Reasoning about Quality and Fuzziness of Strategic Behaviours

Patricia Bouyer[1] , **Orna Kupferman**[2] , **Nicolas Markey**[3]
Bastien Maubert[4] , **Aniello Murano**[4] , **Giuseppe Perelli**[5]
[1] LSV, CNRS & ENS Paris-Saclay, Univ. Paris-Saclay, France
[2] Hebrew University, Israel
[3] Irisa, CNRS & Inria & Univ. Rennes, France
[4] Università degli Studi di Napoli "Federico II", Italy
[5] University of Leicester, UK

Abstract

We introduce and study $\mathsf{SL}[\mathcal{F}]$—a quantitative extension of SL (*Strategy Logic*), one of the most natural and expressive logics describing strategic behaviours. The satisfaction value of an $\mathsf{SL}[\mathcal{F}]$ formula is a real value in $[0, 1]$, reflecting "*how much*" or "*how well*" the strategic on-going objectives of the underlying agents are satisfied. We demonstrate the applications of $\mathsf{SL}[\mathcal{F}]$ in quantitative reasoning about multi-agent systems, by showing how it can express concepts of stability in multi-agent systems, and how it generalises some fuzzy temporal logics. We also provide a model-checking algorithm for our logic, based on a quantitative extension of Quantified $\mathsf{CTL}^\star$.

1 Introduction

One of the significant developments in formal reasoning has been the use of *temporal logics* for the specification of on-going behaviours of reactive systems [Pnueli, 1981; Emerson and Halpern, 1986]. The need to reason about multi-agent systems has led to the development of specification formalisms that enable the specification of on-going strategic behaviours in multi-agent systems. Essentially, these formalisms, most notably ATL, $\mathsf{ATL}^\star$ [Alur *et al.*, 2002] and Strategy Logic (SL) [Mogavero *et al.*, 2014], include quantification over strategies of the different agents, making it possible to specify concepts that have been traditionally studied in game theory.

While SL, which subsumes $\mathsf{ATL}^\star$, enables the specification of rich strategic behaviours in on-going multi-agent systems, its semantics is Boolean: a system may satisfy a specification or it may not. The Boolean nature of traditional temporal logic is a real obstacle in the context of strategic reasoning. Indeed, while many strategies may attain a desired objective, they may do so at different levels of quality or certainty. Consequently, designers would be willing to give up manual design only after being convinced that the automatic procedure that replaces it generates systems of comparable quality and certainty. For this to happen, one should first extend the specification formalism to one that supports quantitative aspects of the systems and the strategies.

The logic $\mathsf{LTL}[\mathcal{F}]$ is a multi-valued logic that augments LTL with quality operators [Almagor *et al.*, 2016]. The satisfaction value of an $\mathsf{LTL}[\mathcal{F}]$ formula is a real value in $[0, 1]$, where the higher the value, the higher the quality in which the computation satisfies the specification. The quality operators in $\mathcal{F}$ can prioritise different scenarios or reduce the satisfaction value of computations in which delays occur. For example, the set $\mathcal{F}$ may contain the $\min\{x, y\}$, $\max\{x, y\}$, and $1-x$ functions, which are the standard quantitative analogues of the $\wedge$, $\vee$, and $\neg$ operators. The novelty of $\mathsf{LTL}[\mathcal{F}]$ is the ability to manipulate values by arbitrary functions. For example, $\mathcal{F}$ may contain the weighted-average function $\oplus_\lambda$. The satisfaction value of the formula $\psi_1 \oplus_\lambda \psi_2$ is the weighted (according to λ) average between the satisfaction values of ψ_1 and ψ_2. This enables the specification of the quality of the system to interpolate different aspects of it.

$\mathsf{LTL}[\mathcal{F}]$ with functions $\wedge$, $\vee$ and $\neg$ also corresponds to Fuzzy Linear-time Temporal Logic (Fuzzy LTL) [Lamine and Kabanza, 2000; Frigeri *et al.*, 2014]. Note that by equipping $\mathcal{F}$ with adequate functions, we can capture various classic fuzzy interpretations of Boolean operators, such as the Zadeh, Gödel-Dummett or Łukasiewicz interpretations (see for instance [Frigeri *et al.*, 2014]). However the interpretation of the temporal operators is always that of Fuzzy LTL.

We introduce and study $\mathsf{SL}[\mathcal{F}]$, an analogous multi-valued extension of SL. In addition to the quantitative semantics that arises from the functions in $\mathcal{F}$, another important aspect of $\mathsf{SL}[\mathcal{F}]$ is that its semantics is defined with respect to *weighted* multi-agent systems, namely ones where atomic propositions have truth values in $[0, 1]$, reflecting quality or fuzziness. Thus, a model-checking procedure for $\mathsf{SL}[\mathcal{F}]$, which is our main contribution, enables formal reasoning about both quality and fuzziness of strategic behaviours.

As a motivating example, consider security drones that may patrol different height levels, and whose objectives are to fly above and below all uncontrollable drones and perform certain actions when uncontrollable drones exhibit some disallowed behaviour. In $\mathsf{SL}[\mathcal{F}]$ we can specify the quality of strategies for the drones: the different heights are specified with multi-valued atomic propositions, on-going behaviours are expressed with temporal operators and the functions in $\mathcal{F}$ may be used to refer to these behaviours in a quantitative manner, for example to compare heights and specify the satisfaction level of different possible scenarios. Note that the

$\mathsf{SL}[\mathcal{F}]$ formula does not merely specify the ability of the drones to behave in some desired manner, but rather it associates a satisfaction value in $[0,1]$ with each behaviour. For example (see Section 2.4), beyond specifying that the agents are in a Nash Equilibrium, we can specify how far they are from an equilibrium, namely how much an agent may gain by a deviation. As a result we can express and generalise concepts such as ε-Nash Equilibria [Nisan *et al.*, 2007].

We show that the quantification of strategies in $\mathsf{SL}[\mathcal{F}]$ can be reduced to a Boolean quantification of atomic propositions, which enables us to reduce the model-checking problem of $\mathsf{SL}[\mathcal{F}]$ to that of $\mathsf{BQCTL}^\star[\mathcal{F}]$, a logic that merges Quantified $\mathsf{CTL}^\star$ [Laroussinie and Markey, 2014] (an extension of $\mathsf{CTL}^\star$ with quantifiers on atomic propositions) with $\mathsf{CTL}^\star[\mathcal{F}]$ [Almagor *et al.*, 2016]. A general technique for $\mathsf{CTL}^\star$ model-checking algorithms is to repeatedly evaluate the innermost state subformula by viewing it as an (existentially or universally quantified) LTL formula, and add a fresh atomic proposition that replaces this subformula [Emerson and Lei, 1987]. This general technique is applied also to $\mathsf{CTL}^\star[\mathcal{F}]$, with the fresh atomic propositions being weighted [Almagor *et al.*, 2016]. For $\mathsf{BQCTL}^\star[\mathcal{F}]$ formulas, however, one cannot apply it. Indeed, the externally quantified atomic propositions may appear in different subformulas, and we cannot evaluate them one by one without fixing the same assignment for the quantified atomic propositions. Instead, we extend the automaton-theoretic approach to $\mathsf{CTL}^\star$ model-checking [Kupferman *et al.*, 2000] to handle quantified propositions: given a $\mathsf{BQCTL}^\star[\mathcal{F}]$ formula ψ and predicate $P \subseteq [0,1]$, we construct an alternating parity tree automaton that accepts exactly all the weighted trees t such that the satisfaction value of ψ in t is in P. The translation, and hence the complexity of the model-checking problem, is non-elementary: we show that it is $(k+1)$-EXPTIME-complete for formulas involving at most k nested quantifications on atomic propositions, and we show a similar complexity result for $\mathsf{SL}[\mathcal{F}]$, in terms of nesting of strategy quantifiers.

