta$*.*

Let T be the set of states where $\langle\langle 1 \rangle\rangle\theta$ holds. Consider the following operator $F^{\text{off}} : (T \to \hat{\mathbb{N}}) \to (T \to \hat{\mathbb{N}})$.

$$F^{\text{off}}(f)(s) = 1 \sqcup \begin{cases} \sum_{(s,s') \in \delta \text{ and } s' \in T} f(s') & \text{if } s \in S_1 \\ \prod_{(s,s') \in \delta \text{ and } s' \in T} f(s') & \text{if } s \in S_2. \end{cases} \tag{2}$$

Lemma 2. *Given* $\theta = \Box q$ *or* $\theta = p\,\mathcal{U}q$*, let* T *be the set of states where* $\langle\langle 1 \rangle\rangle\theta$ *holds. The function* $f : s \in T \to grd^{\text{off}}(s, \langle\langle 1 \rangle\rangle\theta) \in \mathbb{N}$ *is the least fixpoint of* F^{off}*.*

Proof. First, we prove that f is a fixpoint of F^{on}. Let T be the set of states where $\langle\langle 1 \rangle\rangle\theta$ holds, and let $s \in T$. For all successors s_i of s in T, let us define $k_i = grd^{\text{off}}(s, \langle\langle 1 \rangle\rangle\theta)$. That is, k_i strategies of Player 1 exist which determine k_i (θ, off)-dissimilar sets of paths, $\text{Outc}(s_i, \cdot)$, consistent with the strategies and satisfying θ.

If $s \in S_1$, then the total number of strategies is the sum of $grd^{\text{off}}(s_i, \langle\langle 1 \rangle\rangle\theta)$, for each successor s_i of s in T. In fact, each path consistent with a strategy of Player 1 starting from s, can be obtained by adding the state s in front of one of the k_i paths consistent with a strategy starting from s_i.

If $s \in S_2$, then Player 1 has no choices in s and thus all the strategies starting from s are obtained by choosing one of the k_i strategies which start from each successor s_i.

Next, we prove that f is the *least* fixpoint of F^{off}. Similarly to the proof of Lemma 1, we prove by induction on n the following statement: Let g be a fixpoint of F^{off} and let $s \in T$, if $g(s) \leq n$ then $f(s) \leq g(s)$. Assume as before that $\theta = \Box q$ (the other case is similar).

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^{off} which is smaller than f. I.e., there is a state $s \in T$ 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 in T, pick u as the next state (notice that $g(u) = g(t)$ and $f(u) = f(t)$); if $t \in S_2$ and t has more than one successor in T, pick as the next state of the path a successor $u \in T$ of t such that $g(u) < f(u)$ (it is a simple matter of algebra to show that such a state exists); finally, if $t \in S_1$ and t has more than one successor in T, stop. If the above process continues forever, it means that Player 2 can force the game in a loop from which Player 1 cannot exit. As before, this means that $f(s) = 1$, which is a contradiction.

Otherwise, the above process stops in a state $t \in S_1$, such that $g(t) \leq g(s)$ and $g(t) < f(t)$. Since t has more than one successor in T, by (2), 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 induction $g(u^*) \geq f(u^*)$, which is a contradiction. ∎

4 Comparing the Two Semantics

The following examples show that in general the two semantics are different. In the figure, states of S_1 are circles and those of S_2 are squares.

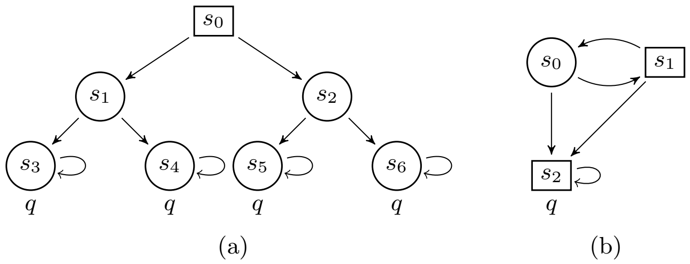

Fig. 1: Two games where the two semantics differ.

