Reasoning about Quality and Fuzziness of Strategic Behaviours

Patricia Bouyer
LSV, CNRS & ENS Paris-Saclay, Univ. Paris-Saclay, France

Orna Kupferman
Hebrew University, Israel

Nicolas Markey
Irisa, CNRS & Inria & Univ. Rennes, France

Bastien Maubert
Universitá degli Studi di Napoli "Federico II", Italy

Aniello Murano
Universitá degli Studi di Napoli "Federico II", Italy

Giuseppe Perelli
University of Leicester, UK

Abstract

Temporal logics are extensively used for the specification of on-going behaviours of reactive systems. Two significant developments in this area are the extension of traditional temporal logics with modalities that enable the specification of on-going *strategic* behaviours in multi-agent systems, and the transition of temporal logics to a *quantitative* setting, where different satisfaction values enable the specifier to formalise concepts such as certainty or quality. 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 $\mathsf{CTL}^\star$.

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).

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 [62, 32, 38]. Traditional temporal logics are interpreted over Kripke structures, modelling closed systems, and can quantify the computations of the systems in a universal and existential manner. The need to reason about multi-agents systems has led to the development of specification formalisms that enable the specification of on-going strategic behaviours in multi-agent systems [7, 29, 58]. Essentially, these formalisms, most notably ATL, $\mathsf{ATL}^\star$, and Strategy Logic (SL), include quantification of strategies of the different agents and of the computations they may force the system into, making it possible to specify concepts that have been traditionally studied in game theory.

The duration of games in game theory is typically finite and the outcome of the game depends on its final position [59, 60]. In contrast, agents in multi-agent systems maintain an *on-going interaction* with each other [45], and reasoning about their behaviour refers not to their final state (in fact, we consider non-terminating systems, with no final state) but rather to the *language* of computations that they generate. While SL, which subsumes $\mathsf{ATL}^\star$ enables the specification of rich on-going strategic behaviours, 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 [3]. 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. As an example, consider the $\mathsf{LTL}[\mathcal{F}]$ formula $\mathsf{G}(\mathit{req} \to (\mathit{grant} \oplus_{\frac{2}{3}} \mathsf{X}\mathit{grant}))$. The formula states that we want requests to be granted immediately and the grant to hold for two transactions. When this always holds, the satisfaction value is $\frac{2}{3} + \frac{1}{3} = 1$. We are quite okay with grants that are given immediately and last for only one transaction, in which case the satisfaction value is $\frac{2}{3}$, and less content when grants arrive with a delay, in which case the satisfaction value is $\frac{1}{3}$.

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 element 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 certainty. 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. Using $\mathsf{SL}[\mathcal{F}]$, we can specify, for example (see specific formulas in Section 2.4), the quality of strategies for the drones whose objectives are to fly above and below all uncontrollable drones and perform certain actions when uncontrollable drones exhibit some disallowed behaviour. Indeed, the multi-valued atomic propositions are used to specify the different heights, temporal operators are used for specifying on-going behaviours, the functions in $\mathcal{F}$ may be used to refer to these behaviours in a quantitative manner, for example to compare heights and to specify the satisfaction level that the designer gives to different possible scenarios. Note that the $\mathsf{SL}[\mathcal{F}]$ formula does not specify the ability of the drones to behave in some desired manner. Rather, it associates a satisfaction value in $[0, 1]$ with each behaviour. This suggests that $\mathsf{SL}[\mathcal{F}]$ can be used not only for a quantitative specification of strategic behaviours but also for quantizing notions from game theory that are traditionally Boolean. For example (see specific formulas in 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 that witnesses the instability. As a result we can express concepts such as ϵ-Nash Equilibria [60].

The logic $\mathsf{SL}[\mathcal{F}]$ enables the quantification of strategies for the agents. We show that the quantification of strategies 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 $\mathsf{BQCTL}^\star[\mathcal{F}]$ – an extension of $\mathsf{CTL}^\star[\mathcal{F}]$ [3] with quantified Boolean atomic propositions. 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 [40]. This general technique is applied also to $\mathsf{CTL}^\star[\mathcal{F}]$, with the fresh atomic propositions being weighted [3]. 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 [50] 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. More precisely we show that it is $(k+1)$-EXPTIME-complete for formulas involving at most k quantifications on atomic propositions, and we show a similar complexity result for $\mathsf{SL}[\mathcal{F}]$, in terms of nesting of strategy quantifiers.

Related works. There have been long lines of works about games with quantitative objectives (in a broad sense), *e.g.* stochastic games [64, 42], timed games [9], or weighted games with various kinds of objectives (parity [39], mean-payoff [37] or energy [26, 15]). This does not limit to zero-sum games, but also includes the study of various solution concepts (see for instance [65, 24, 22, 20, 5]). Similarly, extensions of the classical temporal logics LTL and CTL with quantitative semantics have been studied in different contexts, with discounting [34, 2], averaging [14, 18], or richer constructs [13, 3]. 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 [3, 4, 5, 6], but no logic combines the quantitative aspect of $\mathsf{LTL}[\mathcal{F}]$ with the strategic reasoning offered by SL. Thus, to the best of our knowledge, our model-checking algorithm for $\mathsf{SL}[\mathcal{F}]$ is the first decidability result for a quantitative extension of a strategic specification formalism (without restricting to bounded-memory strategies).