Related work. Various kinds of quantitative objectives have been studied in weighted games, both for two players (parity [Emerson and Jutla, 1991], mean-payoff [Ehrenfeucht and Mycielski, 1979] or energy [Bouyer *et al.*, 2008]) and n-player games (see e.g. [Bruyère *et al.*, 2014; Brenguier *et al.*, 2016; Gutierrez *et al.*, 2017]). Similarly, quantitative versions of LTL and CTL have been studied in different contexts, with discounting [Almagor *et al.*, 2014], averaging [Bollig *et al.*, 2012; Bouyer *et al.*, 2014], or richer constructs [Boker *et al.*, 2014; Almagor *et al.*, 2016]. In contrast, the study of quantitative temporal logics for strategic reasoning has remained rather limited: works on $\mathsf{LTL}[\mathcal{F}]$ include algorithms for synthesis and rational synthesis [Almagor *et al.*, 2016; 2017; 2018], but no logics combine the quantitative aspect of $\mathsf{LTL}[\mathcal{F}]$ with the strategic reasoning offered by SL.

A quantitative version of SL with Boolean goals over one-counter games has been considered in [Bouyer *et al.*, 2015]; only a periodicity property was proven, and no model-checking algorithm is known in that setting.

The other quantitative extensions we know of concern ATL/$\mathsf{ATL}^\star$, and most of the results are actually adaptations of similar (decidability) results for the corresponding extensions of CTL and $\mathsf{CTL}^\star$; this includes multi-valued ATL [Jamroga *et al.*, 2016] and weighted versions of ATL [Laroussinie *et al.*, 2006; Bulling and Goranko, 2013; Vester, 2015]. Finally, some works have considered non-quantitative ATL with quantitative constraints on the set of allowed strategies [Alechina *et al.*, 2010; Della Monica and Murano, 2018], proving decidability of the model-checking problem.

A full version of the present paper with detailed proofs is available as [Bouyer *et al.*, 2019].

2 Quantitative Strategy Logic

Let Σ be an alphabet. A *finite* (resp. *infinite*) *word* over Σ is an element of Σ^* (resp. Σ^ω). The *length* of a finite word $w = w_0 w_1 \dots w_n$ is $|w| := n+1$, and $\text{last}(w) := w_n$ is its last letter. Given a finite (resp. infinite) word w and $0 \leq i < |w|$ (resp. $i \in \mathbb{N}$), the word $w_{\leq i} = w_0 \dots w_i$ is the (non-empty) finite prefix of w that ends at position i and $w_{\geq i} = w_i w_{i+1} \dots$ is the suffix of w that starts at position i. We write $\mathsf{Pref}(w)$ for the set of non-empty prefixes of w. As usual, for any partial function f, we write $\mathsf{dom}(f)$ for the domain of f.

Strategy logic with functions, denoted $\mathsf{SL}[\mathcal{F}]$, generalises both SL and $\mathsf{LTL}[\mathcal{F}]$ by replacing the Boolean operators of SL with arbitrary functions over $[0,1]$. The logic is actually a family of logics, each parameterised by a set $\mathcal{F}$ of functions.

2.1 Syntax

We build on the branching-time variant of SL [Fijalkow *et al.*, 2018], which presents several benefits over classic SL, such as making the connection with Quantified $\mathsf{CTL}^\star$ tighter. Let $\mathcal{F} \subseteq \{f : [0,1]^m \to [0,1] \mid m \in \mathbb{N}\}$ be a set of functions.

Definition 1. *The syntax of* $\mathsf{SL}[\mathcal{F}]$ *is defined with respect to a finite set of* atomic propositions AP*, a finite set of agents* Agt *and a set of strategy variables* Var*. The set of* $\mathsf{SL}[\mathcal{F}]$ *formulas is defined by the following grammar:*

$$\varphi ::= p \mid \langle\!\langle x \rangle\!\rangle \varphi \mid (a,x)\varphi \mid \mathsf{A}\psi \mid f(\varphi,\dots,\varphi)$$
$$\psi ::= \varphi \mid \mathsf{X}\psi \mid \psi \mathsf{U} \psi \mid f(\psi,\dots,\psi)$$

where $p \in \mathsf{AP}$*,* $x \in \mathsf{Var}$*,* $a \in \mathsf{Agt}$*, and* $f \in \mathcal{F}$*.*

Formulas of type φ are called *state formulas*, those of type ψ are called *path formulas*. Formulas $\langle\!\langle x \rangle\!\rangle\varphi$ are called *strategy quantifications* whereas formulas $(a,x)\varphi$ are called *bindings*. Modalities X and U are temporal modalities, which take a specific quantitative semantics as we see below.

2.2 Semantics

While SL is evaluated on classic concurrent game structures with Boolean valuations for atomic propositions, $\mathsf{SL}[\mathcal{F}]$ formulas are interpreted on weighted concurrent game structures, in which atomic propositions have values in $[0,1]$.

Definition 2. *A* weighted concurrent game structure *(WCGS) is a tuple* $\mathcal{G} = (\mathsf{AP}, \mathsf{Agt}, \mathsf{Act}, V, v_\iota, \Delta, \mathsf{w})$ *where* AP *is a finite set of atomic propositions,* Agt *is a finite set of agents,* Act *is a finite set of actions,* V *is a finite set of states,* $v_\iota \in V$ *is an initial state,* $\Delta\colon V \times \mathsf{Act}^{\mathsf{Agt}} \to V$ *is the transition function, and* $\mathsf{w}\colon V \to [0,1]^{\mathsf{AP}}$ *is a weight function.*

An element of $\mathsf{Act}^{\mathsf{Agt}}$ is a *joint action*. For $\mathsf{v} \in \mathsf{V}$, we let $\mathsf{succ}(\mathsf{v})$ be the set $\{\mathsf{v}' \in \mathsf{V} \mid \exists \vec{c} \in \mathsf{Act}^{\mathsf{Agt}}.\ \mathsf{v}' = \Delta(\mathsf{v}, \vec{c})\}$.

A *play* in $\mathcal{G}$ is an infinite sequence $\pi = (\mathsf{v}_i)_{i \in \mathbb{N}}$ of states such that $\mathsf{v}_0 = \mathsf{v}_\iota$ and $\mathsf{v}_i \in \mathsf{succ}(\mathsf{v}_{i-1})$ for all $i > 0$. We write $\mathsf{Play}_\mathcal{G}$ for the set of plays in $\mathcal{G}$, and $\mathsf{Play}_\mathcal{G}(v)$ for the set of plays in $\mathcal{G}$ starting from v. Finite prefixes of plays are called *histories*, and we let $\mathsf{Hist}_\mathcal{G}(\mathsf{v}) = \mathsf{Pref}(\mathsf{Play}_\mathcal{G}(\mathsf{v}))$ and $\mathsf{Hist}_\mathcal{G} = \bigcup_{\mathsf{v} \in \mathsf{V}} \mathsf{Hist}_\mathcal{G}(\mathsf{v})$.