Example 1. Consider the game in Figure 1a, 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 $s_0 \models^{\text{off}} \langle\langle 1 \rangle\rangle^k \Diamond q$, $k \leq 4$ and $s_0 \not\models^{\text{off}} \langle\langle 1 \rangle\rangle^5 \Diamond q$. On the other hand, according to the online 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 2. Consider the game in Figure 1b, 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 $i > 0$, there is a strategy of Player 1 that tries i 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 online 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 1. *For all games G, states s in G, and* ATL *state formulas φ, it holds that*

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

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 [Mar75]. Exchanging the classical quantifiers leads from one semantics to the other.∎

This theorem allows us to simply say in the following that a state s *satisfies* an ATL formula, without specifying whether the semantics is referred to, is the on-line or the off-line.

Now we prove that the on-line satisfaction of a formula implies its off-line satisfaction as well. First, let us give a technical lemma.

Lemma 3. *Let $G = (S_1, S_2, \delta, [\cdot])$ be a game and let $\varphi = \langle\langle 1 \rangle\rangle^k \theta$, for $k > 1$ be a graded formula where $\theta = \Box q$ or $\theta = p\mathcal{U}q$. Let $s \in S$:*

1. *if $s \models^{\text{on}} \varphi$, then there exists a finite path ρ with $first(\rho) = s$, $last(\rho) \in S_1$, and $\rho(i) \models^{\text{on}} \langle\langle 1 \rangle\rangle^1 \theta$ for all i. Moreover, $last(\rho)$ has at least two successors s_j in δ, such that $s_j \models^{\text{on}} \langle\langle 1 \rangle\rangle^1 \theta, j = 1, 2$;*
2. *if there exists a finite path ρ with $first(\rho) = s$ and $last(\rho) \in S_1$ and such that $\rho(i) \models^{\text{off}} \langle\langle 1 \rangle\rangle^1 \theta$, for all i, and $last(\rho) \models^{\text{off}} \langle\langle 1 \rangle\rangle^k \theta$, then $first(\rho) \models^{\text{off}} \langle\langle 1 \rangle\rangle^k \theta$.*

Proof. Let us first prove Item 1. From the definition of on-line semantics, at least two paths ρ_1 e ρ_2 exist such that they are (θ, on)-dissimilar and both satisfy θ according to the on-line semantics. From this dissimilarity of ρ_1 and ρ_2, there exists an $i > 1$ such that $\rho_1(i) = \rho_2(i) \in S_1$ but $\rho_1(i+1) \neq \rho_2(i+1)$ and thus $\rho = first(\rho_1) \ldots \rho_1(i)$ satisfies the Lemma. Note that since $k > 1$, s cannot belong to a connected component of G which does not have edges outgoing from it and that does not contain states of S_1.

Consider now Item 2. Let ρ be a path satisfying the hypothesis. We prove that $\rho(i) \models^{\text{off}} \langle\langle 1 \rangle\rangle^k \theta$, for all $0 \leq i < |\rho|$. Since $\rho(i) \models^{\text{off}} \langle\langle 1 \rangle\rangle^1 \theta$, then $grd^{\text{off}}(s', \langle\langle 1 \rangle\rangle \theta) > 0$, for each successor s' of $\rho(i)$ in δ and thus $grd^{\text{off}}(\rho(i+1), \langle\langle 1 \rangle\rangle \theta) \leq grd^{\text{off}}(\rho(i), \langle\langle 1 \rangle\rangle \theta)$, from Lemma 2. Hence, if $\rho(i+1) \models^{\text{off}} \langle\langle 1 \rangle\rangle^k \theta$ then, $k \leq grd^{\text{off}}(\rho(i+1), \langle\langle 1 \rangle\rangle \theta) \leq grd^{\text{off}}(\rho(i), \langle\langle 1 \rangle\rangle \theta)$ and the thesis follows. ∎

Theorem 2. *For all games G, states s in G, and a graded* ATL *formula φ*

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

Proof. Let us first consider formulas $\varphi = \langle\langle 1 \rangle\rangle^k \theta$ con $\theta = \Box q$ or $\theta = p\mathcal{U}q$, where $p, q \in \Sigma$. The proof proceeds by induction on k. The base case for $k = 1$ follows from Theorem 1.