Baier and others have focused on a variant of SL in a stochastic setting [10]; model checking was proven decidable for memoryless strategies, and undecidable in the general case. A quantitative version of SL with Boolean goals over one-counter games has been considered in [16]; only a periodicity property was proven, and no model-checking algorithm is known in that setting as well. Finally, Graded SL [8] extends SL by quantifying on the number of strategies witnessing a given strategy quantifier, and is decidable.

The other quantitative extensions we know of concern $\mathsf{ATL}/\mathsf{ATL}^*$, and most of the results are actually adaptations of similar (decidability) results for the corresponding extensions of CTL and CTL^*; this includes probabilistic ATL [30], timed ATL [47, 23], multi-valued ATL [48], and weighted versions of ATL [55, 25, 66]. Finally, some works have considered non-quantitative ATL with quantitative constraints on the set of allowed strategies [1, 36], proving decidability of the model-checking problem.

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 $\mathrm{last}(w) := w_n$ is its last letter. Given a finite (resp. infinite) word w and $0 \leq i < |w|$ (resp. $i \in \mathbb{N}$), we let w_i be the letter at position i in w, $w_{\leq i} = w_0 \dots w_i$ is the (nonempty) 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. 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 [29, 58] and $\mathsf{LTL}[\mathcal{F}]$ [3] 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 [41], which does not add expressiveness with respect to the classic semantics [58] but presents several benefits (see [41] for more details), one of them being that it makes the connection with Quantified CTL tighter.

▶ **Definition 1** (Syntax). *Let $\mathcal{F} \subseteq \{f \colon [0,1]^m \to [0,1] \mid m \in \mathbb{N}\}$ be a set of functions over $[0,1]$ of possibly different arities. 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.

We may use $\top$, $\vee$, and $\neg$ to denote functions 1, max and $1-x$, respectively. We can then define the following classic abbreviations: $\bot := \neg\top$, $\varphi \wedge \varphi' := \neg(\neg\varphi \vee \neg\varphi')$, $\varphi \to \varphi' := \neg\varphi \vee \varphi'$, $\mathsf{F}\psi := \top\mathsf{U}\psi$, $\mathsf{G}\psi := \neg\mathsf{F}\neg\psi$ and $[\![x]\!]\varphi := \neg\langle\!\langle x \rangle\!\rangle\neg\varphi$.

Intuitively, the value of formula $\varphi \vee \varphi'$ is the maximal value of the two formulas φ and φ', $\varphi \wedge \varphi'$ takes the minimal value of the two formulas, and the value of $\neg\varphi$ is one minus that of φ. The implication $\varphi \to \varphi'$ thus takes the maximal value between that of φ' and one minus that of φ.

In a Boolean setting, we assume that the values of the atomic propositions are in $\{0,1\}$ (0 represents false whereas 1 represents true), and so are the output values of functions in $\mathcal{F}$. One can then check that $\varphi \vee \varphi'$, $\varphi \wedge \varphi'$, $\neg\varphi$ and $\varphi \to \varphi'$ take their usual Boolean meaning.

We will come back later to temporal modalities, strategy quantifications and bindings.

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 $[0,1]$, and that we now present.

▶ **Definition 2.** *A* weighted concurrent game structure *(WCGS) is a tuple $\mathcal{G} = (\mathsf{AP}, \mathsf{Agt}, \mathsf{Act}, \mathsf{V}, \mathsf{v}_\iota, \Delta, \mathsf{w})$ 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, $\mathsf{v}_\iota \in \mathsf{V}$ is an initial state, $\Delta \colon \mathsf{V} \times \mathsf{Act}^{\mathsf{Agt}} \to \mathsf{V}$ is the transition function, and $\mathsf{w} \colon \mathsf{V} \to [0,1]^{\mathsf{AP}}$ 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 $\{\mathsf{v}' \in \mathsf{V} \mid \exists \vec{c} \in \mathsf{Act}^{\mathsf{Agt}}.\ \mathsf{v}' = \Delta(\mathsf{v}, \vec{c})\}$. For the sake of simplicity, we assume in the sequel that $\mathsf{succ}(\mathsf{v}) \neq \varnothing$ for all $\mathsf{v} \in \mathsf{V}$.

A *play* in $\mathcal{G}$ is an infinite sequence $\pi = (\mathsf{v}_i)_{i\in\mathbb{N}}$ of states in V 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. In this and all similar notations, we might omit to mention $\mathcal{G}$ when it is clear from the context. A (strict) prefix of a play π is a finite sequence $\rho = (\mathsf{v}_i)_{0\leq i\leq L}$, for some $L \in \mathbb{N}$, which we denote $\pi_{\leq L}$. We write $\mathsf{Pref}(\pi)$ for the set of strict prefixes of play π. Such finite prefixes are called *histories*, and we let $\mathsf{Hist}_{\mathcal{G}}(\mathsf{v}) = \mathsf{Pref}(\mathsf{Play}_{\mathcal{G}}(\mathsf{v}))$ and $\mathsf{Hist}_{\mathcal{G}} = \bigcup_{\mathsf{v}\in\mathsf{V}} \mathsf{Hist}_{\mathcal{G}}(\mathsf{v})$. We extend the notion of strict prefixes and the notation Pref to histories in the natural way, requiring in particular that $\rho \notin \mathsf{Pref}(\rho)$. A (finite) extension of a history ρ is any history ρ' such that $\rho \in \mathsf{Pref}(\rho')$.