A *strategy* is a mapping $\sigma\colon \mathsf{Hist}_\mathcal{G} \to \mathsf{Act}$, and $\mathrm{Str}_\mathcal{G}$ is the set of strategies in $\mathcal{G}$. An *assignment* is a partial function $\chi\colon \mathsf{Var} \cup \mathsf{Agt} \rightharpoonup \mathrm{Str}_\mathcal{G}$, that assigns strategies to variables and agents. Assignment $\chi[a \mapsto \sigma]$ maps a to σ and is equal to χ otherwise. The set of *outcomes* of assignment χ from history ρ is the set $\mathsf{Out}(\chi, \rho)$ of plays $\pi = \rho \cdot \mathsf{v}_1 \mathsf{v}_2 \dots$ such that for every $i \in \mathbb{N}$, there exists a joint action $\vec{c} \in \mathsf{Act}^{\mathsf{Agt}}$ such that for each agent $a \in \mathsf{dom}(\chi), \vec{c}(a) = \chi(a)(\pi_{\leq |\rho|+i-1})$ and $\mathsf{v}_{i+1} = \Delta(\mathsf{v}_i, \vec{c})$, where $\mathsf{v}_0 = \mathrm{last}(\rho)$.

Definition 3. *Let $\mathcal{G}$ be a WCGS over AP and Agt, and χ an assignment for Agt and Var. The satisfaction value of an $\mathsf{SL}[\mathcal{F}]$ state formula φ in the last state of a history ρ with assignment χ, denoted $[\![\varphi]\!]^\mathcal{G}_\chi(\rho)$, and the satisfaction value of an $\mathsf{SL}[\mathcal{F}]$ path formula ψ in a play π starting at point $i \in \mathbb{N}$, denoted $[\![\varphi]\!]^\mathcal{G}_\chi(\rho)$, are defined inductively as follows:*

$$
\begin{aligned}
[\![p]\!]^\mathcal{G}_\chi(\rho) &= \mathsf{w}(\mathrm{last}(\rho))(p)\\
[\![\langle\!\langle x\rangle\!\rangle\varphi]\!]^\mathcal{G}_\chi(\rho) &= \sup_{\sigma \in \mathrm{Str}_\mathcal{G}} [\![\varphi]\!]^\mathcal{G}_{\chi[x\mapsto\sigma]}(\rho)\\
[\![(a,x)\varphi]\!]^\mathcal{G}_\chi(\rho) &= [\![\varphi]\!]^\mathcal{G}_{\chi[a\mapsto\chi(x)]}(\rho)\\
[\![\mathsf{A}\psi]\!]^\mathcal{G}_\chi(\rho) &= \inf_{\pi \in \mathsf{Out}(\chi,\rho)} [\![\psi]\!]^\mathcal{G}_\chi(\pi, |\rho|-1)\\
[\![f(\varphi_1,\dots,\varphi_m)]\!]^\mathcal{G}_\chi(\rho) &= f([\![\varphi_1]\!]^\mathcal{G}_\chi(\rho),\dots,[\![\varphi_m]\!]^\mathcal{G}_\chi(\rho))\\
[\![\varphi]\!]^\mathcal{G}_\chi(\pi,i) &= [\![\varphi]\!]^\mathcal{G}_\chi(\pi_{\leq i})\\
[\![\mathsf{X}\psi]\!]^\mathcal{G}_\chi(\pi,i) &= [\![\psi]\!]^\mathcal{G}_\chi(\pi,i+1)\\
[\![\psi_1\mathsf{U}\psi_2]\!]^\mathcal{G}_\chi(\pi,i) &= \sup_{j\geq i}\min\Big\{[\![\varphi_2]\!]^\mathcal{G}_\chi(\pi,j),\\
&\qquad \min_{k\in[i,j-1]}[\![\varphi_1]\!]^\mathcal{G}_\chi(\pi,k)\Big\}\\
[\![f(\psi_1,\dots,\psi_m)]\!]^\mathcal{G}_\chi(\pi,i) &= f([\![\psi_1]\!]^\mathcal{G}_\chi(\pi,i),\dots,[\![\psi_m]\!]^\mathcal{G}_\chi(\pi,i))
\end{aligned}
$$

That is, the satisfaction value of $\langle\!\langle x\rangle\!\rangle\varphi$ is the maximal value that a choice of strategy for variable x can give to φ. Binding $(a,x)\varphi$ just assigns strategy given by x to agent a. In the semantics of $\psi_1\mathsf{U}\psi_2$, while in the Boolean semantics we care about the first position in which ψ_2 holds, in the quantitative setting we maximise over all positions along the play, the minimum between the value of ψ_2 at that position and the minimal value of ψ_1 before this position.

We may use $\top$, $\vee$, and $\neg$ to denote functions 1, $\max$ and $1-x$, respectively. We can then define the following classic abbreviations: $\perp := \neg\top$, $\varphi\wedge\varphi' := \neg(\neg\varphi\vee\neg\varphi')$, $\varphi\to\varphi' := \neg\varphi\vee\varphi'$, $\mathsf{F}\psi := \top\mathsf{U}\psi$, $\mathsf{G}\psi := \neg\mathsf{F}\neg\psi$ and $[\![x]\!]\varphi := \neg\langle\!\langle x\rangle\!\rangle\neg\varphi$. When the values of the atomic propositions are in $\{0,1\}$, one can then check that all standard notations take their usual Boolean meaning.

Remark 1. *We will see that when the set of possible satisfaction values of atomic propositions is finite, as is the case when a model is chosen, then each formula has a finite set of possible satisfaction values. Therefore, the infima and suprema in the above definition are in fact minima and maxima.*

For a state formula φ and a WCGS $\mathcal{G}$, let $[\![\varphi]\!]^\mathcal{G} = [\![\varphi]\!]^\mathcal{G}_\emptyset(\mathsf{v}_\iota)$.

2.3 Model checking

The problem we are interested in is the following generalisation of the model checking problem, which is solved in [Almagor *et al.*, 2016] for $\mathsf{LTL}[\mathcal{F}]$ and $\mathsf{CTL}^\star[\mathcal{F}]$.

Definition 4. *Given an $\mathsf{SL}[\mathcal{F}]$ state formula φ, a WCGS $\mathcal{G}$ and a predicate $P \subseteq [0,1]$, decide whether $[\![\varphi]\!]^\mathcal{G} \in P$.*[1]

The precise complexity of the model-checking problem is stated in terms of *nesting depth* of formulas, which counts the maximal number $\mathrm{nd}(\varphi)$ of nested strategy quantifiers in a formula φ. We establish the following result in Section **??**:

Theorem 1. *The model-checking problem for $\mathsf{SL}[\mathcal{F}]$ formulas of nesting depth at most k is $(k+1)$-EXPTIME-complete.*

2.4 What can $\mathsf{SL}[\mathcal{F}]$ express?

$\mathsf{SL}[\mathcal{F}]$ naturally embeds SL. Indeed, if the values of the atomic propositions are in $\{0,1\}$ and $\mathcal{F} = \{\vee,\wedge,\neg\}$, then the satisfaction value of every $\mathsf{SL}[\mathcal{F}]$ formula φ is in $\{0,1\}$ and coincides with the semantics of φ seen as an SL formula. Also, the same way $\mathsf{ATL}^\star$ [Alur *et al.*, 2002] is embedded in SL, so is $\mathsf{ATL}^*[\mathcal{F}]$ embedded in $\mathsf{SL}[\mathcal{F}]$, where $\mathsf{ATL}^*[\mathcal{F}]$ is the natural quantitative extension of ATL^*.