If $k > 1$, from Item 1 of Lemma 3 a path ρ of length $h \geq 0$ exists, whose states satisfy $\langle\langle 1 \rangle\rangle^1 \theta$ and with $\rho(0) = s$ and $\rho(h) \in S_1$, and this latter having two successors satisfying $\langle\langle 1 \rangle\rangle^1 \theta$. Without loss of generality suppose that each $\rho(j)$, $0 \leq j < h$ either belongs to S_2 or has just one successor satisfying $\langle\langle 1 \rangle\rangle^1 \theta$ (that is $\rho(h)$ is the first state occurring in ρ which belongs to S_1 and which has two successors satisfying $\langle\langle 1 \rangle\rangle^1 \theta$). This implies, by Proposition 1, that if $\rho(j) \models^{\text{on}} \varphi$ then also $\rho(j+1) \models^{\text{on}} \varphi$. Hence since $\rho(0) \models^{\text{on}} \varphi$, then $\rho(h) \models^{\text{on}} \varphi$, as well.

On the other hand, again from Proposition 1 also follows that there are $s_1, \ldots, s_t$ successors of $\rho(h)$ in δ, and $k_1, \ldots, k_t$ such that $\sum_{i=1}^{t} k_i = k$ e $s_i \models^{\text{on}} \langle\langle 1\rangle\rangle^{k_i}\theta$. Since $\rho(h)$ has at least two successors satisfying $\langle\langle 1\rangle\rangle^{1}\theta$, then $t > 0$ and $k_i < k$. Thus, from the inductive hypothesis, $s_i \models^{\text{off}} \langle\langle 1\rangle\rangle^{k_i}\theta$. Moreover from Lemma 2 $k = \sum_{i=1}^{t} k_i \leq \sum_{i=1}^{t} grd^{\text{off}}(s_i, \langle\langle 1\rangle\rangle\theta) = grd^{\text{off}}(\rho(h), \langle\langle 1\rangle\rangle\theta)$ and thus $\rho(h) \models^{\text{off}} \varphi$. Now noticing that the path ρ satisfies the hypothesis of Item 2 of Lemma 3 and $s = \rho(0)$, we can conclude that $s \models^{\text{off}} \varphi$.

To complete the proof of our statement we proceed by structural induction on a generic graded ATL formula. The base is trivial for the atomic propositions and for the non-temporal operators. Let φ be a graded ATL formula for which we inductively suppose that if $r \models^{\text{on}} \varphi$ then $r \models^{\text{off}} \varphi$, for all states r of G. If $s \models^{\text{on}} \langle\langle 1\rangle\rangle^{k}\bigcirc\varphi$, the statement trivially follows. Suppose now that $s \models^{\text{on}} \langle\langle 1\rangle\rangle^{k}\Box\varphi$ and let $\hat{G}$ be a new game obtained from G by simply renaming the states as follows: $\hat{S}_i = \{\hat{s} | s \in S_i\}, i = 1, 2$ and adding the new atomic proposition q_φ, holding true in all the states $\hat{r}$ such that $r \models^{\text{on}} \varphi$. Clearly, $\hat{s} \models^{\text{on}} \langle\langle 1\rangle\rangle^{k}\Box q_\varphi$ and, as shown above, $\hat{s} \models^{\text{off}} \langle\langle 1\rangle\rangle^{k}\Box q_\varphi$. It is easy to see that this implies that $s \models^{\text{off}} \langle\langle 1\rangle\rangle^{k}\Box\varphi$, as well. The proof for the $\mathcal{U}$ operator is similar. ∎

5 Model Checking

Given a state s in G and a graded ATL formula φ, the model checking problem asks whether $s \models \varphi$. Notice that we only need to treat the $\langle\langle 1\rangle\rangle$ quantifier. The $\langle\langle \emptyset\rangle\rangle$ and $\langle\langle 1, 2\rangle\rangle$ quantifiers coincide with the $\forall$ and $\exists$ quantifiers of CTL, respectively. Therefore, they can be model-checked using the results of [FNP08]. Finally, the $\langle\langle 2\rangle\rangle$ quantifier is the dual of $\langle\langle 1\rangle\rangle$, and can therefore be evaluated using the algorithms developed for $\langle\langle 1\rangle\rangle$ on a game where the roles of Player 1 and Player 2 have been reversed.