A *strategy* is a mapping $\sigma\colon \mathsf{Hist}_{\mathcal{G}} \to \mathsf{Act}$, and we write $\mathrm{Str}_{\mathcal{G}}$ for 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. The assignment $\chi[a \mapsto \sigma]$ maps a to σ and is equal to χ otherwise. Let χ be an assignment and ρ a history. We define the set of *outcomes* of χ from ρ as the set $\mathsf{Out}(\chi, \rho)$ of plays $\pi = \rho \cdot \mathsf{v}_1\mathsf{v}_2 \ldots$ 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** (Semantics). *Consider a WCGS $\mathcal{G} = (\mathsf{AP}, \mathsf{Agt}, \mathsf{Act}, V, \mathsf{v}_\iota, \Delta, \mathsf{w})$, a set of variables Var, and a partial assignment χ of strategies for Agt and Var. Given an $\mathsf{SL}[\mathcal{F}]$ state formula φ and a history ρ, we use $[\![\varphi]\!]^{\mathcal{G}}_{\chi}(\rho)$ to denote the satisfaction value of φ in the last state of ρ under the assignment χ. Given an $\mathsf{SL}[\mathcal{F}]$ path formula ψ, a play π, and a point in time $i \in \mathbb{N}$, we use $[\![\psi]\!]^{\mathcal{G}}_{\chi}(\pi, i)$ to denote the satisfaction value of ψ in the suffix of π that starts in position i. The satisfaction value is defined inductively as follows:*

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

Strategy quantification $\langle\!\langle x\rangle\!\rangle\varphi$ computes the optimal value a choice of strategy for variable x can give to formula φ. Dually, $[\![x]\!]\varphi$ computes the minimal value a choice of strategy for variable x can give to formula φ. Binding $(a, x)\varphi$ just assigns strategy given by x to agent a. Temporal modality $\mathsf{X}\psi$ takes the value of ψ at the next step, while $\psi_1\mathsf{U}\psi_2$ maximises, 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.

In a Boolean setting, we recover the standard semantics of SL. Also the fragment of $\mathsf{SL}[\mathcal{F}]$ with only temporal operators and functions $\vee$ and $\neg$ corresponds to Fuzzy Linear-time Temporal Logic [52, 44]. 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 [44]). However the interpretation of the temporal operators is fixed in our logic.

▶ Remark 4. As we shall see, once we fix a finite set of possible satisfaction values for the atomic propositions in a formula φ, as is the case when a model is chosen, the set of possible satisfaction values for subformulas of φ becomes finite. Therefore, the infima and suprema in the above definition are in fact minima and maxima.

For a state formula φ and a weighted game structure $\mathcal{G}$, we write $[\![\varphi]\!]^{\mathcal{G}}$ for $[\![\varphi]\!]^{\mathcal{G}}_{\emptyset}(\mathsf{v}_\iota)$.

2.3 Model checking

The problem we are interested in is the following generalisation of the model checking problem, which is solved in [3] for $\mathsf{LTL}[\mathcal{F}]$ and $\mathsf{CTL}^\star[\mathcal{F}]$.

▶ **Definition 5** (Model-checking problem). *Given an* $\mathsf{SL}[\mathcal{F}]$ *state formula* φ*, a WCGS* $\mathcal{G}$ *and a predicate* $P \subseteq [0,1]$*, decide whether* $[\![\varphi]\!]^{\mathcal{G}} \in P$*.*

Note that P should be finitely represented, typically as a threshold or an interval.

The precise complexity of the model-checking problem will be stated in terms of *nesting depth* of formulas, which counts the maximal number of strategy quantifiers in a formula φ, and is written $\mathrm{nd}(\varphi)$. We establish the following result in Section 5:

▶ **Theorem 6.** *The model-checking problem for* $\mathsf{SL}[\mathcal{F}]$ *is decidable. It is* $(k+1)$-EXPTIME-*complete for formulas of nesting depth at most* k*.*

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 the only allowed functions in $\mathcal{F}$ are $\vee, \wedge$, and $\neg$, then the satisfaction value of the formula is in $\{0,1\}$ and coincides with the value of the corresponding SL formula. 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 artefact is never rescued.

The strategies of the carrier and the guard being quantified before that of the villain implies that they are unaware of the villain's future moves. Now assume the guard is a double agent to whom the villain communicates his plan. Then his strategy can depend on the villain's strategy, which is captured by the following formula:

$$\varphi_{\mathrm{spy}} = \langle\!\langle x \rangle\!\rangle [\![z]\!] \langle\!\langle y \rangle\!\rangle (c,x)(g,y)(v,z)\ \mathsf{A}(\mathsf{dist}(p_{\vec{\gamma}}, q_{\vec{\gamma}})\mathsf{U}\ \mathsf{safe})$$

Note that the formula $\varphi_{\mathrm{rescue}}$ can be written in $\mathsf{ATL}[\mathcal{F}]$, whereas φ_{spy} requires $\mathsf{SL}[\mathcal{F}]$. In fact φ_{spy} actually belongs to the fragment $\mathsf{SL}_{\mathsf{1G}}[\mathcal{F}]$, which we study in Section 6.