Below we illustrate how quantities enable the specification of rich strategic properties.

Drone battle. A "carrier" drone c helped by a "guard" drone g try to bring an artefact to a rescue point and keep it away from the "villain" adversarial drone v. They evolve in a three dimensional cube of side length 1 unit, in which coordinates are triples $\vec{\gamma} = (\gamma_1,\gamma_2,\gamma_3) \in [0,1]^3$. We use the triples of atomic propositions $p_{\vec{\gamma}} = (p_{\gamma_1},p_{\gamma_2},p_{\gamma_3})$ and $q_{\vec{\gamma}} = (q_{\gamma_1},q_{\gamma_2},q_{\gamma_3})$ to denote the coordinates of c and v, respectively. Write $\mathsf{dist}\colon [0,1]^3\times[0,1]^3 \to [0,1]$ for the (normalized) distance between two points in the cube. Let the atomic proposition safe denote that the artefact has reached the rescue point. In $\mathsf{SL}[\mathcal{F}]$, we can express the level of safety for the artefact defined as the minimum distance between the carrier and the villain along a trajectory to reach the rescue point. Indeed, the formula

$$\varphi_{\text{rescue}} = \langle\!\langle x\rangle\!\rangle\langle\!\langle y\rangle\!\rangle(c,x)(g,y)\ \mathsf{A}(\mathsf{dist}(p_{\vec{\gamma}},q_{\vec{\gamma}})\mathsf{U}\ \mathsf{safe})$$

states that the carrier and guard drones cooperate to keep the villain as far as possible from the artefact, until it is rescued. Note that the satisfaction value of the $\mathsf{LTL}[\mathcal{F}]$ specification is 0 if there is a path in which the artifact is never rescued.

Synthesis with quantitative objectives. In LTL reactive synthesis [Pnueli and Rosner, 1989], a controller and an environment operate on two disjoint sets of variables in the system. The problem consists in synthesising a strategy for the controller such that, no matter the behaviour of the environment, the resulting execution satisfies ψ. Recently, this problem has been addressed in the context of $\mathsf{LTL}[\mathcal{F}]$, where the

[1] Predicate P is finitely represented, typically as an interval.

controller aims at maximising the value of an $\mathsf{LTL}[\mathcal{F}]$ formula φ, while the environment acts as minimiser. This problem can be expressed in $\mathsf{SL}[\mathcal{F}]$ with the formula

$$\varphi_{\texttt{synt}} = \langle\!\langle x \rangle\!\rangle\ (c, x)\ \mathsf{A}\psi$$

where c and e are the controller and environment, and ψ is the specification in $\mathsf{LTL}[\mathcal{F}]$.

We mention (but do not develop, by lack of space) that we can also express the variant of reactive synthesis called *rational synthesis* [Kupferman *et al.*, 2016]. This problem considers an environment composed of several rational components, and aims at synthesising a strategy that ensures the specification against all (or some) rational behaviours of the environment, where rational is characterised by some game-theoretic concept such as Nash equilibrium of subgame-perfect equilibrium. Our expressive logic allows us to deal with richer objectives than in [Almagor *et al.*, 2018].

Nash Equilibria in weighted games.

An important feature of SL is that it captures Nash equilibria (NE, for short) and other common solution concepts. This extends to $\mathsf{SL}[\mathcal{F}]$, but in a much stronger sense: first, objectives in $\mathsf{SL}[\mathcal{F}]$ are quantitative, so that *profitable deviation* is not a simple Boolean statement; second, the semantics of the logic is quantitative, so that *being a NE* is a quantitative property, and we can actually express *how far* a strategy profile is from being a NE.

A strategy profile $(\sigma_i)_{a_i \in \mathsf{Agt}}$ is a NE if all agents play one of their best responses against their opponents' strategies. Consider formula

$$\varphi_{\mathsf{NE}} = (a_1, x_1)...(a_n, x_n)\Big(\bigwedge_{a_i \in \mathsf{Agt}} [\![y_i]\!]\ ((a_i, y_i)\mathsf{A}\varphi_i) \preceq \mathsf{A}\varphi_i\Big)$$

where $\alpha \preceq \beta$ is actually a function $\preceq\colon [0,1]^2 \to [0,1]$ such that $\preceq(\alpha,\beta) = 1$ if $\alpha \le \beta$ and $\preceq(\alpha,\beta) = 0$ otherwise. Then $[\![\varphi_{\mathsf{NE}}]\!]^{\mathcal{G}}_{\chi}(\mathsf{v}) = 1$ if, and only if, $(\chi(x_i))_{a_i \in \mathsf{Agt}}$ form a NE from position v.

Adopting a more quantitative point of view, we can measure how much agent i can benefit from a selfish deviation using formula $\langle\!\langle y_i \rangle\!\rangle \mathsf{diff}((a_i, y_i)\varphi_i, \varphi_i)$, where $\mathsf{diff}(x, y) = \max\{0, x - y\}$. The maximal benefit that some agent may get is then captured by the following formula:

$$\varphi_{\overline{\mathsf{NE}}} = \langle\!\langle y \rangle\!\rangle\ (a_1, x_1)...(a_n, x_n)\Big(\bigvee_{a_i \in \mathsf{Agt}} \mathsf{diff}((a_i, y)\mathsf{A}\varphi_i, \mathsf{A}\varphi_i)\Big).$$

Formula $\varphi_{\overline{\mathsf{NE}}}$ can be used to characterise ε-NE, by requiring that $\varphi_{\overline{\mathsf{NE}}}$ has value less than or equal to ε; of course it also characterises classical NE as a special case.

Secure equilibria in weighted games.

Secure equilibria [Chatterjee *et al.*, 2006] are special kinds of NEs in two-player games, where besides improving their objectives, the agents also try to harm their opponent. Following the ideas above, we characterise secure equilibria in $\mathsf{SL}[\mathcal{F}]$ as follows:

$$\varphi_{\mathsf{SE}} = (a_1, x_1)(a_2, x_2) \bigwedge_{i \in \{1,2\}} [\![y_i]\!]((a_i, y_i)\mathsf{A}\varphi_1, (a_i, y_i)\mathsf{A}\varphi_2) \preceq_i (\mathsf{A}\varphi_1, \mathsf{A}\varphi_2)$$

where $(\alpha_1, \alpha_2) \preceq_i (\beta_1, \beta_2)$ is equal to 1 when $\alpha_i \le \beta_i$ or both $\alpha_i = \beta_i$ and $\alpha_{3-i} \le \beta_{3-i}$, and 0 otherwise.

Secure equilibria have also been studied in $\mathbb{Q}$-weighted games [Bruyère *et al.*, 2014]: in that setting, the objective of the agents is to optimise e.g. the (limit) infimum or supremum of the sequence of weights encountered along the play. We can characterise secure equilibria in such setting (after first applying an affine transformation to have all weights in $[0, 1]$): indeed, assuming that weights are encoded as the value of atomic proposition w, the value of formula $\mathsf{G}w$ is the infimum of the weights, while the value of $\mathsf{FG}w$ is the limit infimum. We can then characterise secure equilibria with (limit) infimum and supremum objectives by using those formulas as the objectives for the agents in formula φ_{SE}.