In this section, we provide algorithms for solving a stronger form of model checking: we compute $grd^{x}(s, \varphi)$, that is the greatest k such that $s \models^{x} \langle\langle 1\rangle\rangle^{k}\theta$ holds, for a path formula θ and $x \in \{\text{on}, \text{off}\}$.

5.1 Off-line Semantics

Given a path formula $\theta = \Box q$ or $\theta = p\mathcal{U}q$, we describe an algorithm for computing $grd^{\text{off}}(s, \langle\langle 1\rangle\rangle\theta)$ for all states $s \in S$ (the $\bigcirc$ operator is a simple case). Then, we show how to solve the model checking problem using said algorithm. In the following, we say that a strongly connected component of a graph is a *sink* is there are no outgoing edges from it.

Lemma 4. *For each state s, Algorithm 1 computes $grd^{\text{off}}(s, \langle\langle 1\rangle\rangle\theta)$, for $\theta = \Box q$ 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 $\langle\langle 1\rangle\rangle\theta$, and removes states and moves which do not contribute to winning this game. Then, it computes in the new game the strongly connected

Algorithm 1 The algorithm computing $grd^{\text{off}}(\cdot, \langle\langle 1 \rangle\rangle\theta)$, given a path formula $\theta = \Box q$ or $\theta = p\mathcal{U}q$.

1. Using standard ATL algorithms, compute the set of states T where $\langle\langle 1 \rangle\rangle^1\theta$ holds, and assign 0 to the states in $S \setminus T$. Then, compute the corresponding sub-game, i.e., remove the states not in T and the moves of Player 1 that leave T. If $\theta = p\mathcal{U}q$, remove also all moves leaving the states where $\neg p$ holds.
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 having more than one state and containing a decision point are assigned grade ∞.
   (b) Sink components which do not fall in case 3a are assigned grade 1.
   (c) Non-sink components having more than one state and containing a decision point are assigned grade ∞.
   (d) Non-sink components which have more than one state and do not fall in case 3c are assigned ∞ if they have a successor component with grade greater than 1; otherwise, they are assigned 1.
   (e) Non-sink components containing only one state: if this state is of Player 1 then it is assigned the sum of the grades of the successor components; while if the state is of Player 2, then it is assigned the product of the grades of the successor components.

components (we assume that there exists at least one such component, otherwise the statement trivially holds).

The algorithm computes the greatest grade k for each state s, such that $s \models \langle\langle 1 \rangle\rangle^k\theta$. It is immediate to observe that all the states belonging to a strongly connected component have the same grade, as they all have the same number of strategies, thus the same value is assigned to the whole component.

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 contains a decision point, then the value ∞ is assigned to it, otherwise the value 1 is assigned. Let us prove that this is correct. Let r be a decision point and call r_1, r_2 two of its successors. Informally speaking, for all $h > 0$ there is a strategy of Player 1 that, in the state r, chooses to visit r_1 h times, before visiting r_2. More precisely, let $\alpha = r_1 \dots r$ be a finite path in G, define the strategy $\sigma_h = (\{1\}, f)$ where $f(\alpha^h) = r_2$ and for $j < h$, $f(\alpha^j) = r_1$. Clearly, for each $h > 0$, the strategies σ_i, $i \leq h$, determine the pairwise (θ, off)-dissimilar $\text{Outc}(r_2, \sigma_i)$ and thus, we have $r_2 \models^{\text{off}} \langle\langle 1 \rangle\rangle^k\theta$, for all $k > 0$. On the contrary, if there is not such a decision point, there is only one strategy in the connected component. Thus, steps 3a and 3b are correct and the correctness of step 3c can be derived in a similar way.