Synthesis with quantitative objectives

The problem of synthesis for LTL specifications dates back to [63]. The setting is simple: two agents, a controller and an environment, operate on two disjoint sets of variables in the system. The controller wants a given LTL specification ψ to be satisfied in the infinite execution, while the environment wants to prevent it. The problem consists into 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 controller aims at maximising the value of an $\mathsf{LTL}[\mathcal{F}]$ formula φ, while the environment acts as minimiser. Both problems can be easily represented in SL and $\mathsf{SL}[\mathcal{F}]$ respectively, with the formula

$$\varphi_{\mathtt{synt}} = \langle\!\langle x \rangle\!\rangle [\![y]\!]\ (c,x)(e,y)\ \mathsf{A}\psi$$

where c and e are the controller and environment agent, respectively, and ψ the temporal specification expressed in either LTL or $\mathsf{LTL}[\mathcal{F}]$.

Assume now that controller and environment are both composed of more than one agent, namely $c_1, \ldots, c_n$ and $e_1, \ldots, e_n$, and each controller component has the power to adjust its strategic choice based on the strategies selected by the environmental agents of lower rank. That is, the strategy selected by agent c_k depends, on the strategies selected by agents e_j, for every $j < k$. We can write a $\mathsf{SL}[\mathcal{F}]$ formula to represent this generalised synthesis problem as follows:

$$\varphi_{\mathtt{synt}} = \langle\!\langle x_1 \rangle\!\rangle [\![y_1]\!] \cdots \langle\!\langle x_n \rangle\!\rangle [\![y_n]\!]\ (c_1,x_1)(e_1,y_1)\ldots(c_n,x_n)(e_n,y_n)\ \mathsf{A}\psi.$$

Notice that every controller agent is bound to an existentially quantified variable, that makes it to maximise the satisfaction value of the formula in its scope. On the other hand, the environmental agents are bound to a universally quantified variable, that makes them to minimise the satisfaction value.

Notice that in general each alternation between existential and universal quantification yields an additional exponential in the complexity of the model-checking problem, as we show in the overall quantification alternates from existential to universal $2n - 1$ times, which would induce a . In section 6, we show that, for the special case of these formulas, such alternation does not affect the computational complexity of the model-checking problem.

NE in weighted games

An important feature of SL in terms of expressiveness 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.

Consider a strategy profile $(x_i)_{a_i \in \mathsf{Agt}}$. Assuming all agents follow their strategies in that profile, a NE can be characterised by the fact that all agents play one of their best responses against their opponents' strategies. We would then write

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

where $\alpha \preceq \beta$ equals 1 if $\alpha \leq \beta$ and zero otherwise.[1] Strategy profile $(x_i)_{a_i \in \mathsf{Agt}}$ is a NE if, and only if, $\varphi_{\mathsf{NE}}((x_i)_{a_i \in \mathsf{Agt}})$ evaluates to 1.

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:

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

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 [28] 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:

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

where $(\alpha_1, \alpha_2) \preceq_i (\beta_1, \beta_2)$ is 1 when $(\alpha_i \leq \beta_i) \vee (\alpha_i = \beta_i \wedge \alpha_{3-i} \leq \beta_{3-i})$, and 0 otherwise.

Secure equilibria have also been studied in $\mathbb{Q}$-weighted games [24]: 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}.

Other classical properties of games can be expressed, such as doomsday equilibria (which generalise winning secure equilibria in n-player games) [27], robust NE (considering profitable deviations of coalitions of agents) [19], or strategy dominance and admissibility [12, 21], to cite a few.

Rational synthesis

Weak rational synthesis [43, 49, 5] aims at synthesising a strategy profile for a controller C_0 and the n components $(C_i)_{1 \leq i \leq n}$ constituting the environment, in such a way that (1) the whole system satisfies some objective φ_0, and (2) under the strategy of the controller, the strategies of the n components form an NE (or any given solution concept) for their own individual objectives $(\varphi_i)_{1 \leq i \leq n}$.

[1] The conference version of this paper [17] contains a typo: in φ_{NE}, the quantification $[\![y_i]\!]$ was existential instead of being universal, and similarly for the formula φ_{SE} that characterises secure equilibria.