3 Booleanly Quantified $\mathsf{CTL}^\star[\mathcal{F}]$

In this section we introduce Booleanly Quantified $\mathsf{CTL}^\star[\mathcal{F}]$ ($\mathsf{BQCTL}^\star[\mathcal{F}]$, for short) which extends both $\mathsf{CTL}^\star[\mathcal{F}]$ and $\mathsf{QCTL}^\star$ [Laroussinie and Markey, 2014].

3.1 Syntax

Let $\mathcal{F} \subseteq \{f\colon [0,1]^m \to [0,1] \mid m \in \mathbb{N}\}$ be a set of functions.

Definition 5. *The syntax of* $\mathsf{BQCTL}^\star[\mathcal{F}]$ *is defined with respect to a finite set* AP *of atomic propositions, using the following grammar:*

$$\varphi ::= p \mid \exists p.\, \varphi \mid \mathsf{E}\psi \mid f(\varphi, \ldots, \varphi)$$
$$\psi ::= \varphi \mid \mathsf{X}\psi \mid \psi \mathsf{U} \psi \mid f(\psi, \ldots, \psi)$$

where p *ranges over* AP *and* f *over* $\mathcal{F}$.

Formulas of type φ are called *state formulas*, those of type ψ are called *path formulas*, and $\mathsf{BQCTL}^\star[\mathcal{F}]$ consists of all state formulas. We again use $\top$, $\vee$, and $\neg$ to denote functions 1, $\max$ and $1 - x$, as well as classic abbreviations already introduced for $\mathsf{SL}[\mathcal{F}]$, plus $\mathsf{A}\psi := \neg\mathsf{E}\neg\psi$.

3.2 Semantics

While $\mathsf{BQCTL}^\star[\mathcal{F}]$ formulas are interpreted over weighted Kripke structures, thus with atomic propositions having values in $[0, 1]$, the possible assignment for the quantified propositions are Boolean, hence the name of the logic.

Definition 6. *A* weighted Kripke structure *(WKS) is a tuple* $\mathcal{K} = (\mathsf{AP}, S, \mathsf{s}_\iota, R, \mathsf{w})$ *where* AP *is a finite set of atomic propositions,* S *is a finite set of states,* $\mathsf{s}_\iota \in S$ *is an initial state,* $R \subseteq S \times S$ *is a left-total transition relation, and* $\mathsf{w}\colon S \to [0,1]^{\mathsf{AP}}$ *is a weight function.*

A *path* in $\mathcal{K}$ is an infinite word $\pi = \pi_0\pi_1 \ldots$ over S such that $\pi_0 = \mathsf{s}_\iota$ and $(\pi_i, \pi_{i+1}) \in R$ for all i. We call finite prefixes of paths *histories*, and $\mathsf{Hist}_{\mathcal{K}}$ is the set of all histories in $\mathcal{K}$. We also let $V_{\mathcal{K}} = \{\mathsf{w}(\mathsf{s})(p) \mid \mathsf{s} \in \mathsf{S} \text{ and } p \in \mathsf{AP}\}$ be the finite set of values appearing in $\mathcal{K}$.

Given finite sets D of directions, AP of atomic propositions, and $V \subseteq [0, 1]$ of possible values, an (AP, V)-labelled D-tree, (or tree for short when the parameters are understood), is a pair $t = (\tau, \mathsf{w})$ where $\tau \subseteq D^+$ is closed under non-empty prefixes, all *nodes* $u \in \tau$ start with the same direction r, called the *root*, and have at least one *child* $u{\cdot}d \in \tau$, and

$\mathsf{w} : \tau \to V^{\mathsf{AP}}$ is a *weight function*. A *branch* $\lambda = u_0 u_1 \dots$ is an infinite sequence of nodes such that for all $i \geq 0$, we have that u_{i+1} is a child of u_i. We let $\mathrm{Br}(u)$ be the set of branches that start in node u. We say that a tree $t = (\tau, \mathsf{w})$ is *Boolean in* p, written $\mathrm{Bool}(t, p)$, if for all $u \in \tau$ we have $\mathsf{w}(u)(p) \in \{0, 1\}$. As with weighted Kripke structures, we let $V_t = \{\mathsf{w}(u)(p) \mid u \in \tau \text{ and } p \in \mathsf{AP}\}$.

Given two (AP, V)-labelled D-trees t, t' and $p \in \mathsf{AP}$, we write $t \equiv_p t'$ if t and t' differ only in assignments to p, which must be Boolean in t': formally, $t = (\tau, \mathsf{w})$, $t' = (\tau, \mathsf{w}')$, for the same domain τ, $\mathrm{Bool}(t', p)$, and for all $p' \in \mathsf{AP}$ such that $p' \neq p$ and all $u \in \tau$, we have $\mathsf{w}(u)(p') = \mathsf{w}'(u)(p')$.

Finally, the *tree unfolding of a WKS* $\mathcal{K}$ over atomic propositions AP and states S is the $(\mathsf{AP}, V_{\mathcal{K}})$-labelled S-tree $t_{\mathcal{K}} = (\mathsf{Hist}_{\mathcal{K}}, \mathsf{w}')$, where $\mathsf{w}'(u) = \mathsf{w}(\mathrm{last}(u))$ for every $u \in \mathsf{Hist}_{\mathcal{K}}$.

Definition 7. *Given a tree t, the satisfaction value $[\![\varphi]\!]^t(u)$ of a $\mathsf{BQCTL}^\star[\mathcal{F}]$ state formula φ in node u of t, and the satisfaction value $[\![\psi]\!]^t(\lambda)$ of a $\mathsf{BQCTL}^\star[\mathcal{F}]$ path formula ψ along a path λ of t, are defined inductively as follows:*

$$
\begin{aligned}
[\![p]\!]^t(u) &= \mathsf{w}(u)(p) \\
[\![\exists p.\, \varphi]\!]^t(u) &= \sup_{t' \mid t \equiv_p t'} [\![\varphi]\!]^{t'}(u) \\
[\![\mathsf{E}\psi]\!]^t(u) &= \sup_{\lambda \in Br(u)} [\![\psi]\!]^t(\lambda) \\
[\![f(\varphi_1, ..., \varphi_n)]\!]^t(u) &= f([\![\varphi_1]\!]^t(u), ..., [\![\varphi_n]\!]^t(u)) \\
[\![\varphi]\!]^t(\lambda) &= [\![\varphi]\!]^t(\lambda_0) \\
[\![\mathsf{X}\psi]\!]^t(\lambda) &= [\![\psi]\!]^t(\lambda_{\geq 1}) \\
[\![\psi_1 \mathsf{U} \psi_2]\!]^t(\lambda) &= \sup_{i \geq 0} \min([\![\psi_2]\!]^t(\lambda_{\geq i}), \min_{0 \leq j < i} [\![\psi_1]\!]^t(\lambda_{\geq j})) \\
[\![f(\psi_1, ..., \psi_n)]\!]^t(\lambda) &= f([\![\psi_1]\!]^t(\lambda), ..., [\![\psi_n]\!]^t(\lambda))
\end{aligned}
$$

Remark 2. *As with $\mathsf{SL}[\mathcal{F}]$, the suprema in the above definition can be replaced with maxima (see Lemma 3 below).*

First, note that if $\mathcal{F} = \{\top, \vee, \neg\}$, then $\mathsf{BQCTL}^\star[\mathcal{F}]$ evaluated on Boolean Kripke structures corresponds to classic $\mathsf{QCTL}^\star$. Note also that the quantifier on propositions does not range over arbitrary values in $[0, 1]$. Instead, as in $\mathsf{QCTL}^\star$, it quantifies only on Boolean valuations. It is still quantitative though, in the sense that instead of merely stating the existence of a valuation, $\exists p.\, \varphi$ maximises the value of φ over all possible (Boolean) valuations of p.