Consider now a non-sink component C having more than one state and not containing a decision point. If the algorithm has assigned 1 to all the successor components of C, then there is only one strategy for the team $\{1\}$. Otherwise, suppose that there is a state r in C of Player 2 having a successor r' in another component and that there exist two strategies of Player 1 starting from r'. Then, for any way of alternating these two strategies, whenever the state r' is entered, there is a strategy of Player 1 from r, and thus the algorithm correctly assigns grade infinite.

Case 3e refers to singleton connected components. The algorithm correctly assigns the values of $grd^{\text{off}}(\cdot, \langle\langle 1\rangle\rangle\theta)$ from Lemma 2.

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 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 4 as follows. Suppose that G has been model-checked against a given graded ATL formula φ. Then, to check whether $s \models^{\text{off}} \langle\langle 1\rangle\rangle^{\hat{k}} \Box\varphi$, for a given grade $\hat{k}$, Algorithm 1 can determine the greatest grade k such that $s \models \langle\langle 1\rangle\rangle^{k} \Box q_\varphi$ holds, for a new atomic proposition q_φ, holding true in each state r such that $r \models^{\text{off}} \varphi$. Similarly for the $\mathcal{U}$ operator. Thus, the following theorem holds.

Theorem 3. *Given a game $G = (S_1, S_2, \delta, [\cdot])$, a state s in G and a graded* ATL *formula φ, the graded model checking problem, $s \models^{\text{off}} \varphi$, can be solved in time $\mathcal{O}(|\delta| \cdot |\varphi|)$, where $|\varphi|$ is the number of operators occurring in φ.*

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 $grd^{\text{off}}(s, \langle\langle 1\rangle\rangle\theta)$ for $\theta = \Box q$ or $\theta = p\mathcal{U}q$, and for all states $s \in S$.

Algorithm 2 The algorithm for computing $grd^{\text{on}}(\cdot, \langle\langle 1\rangle\rangle\theta)$, given a path formula $\theta = \Box q$ or $\theta = p\mathcal{U}q$.

---

1. Using standard ATL algorithms, compute the set of states T where $\langle\langle 1\rangle\rangle^1\theta$ holds. The following steps are performed in the subgame with state-space T. In other words, the moves of Player 1 that leave T are removed. If $\theta = p\mathcal{U}q$, remove also all moves leaving the states where $\neg p$ holds. Assign grade 0 to the states in $S \setminus T$.
2. Let d be a new atomic proposition which holds in the decision points (of the subgame). Find the states where $\langle\langle 2\rangle\rangle^1\Box\neg d$ holds, and assign grade 1 to them.
3. Find the states where $\langle\langle 1\rangle\rangle^1\Box\Diamond d$ holds, and assign grade ∞ to them.
4. For the remaining states, compute their value by inductively applying equation (1) to those states whose successors have already been assigned a value.

Given a path formula $\theta = \Box q$ or $\theta = p\mathcal{U}q$, Algorithm 2 computes $grd^{\text{on}}(s, \langle\langle 1 \rangle\rangle \theta)$, for all states $s \in S$. The complexity of the algorithm is dominated by step 3, which involves the solution of a Büchi game [Tho95]. 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.

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. *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.

(a) First, we prove that all loops with no value contain a state which is labeled with d. If, during step 2, 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 $\langle\langle 2 \rangle\rangle^1 \Box \neg d$ and receive value 1 during step 2, which is a contradiction. Thus, at least one state in the loop is labeled with d (i.e., it is a decision point).

(b) It is clear that 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.

(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)* remain 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 $\langle\langle 1 \rangle\rangle^1 \Box \Diamond d$, which is a contradiction. Therefore, A must be empty and we obtain the thesis. ∎

Lemma 6. *Given a path formula $\theta = \Box q$ or $\theta = p\mathcal{U}q$, at the end of Algorithm 2, each state s has value $grd^{\text{on}}(s, \langle\langle 1 \rangle\rangle \theta)$. 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 $s \not\models^{\text{on}} \langle\langle 1 \rangle\rangle \theta$. Therefore, zero is indeed the largest integer k such that $s \models^{\text{on}} \langle\langle 1 \rangle\rangle^k \theta$ holds.