That a given strategy profile $(x_i)_{C_i \in \mathsf{Agt}}$ satisfies the two conditions above can be expressed as follows:

$$\varphi_{\mathsf{wRS}}((x_i)_{0 \le i \le n}) = (C_0, x_0)(C_1, x_1) \dots (C_n, x_n)[\mathsf{A}\varphi_0 \wedge \varphi_{\mathsf{NE}}((x_i)_{1 \le i \le n})]$$

The formula returns the minimum between the satisfaction value of φ_0 and that of $\varphi_{\mathsf{NE}}((x_i)_{1 \le i \le n})$. Thus, the satisfaction value of φ_{wRS} is zero if the strategy profile $(x_i)_{1 \le i \le n}$ is not a NE under strategy x_0 assigned to C_0, and it returns the satisfaction value of φ_0 under the whole strategy profile otherwise. Then the value of

$$\langle\!\langle x_0 \rangle\!\rangle \langle\!\langle x_1 \rangle\!\rangle \dots \langle\!\langle x_n \rangle\!\rangle \; \varphi_{\mathsf{wRS}}((x_i)_{0 \le i \le n})$$

is the best value of φ_0 that the system can collectively achieve under the condition that the components in the environment are in an NE. Obviously, we can go beyond NE and use any other solution concept that can be expressed in $\mathsf{SL}[\mathcal{F}]$.

The counterpart of weak rational synthesis is strong rational synthesis, that aims at synthesising a strategy only for controller C_0 in such a way that the objective φ_0 is maximised over the worst NE that can be played by the environment component over the strategy of C_0 itself.

This can be expressed as follows:

$$\varphi_{\mathsf{sRS}}(x_0) = [\![x_1]\!] \dots [\![x_n]\!] \; (C_0, x_0)(C_1, x_1) \dots (C_n, x_n)[\neg\varphi_{\mathsf{NE}}((x_i)_{1 \le i \le n}) \vee \varphi_0]$$

The formula in the scope of the quantifications and bindings returns the maximum value between $\neg\varphi_{\mathsf{NE}}((x_i)_{1 \le i \le n})$ and φ_0. Given that the former is 1 if there is no NE and 0 otherwise, the disjunction takes value 1 over the path with no NEs and the value of φ_0, otherwise. Given that the environment components have universally quantified strategies, the formula φ_{sRS} amounts at minimising such disjunction. Thus, the components will select (if any) the NE that minimises the satisfaction value of φ_0. Then the value of

$$\langle\!\langle x_0 \rangle\!\rangle \varphi_{\mathsf{sRS}}(x_0)$$

is the best value of φ_0 that the controller can achieve under the condition that the components in the environment are playing the Nash Equilibrium that worsens it.

Core equilibria

In cooperative game theory, *core equilibrium* is probably the best known solution concept and sometimes related to the one of Nash Equilibrium for noncooperative games. Differently from NEs (but similarly to Strong NEs) it accounts multilateral deviations (also called coalition deviations) that, in order to be beneficial, must improve the payoff of the deviating agents *no matter* what is the reaction of the opposite coalition. More formally, for a given strategy profile $(x_i)_{a_i \in \mathsf{Agt}}$, a coalition $C \subseteq \mathsf{Agt}$ has a *beneficial deviation* $(y_i)_{a_i \in C}$ if, for all strategy profiles $(z_i)_{a_i \in \mathsf{Agt} \setminus C}$ it holds that $(x_i)_{a_i \in \mathsf{Agt}} \varphi_i \prec (y_i)_{a_i \in C} (z_i)_{a_i \in \mathsf{Agt} \setminus C} \varphi_i$, for every $a_i \in C$. We say that a strategy profile $(x_i)_{a_i \in \mathsf{Agt}}$ is a *core equilibrium* if, for every coalition, there is no beneficial deviation. This can be written in $\mathsf{SL}[\mathcal{F}]$ as follows:

$$\varphi_{\mathsf{core}}(x_i)_{a_i \in \mathsf{Agt}} = \bigwedge_{C \subseteq \mathsf{Agt}} [\![y_i]\!]_{a_i \in C} \langle\!\langle y_i \rangle\!\rangle_{a_j \in \mathsf{Agt} \setminus C} \big(\bigwedge_{a_j \in C} (a_i, y_i)_{a_i \in \mathsf{Agt}} \varphi_j \preceq (a_i, x_i)_{a_i \in \mathsf{Agt}} \varphi_j \big)$$

The strategy profile $(x_i)_{a_i \in \mathsf{Agt}}$ is a core equilibrium if, and only if, the formula $\varphi_{\mathsf{core}}(x_i)_{a_i \in \mathsf{Agt}}$ evaluates to 1. The existence of a core equilibrium could be than expressed with the formula $\langle\!\langle x_1 \rangle\!\rangle \dots \langle\!\langle x_n \rangle\!\rangle \varphi_{\mathsf{core}}(x_i)_{a_i \in \mathsf{Agt}}$, which takes value 1 if and only if there exists a core equilibrium.

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 $\mathsf{QCTL}^\star$ [53]. On the one hand, it extends $\mathsf{CTL}^\star[\mathcal{F}]$ with second order quantification over Boolean atomic propositions, on the other hand it extends $\mathsf{QCTL}^\star$ to the quantitative setting of $\mathsf{CTL}^\star[\mathcal{F}]$. 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.

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 over $[0,1]$.