For a tree t with root r we write $[\![\varphi]\!]^t$ for $[\![\varphi]\!]^t(r)$, and for a WKS $\mathcal{K}$ we write $[\![\varphi]\!]^{\mathcal{K}}$ for $[\![\varphi]\!]^{t_{\mathcal{K}}}$. Note that this semantics is an extension of the tree semantics of $\mathsf{QCTL}^\star$, in which the valuation of quantified atomic propositions is chosen on the unfolding of the Kripke structure instead of the states. This allows us to capture the semantics of SL based on strategies with perfect recall, where moves can depend on the history, as opposed to the memoryless semantics, where strategies can only depend on the current state (see [Laroussinie and Markey, 2014] for more details).

We study the following model-checking problem:

Definition 8. *Given a $\mathsf{BQCTL}^\star[\mathcal{F}]$ state formula φ, a WKS $\mathcal{K}$ and a predicate $P \subseteq [0, 1]$, decide whether $[\![\varphi]\!]^{\mathcal{K}} \in P$.*

Similarly to $\mathsf{SL}[\mathcal{F}]$, the precise complexity of the model-checking problem will be stated in terms of *nesting depth* of formulas, which counts the maximal number of nested propositional quantifiers in a formula φ, and is written $\mathrm{nd}(\varphi)$.

In the next section we establish our main technical contribution, which is the following:

Theorem 2. *The model-checking problem for $\mathsf{BQCTL}^\star[\mathcal{F}]$ formulas of nesting depth k is $(k+1)$-*EXPTIME*-complete.*

This result, together with a reduction from $\mathsf{SL}[\mathcal{F}]$ to $\mathsf{BQCTL}^\star[\mathcal{F}]$ that we present in Section **??**, entails the decidability of model checking $\mathsf{SL}[\mathcal{F}]$ announced in Theorem 1.

4 Model checking

We start by proving that, as for $\mathsf{LTL}[\mathcal{F}]$, when the set of possible values of atomic propositions is finite, so is the set of possible satisfaction values of each $\mathsf{BQCTL}^\star[\mathcal{F}]$ formula. This property allows to use $\max$ instead of $\sup$ in Definition 7.

Lemma 3. *Let $V \subset [0, 1]$ be a finite set of values with $\{0, 1\} \subseteq V$ and φ a $\mathsf{BQCTL}^\star[\mathcal{F}]$ state formula. Define*

$$V_\varphi = \{[\![\varphi]\!]^t(u) \mid t \text{ is a } (\mathsf{AP}, V)\text{-labelled tree and } u \in t\}$$

be the set of values taken by φ in nodes of (AP, V)-labelled trees. Then, $|V_\varphi| \leq |V|^{|\varphi|}$, and one can compute $\widetilde{V}_\varphi$ of size at most $|V|^{|\varphi|}$ such that $V_\varphi \subseteq \widetilde{V}_\varphi$.

The finite over-approximation of the set of possible satisfaction values induces a finite alphabet for the automata our model-checking procedure uses.

In the following, we use *alternating parity tree automata* (APT in short), and their purely non-deterministic (resp. universal) variants, denoted NPT (resp. UPT). Given two APT $\mathcal{A}$ and $\mathcal{A}'$ we denote $\mathcal{A} \wedge \mathcal{A}'$ (resp. $\mathcal{A} \vee \mathcal{A}'$) the APT of size linear in $|\mathcal{A}|$ and $|\mathcal{A}'|$ that accepts the intersection (resp. union) of the languages of $\mathcal{A}$ and $\mathcal{A}'$, and we call *index* of an automaton the number of priorities in its parity condition. We refer the reader to [Kupferman and Vardi, 2005] for a detailed exposition of alternating parity tree automata.

We extend the automata construction from [Kupferman *et al.*, 2000] to handle quantification on atomic propositions and the quantitative aspect of our model-checking problem.

Proposition 4. *Let $V \subset [0, 1]$ be a finite set of values such that $\{0, 1\} \subseteq V$, and let D be a finite set of directions. For every $\mathsf{BQCTL}^\star[\mathcal{F}]$ state formula φ and predicate $P \subseteq [0, 1]$, one can construct an APT $\mathcal{A}_\varphi^{V,P}$ such that for every (AP, V)-labelled D-tree t, $\mathcal{A}_\varphi^{V,P}$ accepts t if and only if $[\![\varphi]\!]^t \in P$.*

The APT $\mathcal{A}_\varphi^{V,P}$ has at most $(nd(\varphi)+1)$-exponentially many states, and its index is at most $nd(\varphi)$-exponential.

Proof sketch. We sketch how the two interesting inductive cases are dealt with.

When φ is of the form $\exists p.\, \varphi'$, the automaton has to check that the maximal satisfaction value of φ' for all possible Boolean valuations of p is in P. First, compute finite set $\widetilde{V}_{\varphi'}$ from Lemma 3. Then for each possible $v \in \widetilde{V}_{\varphi'} \cap P$, we build an automaton for checking that the value of φ' is less than v for all p-valuations, and that value v is reached for some p-valuation. Inductively build the APTs $\mathcal{A}_{\varphi'}^{V,\{v\}}$ and $\mathcal{A}_{\varphi'}^{V,[0,v]\cap P}$.

Turn the first one into a NPT $\mathcal{N}_{=v}$ and the second one into a UPT $\mathcal{U}_{\leq v}$. Project $\mathcal{N}_{=v}$ existentially on p, and call the result $\mathcal{N}'_{=v}$. Project $\mathcal{U}_{\leq v}$ universally on p, call the result $\mathcal{U}'_{\leq v}$. Finally, define the APT $\mathcal{A}^{V,P}_{\exists p.\,\varphi'} := \bigvee_{v\in\widetilde{V}_{\varphi'}\cap P}\mathcal{N}'_{=v}\wedge\mathcal{U}'_{\leq v}$. This automaton accepts a tree if, and only if, there exists a value in P that is the maximum of the possible values taken by φ' for all p-valuations.

When φ is of the form $\mathsf{E}\psi$, first let $\mathsf{atoms}(\psi)$ be the set of maximal state sub-formulas of ψ (called *atoms*). In a first step we see ψ as an $\mathsf{LTL}[\mathcal{F}]$ formula over $\mathsf{atoms}(\psi)$. For each $\varphi'\in\mathsf{atoms}(\psi)$, compute the set $\widetilde{V}_{\varphi'}$ from Lemma 3, and let $\widetilde{V}=\bigcup_{\varphi'\in\mathsf{atoms}(\psi)}\widetilde{V}_{\varphi'}$. Build a non-deterministic parity automaton $\mathcal{W}^{\psi}_{P}$ that accepts the set of words $w\in(\widetilde{V}^{\mathsf{atoms}(\psi)})^{\omega}$ such that $[\![\psi]\!](w)\in P$ [Almagor *et al.*, 2016]. Then compute $\widetilde{V}_{\mathsf{E}\psi}$ (again using Lemma 3), and for each $v\in\widetilde{V}_{\mathsf{E}\psi}\cap P$, construct an NPT $\mathcal{N}^{E=v}$ that guesses a branch in its input and simulates $\mathcal{W}^{\psi}_{\{v\}}$ on it. Dually, build a universal word automaton $\mathcal{W}^{\psi}_{[0,v]}$ and then a UPT $\mathcal{U}^{A\leq v}$ that executes $\mathcal{W}^{\psi}_{[0,v]}$ on all branches of its input. Finally, define the APT $\mathcal{A}^{P}$ on $\widetilde{V}^{\mathsf{atoms}(\psi)}$-trees as $\mathcal{A}^{P}=\bigvee_{v\in\widetilde{V}_{\mathsf{E}\psi}\cap P}\mathcal{N}^{E=v}\wedge\mathcal{U}^{A\leq v}$.