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

If s receives its value (infinity) during step 3, it means that $s \models^{\text{on}} \langle\!\langle 1 \rangle\!\rangle^1 \Box\Diamond d$. Consider any strategy τ of Player 2, and a strategy σ of Player 1 ensuring $\Box\Diamond d$. The resulting infinite path ρ contains infinitely many decision points for Player 1. For each decision point $\rho(i)$, let σ_i be a strategy of Player 1 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, $s \models^{\text{on}} \langle\!\langle 1 \rangle\!\rangle^k \theta$ holds for all $k > 0$.

Finally, if s receives its value during step 4, the correctness of the value is a consequence of Lemma 1. 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 4. *Given a game $G = (S_1, S_2, \delta, [\cdot])$, a state s in G and a graded* ATL *formula φ, the graded model checking problem, $s \models^{\text{on}} \varphi$, can be solved in time $\mathcal{O}(|S| \cdot |\delta| \cdot |\varphi|)$, where $|\varphi|$ is the number of operators occurring in φ.*

As before, under the constant-time assumption for basic integer operations, the above complexity is independent of the integer constants appearing in the formula.

6 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, according to the respective order of the two classical quantifiers hidden within each ATL team quantifier.

Given an ATL formula, an interesting computational question is then 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 a straightforward method for computing such value, namely Picard iteration: start with the lowest possible value and repeatedly apply the fixpoint operator. 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, we provide ad-hoc algorithms, that compute the maximum grade of a formula in polynomial time, avoiding the above-mentioned issues while still exploiting the fixpoint characterization.

Future work along these lines includes an investigation into the practical applications of these formalisms and algorithms, possibly in the field of fault-tolerant systems.

References

[ÅGJ07] T. Ågotnes, V. Goranko, and W. Jamroga. Alternating-time temporal logics with irrevocable strategies. In *TARK '07: Proceedings of the 11th conference on Theoretical aspects of rationality and knowledge*, pages 15–24. ACM, 2007.

[AHK02] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49:672–713, 2002.

[AHM+98] R. Alur, T.A. Henzinger, F.Y.C. Mang, S. Qadeer, S.K. Rajamani, and S. Tasiran. Mocha: modularity in model checking. In *CAV 98: Proc. of 10th Conf. on Computer Aided Verification*, volume 1427 of *Lect. Notes in Comp. Sci.*, pages 521–525. Springer-Verlag, 1998.

[CDL99] F. Corradini, R. De Nicola, and A. Labella. Graded modalities and resource bisimulation. In *Proceedings of FST/TCS'99*, volume 1738 of *Lect. Notes in Comp. Sci.*, pages 381–393. Springer-Verlag, 1999.

[Fin72] K. Fine. In so many possible worlds. *Notre Dame journal of Formal Logic*, 13(4):516–520, 1972.

[FMP08] A. Ferrante, A. Murano, and M. Parente. Enriched μ-calculi module checking. *Logical Methods in Computer Science*, 4(3), 2008.

[FNP08] A. Ferrante, M. Napoli, and M. Parente. Ctl model-checking with graded quantifiers. In *ATVA '08: Proc. of the 6th International Symposium on Automated Technology for Verification and Analysis.* (A preliminary version was presented at CILC '08*)*, volume 5311 of *Lect. Notes in Comp. Sci.*, pages 18–32, 2008.

[GOR97] E. Gradel, M. Otto, and E. Rosen. Two-variable logic with counting is decidable. In *Proc. 12th IEEE Symp. Logic in Comp. Sci.*, 1997.

[HB91] 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.

[KSV02] 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.

[LR06] A. Lomuscio and F. Raimondi. Mcmas: A model checker for multi-agent systems. In *TACAS 06: Tools and Algorithms for the Construction and Analysis of Systems*, volume 3920 of *Lect. Notes in Comp. Sci.*, pages 450–454. Springer-Verlag, 2006.

[Mar75] D.A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975.

[OR94] M.J. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, 1994.

[Tho95] W. Thomas. On the synthesis of strategies in infinite games. In *Proc. of 12th Annual Symp. on Theor. Asp. of Comp. Sci.*, volume 900 of *Lect. Notes in Comp. Sci.*, pages 1–13. Springer-Verlag, 1995.

[vdHW03] 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.