▶ **Definition 7.** *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 the state formulas defined by the grammar. An atomic proposition which is not under the scope of a quantification is called *free*. If no atomic proposition is free in a formula φ, then we say that φ is *closed*. 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

$\mathsf{BQCTL}^\star[\mathcal{F}]$ formulas are evaluated on unfoldings of weighted Kripke structures.Note that the terminology Boolean only concerns the quantification of atomic propositions (which is restricted to Boolean atomic propositions), and that formulas are interpreted over weighted Kripke structures.

▶ **Definition 8.** *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*[2]*, 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. By analogy with concurrent game structures we call finite prefixes of paths *histories*, and write $\mathsf{Hist}_\mathcal{K}$ for 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 $\mathcal{K}$.

Trees 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 or irrelevant), 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* $\lambda = u_0u_1\ldots$ is an infinite sequence of nodes

[2] *i.e.*, for all $\mathsf{s} \in \mathsf{S}$, there exists s' such that $(\mathsf{s}, \mathsf{s}') \in R$.

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. Given a tree $t = (\tau, \mathsf{w})$ and a node $u \in \tau$, we define the *subtree of t rooted in u* as the tree $t_u = (\tau_u, \mathsf{w}')$ where $\tau_u = \{v \in \mathsf{S}^+ : u \preccurlyeq v\}$ ($\preccurlyeq$ denotes the non-strict prefix relation) and w' is w restricted to τ_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, we define the *tree unfolding of a weighted Kripke structure $\mathcal{K}$* over atomic propositions AP and states S as 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 9** (Semantics). *Consider finite sets D of directions, AP of atomic propositions, and $V \subseteq [0, 1]$ of possible values. We fix an (AP, V)-labelled D-tree τ. Given a $\mathsf{BQCTL}^\star[\mathcal{F}]$ state formula φ and a node u of τ, we use $[\![\varphi]\!]^t(u)$ to denote the satisfaction value of φ in node u. Given a $\mathsf{BQCTL}^\star[\mathcal{F}]$ path formula ψ and a branch λ of τ, we use $[\![\psi]\!]^t(\lambda)$ to denote the satisfaction value of ψ along λ. The satisfaction value is defined inductively as follows:*

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

▶ Remark 10. As with $\mathsf{SL}[\mathcal{F}]$, we will see that the suprema in the above definition can be replaced with maxima (see Lemma 13 below).

First, we point out that if $\mathcal{F} = \{\top, \vee, \neg\}$, then $\mathsf{BQCTL}^\star[\mathcal{F}]$ evaluated on boolean Kripke structures corresponds to classic $\mathsf{QCTL}^\star$ [53]. 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 weighted Kripke structure $\mathcal{K}$ 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 Strategy Logic based on strategies with perfect recall, where moves can depend on the history, as apposed to the memoryless semantics, where strategies can only depend on the current state (see [53] for more detail).

As for $\mathsf{SL}[\mathcal{F}]$, we are interested in the following model-checking problem:

▶ **Definition 11.** *Given a* $\mathsf{BQCTL}^\star[\mathcal{F}]$ *state formula* φ*, a weighted Kripke structure* $\mathcal{K}$*, 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 12.** *The quantitative model-checking problem for* $\mathsf{BQCTL}^\star[\mathcal{F}]$ *is decidable. It is* $(k+1)$-EXPTIME*-complete for formulas of nesting depth at most* k.

This result, together with a reduction from $\mathsf{SL}[\mathcal{F}]$ to $\mathsf{BQCTL}^\star[\mathcal{F}]$ that we present in Section 5, entails the decidability of model checking $\mathsf{SL}[\mathcal{F}]$ announced in Theorem 6.

4 Model checking $\mathsf{BQCTL}^\star[\mathcal{F}]$

We start by proving that, as has been the case for $\mathsf{LTL}[\mathcal{F}]$, since the set of possible satisfaction values of an atomic proposition is finite, so is the set of satisfaction values of each $\mathsf{BQCTL}^\star[\mathcal{F}]$ formula. This property allows to use max instead of sup in Definition 9.

▶ **Lemma 13.** *Let* $V \subset [0,1]$ *be a finite set of values with* $\{0,1\} \subseteq V$*, let* φ *be a* $\mathsf{BQCTL}^\star[\mathcal{F}]$ *state formula and* ψ *a* $\mathsf{BQCTL}^\star[\mathcal{F}]$ *path formula, with respect to* AP*. 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. Similarly, define*

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

Then, $|V_\varphi| \leq |V|^{|\varphi|}$ *and* $|V_\psi| \leq |V|^{|\psi|}$*. Moreover, one can compute sets* $\widetilde{V}_\varphi$ *and* $\widetilde{V}_\psi$ *such that* $V_\varphi \subseteq \widetilde{V}_\varphi$ *and* $V_\psi \subseteq \widetilde{V}_\psi$ *of size at most* $|V|^{|\varphi|}$ *and* $|V|^{|\psi|}$*, respectively.*

Proof. We prove the result by mutual induction on φ and ψ. Clearly, $V_p = V$.