To go from atoms to atomic propositions, define an APT $\mathcal{A}^{V,P}_{\mathsf{E}\psi}$ that, in each state and each node of its input, guesses a value v_i in $\widetilde{V}_{\varphi_i}$ for each $\varphi_i\in\mathsf{atoms}(\psi)$, simulates $\mathcal{A}^{P}$ on the resulting label, and launches a copy of $\mathcal{A}^{V,\{v_i\}}_{\varphi_i}$ for each $\varphi_i\in\mathsf{atoms}(\psi)$. Note that the automaton is alternating and thus may have to guess several times the satisfaction value of a formula φ_i in a same node, but launching the $\mathcal{A}^{V,\{v_i\}}_{\varphi_i}$ forces it to always guess the same, correct value. □

To check whether $[\![\varphi]\!]^{\mathcal{K}}\in P$, where atomic propositions in $\mathcal{K}$ take values in V, it is enough to build $\mathcal{A}^{V,P}_{\varphi}$ as in Proposition 4, take its product with a deterministic tree automaton that accepts only $t_{\mathcal{K}}$, and check for emptiness of the product automaton. The formula complexity is $(\mathrm{nd}(\varphi)+1)$-exponential, but the structure complexity is polynomial. The time complexity is $(\mathrm{nd}(\varphi)+1)$-exponential in the size of φ, but it is polynomial in the size of the game structure.

The lower bounds are obtained by reduction from the model-checking problem for $\mathsf{EQ}^k\mathsf{CTL}^\star$, a fragment of $\mathsf{QCTL}^\star$ that consists in formulas in prenex normal form with at most k alternations between existential and universal quantifiers. This problem is $(k+1)$-EXPTIME-hard [Laroussinie and Markey, 2014], and clearly, $\mathsf{EQ}^k\mathsf{CTL}^\star$ can be translated in $\mathsf{BQCTL}^\star[\mathcal{F}]$ with formulas of linear size and nesting depth at most k (alternation is simply coded by inserting function $\neg$ between quantifiers).

The usual reduction for qualitative variants of SL (see e.g. [Laroussinie and Markey, 2015; Berthon *et al.*, 2017; Fijalkow *et al.*, 2018]) can be lifted to the quantitative setting in a straightforward manner: for each instance $(\mathcal{G},\varphi,P)$ of the model-checking problem for $\mathsf{SL}[\mathcal{F}]$, we build a WKS $\mathcal{K}_{\mathcal{G}}$ and a $\mathsf{BQCTL}^\star[\mathcal{F}]$ formula φ' such that $[\![\varphi]\!]^{\mathcal{G}}=[\![\varphi']\!]^{\mathcal{K}_{\mathcal{G}}}$.

The model-checking problem for $(\mathcal{G},\phi,P)$ is then solved by deciding whether $[\![\varphi']\!]^{\mathcal{K}_{\mathcal{G}}}\in P$, which can be done by Theorem 2. This establishes the upper-bounds in Theorem 1.

As in the case of $\mathsf{BQCTL}^\star[\mathcal{F}]$, the lower-bounds are obtained by reduction from the model-checking problem for $\mathsf{EQ}^k\mathsf{CTL}^\star$. This reduction is an adaptation of the one from $\mathsf{QCTL}^\star$ to ATL with strategy context in [Laroussinie and Markey, 2015], and that preserves nesting depth.

Remark 3. *This reduction and Lemma 3 imply that when we fix possible values for atomic propositions, then possible values for an* $\mathsf{SL}[\mathcal{F}]$ *formula are at most exponentially many.*

We now consider the fragment $\mathsf{SL}_{\mathsf{1G}}[\mathcal{F}]$ of $\mathsf{SL}[\mathcal{F}]$, in which all formulas have the form $\wp\,\flat\,\psi$ where $\wp$ only contains strategy quantifiers, $\flat$ binds strategies to all agents, and ψ is an $\mathsf{LTL}[\mathcal{F}]$ formula (possibly involving closed $\mathsf{SL}_{\mathsf{1G}}[\mathcal{F}]$ subformulas as subformulas). It is the quantitative extension of $\mathsf{SL}_{\mathsf{1G}}$, the one-goal fragment of SL [Mogavero *et al.*, 2014].

As in the Boolean case, where $\mathsf{SL}_{\mathsf{1G}}$ strictly subsumes $\mathsf{ATL}^\star$ while enjoying the same elementary complexity for the model-checking problem, in the quantitative setting also $\mathsf{SL}_{\mathsf{1G}}[\mathcal{F}]$ strictly subsumes $\mathsf{ATL}^\star[\mathcal{F}]$, and we have:

Theorem 5. *The model-checking problem for* $\mathsf{SL}_{\mathsf{1G}}[\mathcal{F}]$ *is decidable, and* 2*-EXPTIME-complete.*

5 Discussion

We introduced and studied $\mathsf{SL}[\mathcal{F}]$, a formalism for specifying quality and fuzziness of strategic on-going behaviour. Beyond the applications described in the paper, we highlight here some interesting directions for future research. In classical temporal-logic model checking, *coverage* and *vacuity* algorithms measure the sensitivity of the system and its specifications to mutations, revealing errors in the modelling of the system and lack of exhaustiveness of the specification [Chockler *et al.*, 2006]. When applied to $\mathsf{SL}[\mathcal{F}]$, these algorithms can set the basis to a formal reasoning about classical notions in game theory, like the sensitivity of utilities to price changes, the effectiveness of burning money [Hartline and Roughgarden, 2008; Souza and Rego, 2016] or tax increase [Cole *et al.*, 2006], and more. Recall that our $\mathsf{SL}[\mathcal{F}]$ model-checking algorithm reduces the problem to $\mathsf{BQCTL}^\star[\mathcal{F}]$, where the quantified atomic propositions take Boolean values. It is interesting to extend $\mathsf{BQCTL}^\star[\mathcal{F}]$ to a logic in which the quantified atomic propositions are associated with different agents, which would enable easy specification of controllable events. Also, while in our application the quantified atomic propositions encode the strategies, and hence the restriction of their values to $\{0,1\}$ is natural, it is interesting to study $\mathsf{QCTL}^\star[\mathcal{F}]$, where quantified atomic propositions may take values in $[0,1]$.

Acknowledgements

Perelli thanks the support of the project “dSynMA”, funded by the ERC under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 772459). Bouyer and Markey thank the support of the ERC Project EQualIS (308087).

References

[Alechina *et al.*, 2010] N. Alechina, B. Logan, N. Nguyen Hoang, and A. Rakib. Resource-bounded alternating-time temporal logic. *AAMAS'10*, p. 481–488.