For $\varphi = \exists p.\, \varphi'$, observe that if $V_t \subseteq V$ and $u \in t$, then for all trees t' such that $t' \equiv_p t$ it is also the case that $V_{t'} \subseteq V$ (by assumption V contains 0 and 1). It follows that $[\![\exists p.\, \varphi']\!]^t(u)$ is defined as the supremum of a subset of $V_{\varphi'}$, which by induction hypothesis is of size at most $|V|^{|\varphi'|}$, and thus the supremum is indeed a maximum. It follows that $[\![\exists p.\, \varphi']\!]^t(u) \in V_{\varphi'}$. Hence, $V_{\exists p.\, \varphi'} \subseteq V_{\varphi'}$, and thus $|V_{\exists p.\, \varphi'}| \leq |V_{\varphi'}| \leq |V|^{|\varphi'|} \leq |V|^{|\exists p.\, \varphi'|}$.

For $\varphi = \mathsf{E}\psi$, again $[\![\mathsf{E}\psi]\!]^t(u)$ is a supremum over a subset of V_ψ, which by induction hypothesis is of size at most $V^{|\psi|}$. The supremum is thus reached, hence $V_{\mathsf{E}\psi} \subseteq V_\psi$ and $|V_{\mathsf{E}\psi}| \leq |V_\psi| \leq |V|^{|\psi|} \leq |V|^{|\mathsf{E}\psi|}$.

For $\varphi = f(\varphi_1, \ldots, \varphi_n)$, we have $V_\varphi = \{f(v_1, \ldots, v_n) \mid v_i \in V_{\varphi_i}\}$, hence $|V_\varphi|$ is at most $\prod_{i=1}^n |V_{\varphi_i}|$. By induction hypothesis, we get $|V_\varphi| \leq \prod_{i=1}^n |V|^{|\varphi_i|} \leq |V|^{|\varphi_1|+\ldots+|\varphi_n|} \leq |V|^{|\varphi|}$.

For $\psi = \varphi$, the result follows by hypothesis of mutual induction.

For $\psi = \mathsf{X}\psi'$, we have that $V_\psi = V_{\psi'}$, and the result follows.

For $\psi = \psi_1 \mathsf{U} \psi_2$, the value of ψ is defined via suprema and infima over possible values for ψ_1 and ψ_2, which are finitely many by the induction hypothesis. The suprema and infima are thus maxima and minima, and $V_\psi \subseteq V_{\psi_1} \cup V_{\psi_2}$. Hence, $|V_\psi| \leq |V_{\psi_1}| + |V_{\psi_2}| \leq |V|^{|\psi_1|} + |V|^{|\psi_2|} \leq |V|^{|\psi_1|+|\psi_2|} \leq |V|^{|\psi|}$ (since $|V| \geq 2$).

In all cases, the claim for over-approximations follows by the same reasoning as above. ◀

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 for a detailed exposition of alternating parity tree automata.

We extend the automata-based model-checking procedure for $\mathsf{CTL}^\star$ from [50]. Note that since the quantified atomic propositions may appear in different subformulas, we cannot extend the algorithm for $\mathsf{CTL}^\star[\mathcal{F}]$ from [3], as the latter applies the technique of [40], where the evaluation of each subformula is independent.

▶ **Proposition 14.** *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 $\mathcal{A}^{V,P}_\varphi$ such that for every (AP, V)-labelled D-tree t, $\mathcal{A}^{V,P}_\varphi$ accepts t if and only if $[\![\varphi]\!]^t \in P$.*

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

Proof. The proof proceeds by an induction on the structure of the formula φ and strengthens the induction statement as follows: one can construct an APT $\mathcal{A}^{V,P}_\varphi$ such that for every (AP, V)-labelled D-tree t, for every node $u \in t$, we have that $\mathcal{A}^{V,P}_\varphi$ accepts t from node u if and only if $[\![\varphi]\!]^t(u) \in P$.

If $\varphi = p$, the automaton has one state and accepts a tree $t = (\tau, \mathsf{w})$ in node $u \in \tau$ if $\mathsf{w}(u)(p) \in P$, and rejects otherwise. In addition, $V_p = V$.

If $\varphi = \exists p.\, \varphi'$, we want to check whether the maximal satisfaction value of φ' for all possible Boolean valuations of p is in P. To do so we first compute a finite set $\widetilde{V}_{\varphi'}$ of exponential size such that $V_{\varphi'} \subseteq \widetilde{V}_{\varphi'}$, which we can do as established by Lemma 13. For each possible value $v \in \widetilde{V}_{\varphi'} \cap P$, we check whether this value is reached for some p-valuation, and if the value of φ' is less than or equal to v for all p-valuations. For each $v \in \widetilde{V}_{\varphi'} \cap P$, inductively build the APTs $\mathcal{A}^{V,\{v\}}_{\varphi'}$ and $\mathcal{A}^{V,[0,v]\cap P}_{\varphi'}$. Turn the first one into a NPT $\mathcal{N}_{=v}$ and the second one into a UPT $\mathcal{U}_{\le v}$. Project $\mathcal{N}_{=v}$ existentially on p, and call the result $\mathcal{N}'_{=v}$. Project $\mathcal{U}_{\le v}$ universally on p, call the result $\mathcal{U}'_{\le v}$. Finally, we can define the APT $\mathcal{A}^{V,P}_{\exists p.\,\varphi'} := \bigvee_{v \in \widetilde{V}_{\varphi'} \cap P} \mathcal{N}'_{=v} \wedge \mathcal{U}'_{\le v}$. It is then easy to see that 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.