[Almagor *et al.*, 2014] S. Almagor, U. Boker, and O. Kupferman. Discounting in LTL. *TACAS'14*, LNCS 8413, p. 424–439. Springer.

[Almagor *et al.*, 2016] S. Almagor, U. Boker, and O. Kupferman. Formally reasoning about quality. *J.ACM*, 63(3):24:1–24:56.

[Almagor *et al.*, 2017] S. Almagor, O. Kupferman, J. Ringert, and Y. Velner. Quantitative assume guarantee synthesis. *CAV'17*, LNCS 10427, p. 353–374. Springer.

[Almagor *et al.*, 2018] S. Almagor, O. Kupferman, and G. Perelli. Synthesis of controllable Nash equilibria in quantitative objective games. *IJCAI'18*, p. 35–41.

[Alur *et al.*, 2002] R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J.ACM*, 49(5):672–713.

[Berthon *et al.*, 2017] R. Berthon, B. Maubert, A. Murano, S. Rubin, and M. Y. Vardi. Strategy logic with imperfect information. *LICS'17*, p. 1–12.

[Boker *et al.*, 2014] U. Boker, K. Chatterjee, T. Henzinger, and O. Kupferman. Temporal specifications with accumulative values. *ACM ToCL*, 15(4):27:1–27:25.

[Bollig *et al.*, 2012] B. Bollig, N. Decker, and M. Leucker. Frequency linear-time temporal logic. *TASE'12*, p. 85–92. IEEE.

[Bouyer *et al.*, 2008] P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, and J. Srba. Infinite runs in weighted timed automata with energy constraints. *FORMATS'08*, LNCS 5215, p. 33–47. Springer.

[Bouyer *et al.*, 2014] P. Bouyer, N. Markey, and R. M. Matteplackel. Averaging in LTL. *CONCUR'14*, LNCS 8704, p. 266–280. Springer.

[Bouyer *et al.*, 2015] P. Bouyer, P. Gardy, and N. Markey. Weighted strategy logic with boolean goals over one-counter games. *FSTTCS'15*, LIPIcs 45, p. 69–83. LZI.

[Bouyer *et al.*, 2019] P. Bouyer, O. Kupferman, N. Markey, B. Maubert, A. Murano, and G. Perelli. Reasoning about quality and fuzziness of strategic behaviours. Research Report 1905.11537, arXiv.

[Brenguier *et al.*, 2016] R. Brenguier, G. Pérez, J.-F. Raskin, and M. Sassolas. Admissibility in quantitative graph games. *FSTTCS'16*, LIPIcs 65, p. 42:1–42:14. LZI.

[Bruyère *et al.*, 2014] V. Bruyère, N. Meunier, and J.-F. Raskin. Secure equilibria in weighted games. *CSL/LICS'14*, p. 26:1–26:26. ACM.

[Bulling and Goranko, 2013] N. Bulling and V. Goranko. How to be both rich and happy: Combining quantitative and qualitative strategic reasoning about multi-player games (extended abstract). *SR'13*, EPTCS 112, p. 33–41.

[Chatterjee *et al.*, 2006] K. Chatterjee, T. Henzinger, and M. Jurdziński. Games with secure equilibria. *TCS*, 365(1-2):67–82.

[Chockler *et al.*, 2006] H. Chockler, O. Kupferman, and M. Y. Vardi. Coverage metrics for formal verification. *STTT*, 8(4-5):373–386.

[Cole *et al.*, 2006] R. Cole, Y. Dodis, and T. Roughgarden. How much can taxes help selfish routing? *JCSS*, 72(3):444–467.

[Della Monica and Murano, 2018] D. Della Monica and A. Murano. Parity-energy ATL for qualitative and quantitative reasoning in MAS. *AAMAS'18*, p. 1441–1449.

[Ehrenfeucht and Mycielski, 1979] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *IJGT*, 8(2):109–113.

[Emerson and Halpern, 1986] A. Emerson and J. Halpern. "Sometimes" and "not never" revisited: On branching versus linear time temporal logic. *J.ACM*, 33(1):151–178.

[Emerson and Jutla, 1991] A. Emerson and C. Jutla. Tree automata, mu-calculus and determinacy. *FOCS'91*, p. 368–377. IEEE.

[Emerson and Lei, 1987] E. A. Emerson and C.-L. Lei. Modalities for model checking: Branching time logic strikes back. *SCP*, 8:275–306.

[Fijalkow *et al.*, 2018] N. Fijalkow, B. Maubert, A. Murano, and S. Rubin. Quantifying bounds in strategy logic. *CSL'18*, LIPIcs 119, p. 23:1–23:23. LZI.

[Frigeri *et al.*, 2014] A. Frigeri, L. Pasquale, and P. Spoletini. Fuzzy time in linear temporal logic. *ACM ToCL*, 15(4):30.

[Gutierrez *et al.*, 2017] J. Gutierrez, A. Murano, G. Perelli, S. Rubin, and M. Wooldridge. Nash equilibria in concurrent games with lexicographic preferences. *IJCAI'17*, p. 1067–1073.

[Hartline and Roughgarden, 2008] J. D. Hartline and T. Roughgarden. Optimal mechanism design and money burning. *STOC'08*, p. 75–84.

[Jamroga *et al.*, 2016] W. Jamroga, B. Konikowska, and W. Penczek. Multi-valued verification of strategic ability. *AAMAS'16*, p. 1180–1189.

[Kupferman and Vardi, 2005] O. Kupferman and M. Y. Vardi. Safraless decision procedures. *FOCS'05*, p. 531–540.

[Kupferman *et al.*, 2000] O. Kupferman, M. Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model-checking. *J.ACM*, 47(2):312–360.

[Kupferman *et al.*, 2016] O. Kupferman, G. Perelli, and M. Y. Vardi. Synthesis with rational environments. *Ann. Math. Artif. Intell.*, 78(1):3–20.

[Lamine and Kabanza, 2000] K. Lamine and F. Kabanza. Using fuzzy temporal logic for monitoring behavior-based mobile robots. *Robotics and Applications*, p. 116–121.

[Laroussinie and Markey, 2014] F. Laroussinie and N. Markey. Quantified CTL: Expressiveness and complexity. *LMCS*, 10(4).

[Laroussinie and Markey, 2015] F. Laroussinie and N. Markey. Augmenting ATL with strategy contexts. *I.&C.*, 245:98–123.

[Laroussinie *et al.*, 2006] F. Laroussinie, N. Markey, and G. Oreiby. Model-checking timed ATL for durational concurrent game structures. *FORMATS'06*, LNCS 4202, p. 245–259. Springer.

[Mogavero *et al.*, 2014] F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi. Reasoning about strategies: On the model-checking problem. *ACM ToCL*, 15(4):34:1–34:47.

[Nisan *et al.*, 2007] N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani. *Algorithmic Game Theory*. CUP.

[Pnueli and Rosner, 1989] A. Pnueli and R. Rosner. On the synthesis of a reactive module. *POPL'89*, p. 179–190.

[Pnueli, 1981] A. Pnueli. The temporal semantics of concurrent programs. *TCS*, 13:45–60.

[Souza and Rego, 2016] F. Souza and L. C. Rego. Mixed equilibrium in 2×2 normal form games: when burning money is rational. *Pesqui. Oper.*, 36(1):81–99.

[Vester, 2015] S. Vester. On the complexity of model-checking branching and alternating-time temporal logics in one-counter systems. *ATVA'15*, LNCS 9364, p. 361–377. Springer.