If $\varphi = \mathsf{E}\psi$: as in the classic automata construction for $\mathsf{CTL}^\star$ [51], we first let $\mathsf{atoms}(\psi)$ be the set of maximal state sub-formulas of ψ (that we call *atoms* thereafter – which have to be distinguished from atomic propositions of the formula). In a first step we see elements of $\mathsf{atoms}(\psi)$ as atomic propositions, and ψ as an $\mathsf{LTL}[\mathcal{F}]$ formula over $\mathsf{atoms}(\psi)$. According to Lemma 13 we can compute over-approximations $\widetilde{V}_{\varphi'}$ for each $\varphi' \in \mathsf{atoms}(\psi)$, and we thus let $\widetilde{V} = \bigcup_{\varphi' \in \mathsf{atoms}(\psi)} \widetilde{V}_{\varphi'}$ be a finite over-approximation of the set of possible values for atoms. It is proven in [3] that for every $P \subseteq [0,1]$, one can build a nondeterministic parity automaton $\mathcal{W}^\psi_P$ of size exponential in $|\psi|^2$ such that $\mathcal{W}^\psi_P$ accepts a word $w \in (\widetilde{V}^{\mathsf{atoms}(\psi)})^\omega$ if and only if $[\![\psi]\!](w) \in P$. Now let us compute $\widetilde{V}_{\mathsf{E}\psi}$ (again using Lemma 13), 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. To obtain a universal word automaton of single exponential size that checks whether $[\![\psi]\!](w) \in [0,v]$, first build the nondeterministic automaton $\mathcal{W}^\psi_{(v,1]}$ from [3], and dualize it in linear time. From the resulting universal automaton $\mathcal{W}^\psi_{[0,v]}$ we build a UPT $\mathcal{U}^{A\le v}$ that executes $\mathcal{W}^{\psi}_{[0,v]}$ on all branches of its input.[3] We now define the APT $\mathcal{A}^P$ on $\widetilde{V}^{\mathsf{atoms}(\psi)}$-trees as

$$\mathcal{A}^P = \bigvee_{v \in \widetilde{V}_{\mathsf{E}\psi} \cap P} \mathcal{N}^{E=v} \wedge \mathcal{U}^{A \leq v}.$$

Now to go from atoms to standard atomic propositions, we define an APT $\mathcal{A}^{V,P}_{\mathsf{E}\psi}$ that simulates $\mathcal{A}^P$ by, in each state and each node of its input, guessing a value v_i in $\widetilde{V}_{\varphi_i}$ for each formula $\varphi_i \in \mathsf{atoms}(\psi)$, simulating $\mathcal{A}^P$ on the resulting label, and launching 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.

Finally, if $\varphi = f(\varphi_1, ..., \varphi_n)$, we list all combinations $(v_1, \ldots, v_n)$ of the possible satisfaction values for the subformulas φ_i such that $f(v_1, \ldots, v_n) \in P$, and we build automaton $\mathcal{A}^{V,P}_{\varphi}$ as the disjunction over such $(v_1, \ldots, v_n)$ of the conjunction of automata $\mathcal{A}^{V,v_i}_{\varphi_i}$.

The complexity of this procedure is non-elementary. More precisely, we claim that $\mathcal{A}^{V,P}_{\varphi}$ has size at most $(\mathrm{nd}(\varphi) + 1)$-exponential and index (i.e., number of priorities for the parity condition) at most $\mathrm{nd}(\varphi)$-exponential.

The case where φ is an atomic proposition is trivial.

For $\varphi = \exists p.\, \varphi'$, we transform an exponential number of APTs into NPTs or UPTs. This entails an exponential blowup in the size and index of each automaton. The resulting automaton $\mathcal{A}^{V,P}_{\exists p.\, \varphi'}$ has at most $(\mathrm{nd}(\varphi') + 2)$-exponentially many states and index at most $(\mathrm{nd}(\varphi') + 1)$-exponential. Since $\mathrm{nd}(\varphi) = \mathrm{nd}(\varphi') + 1$, the inductive property is preserved.

If φ has the form $f(\varphi_1, ..., \varphi_n)$, then the automaton for φ is a combination of the automata for all φ_i, and for the various values those subformulas may take. By Lemma 13 there are at most $|V|^{|\varphi_1|+\cdots+|\varphi_n|} \leq |V|^{|\varphi|}$ different combinations, so assuming (from the induction hypothesis) that the automata for φ_i have at most $(\mathrm{nd}(\varphi_i)+1)$-exponentially many states and index at most $h(\varphi_i)$-exponential, the automaton for φ has at most $(\mathrm{nd}(\varphi) + 1)$-exponentially many states and index at most $\mathrm{nd}(\varphi)$-exponential (note that $\mathrm{nd}(\varphi_i) = \mathrm{nd}(\varphi)$).

Finally for $\varphi = \mathsf{E}\psi$: following [3], the size of $\mathcal{W}^{\psi}_{P}$ is exponential in $|\psi|^2$, and at most $|\psi|$ Büchi acceptance conditions. One can turn this automaton into an equivalent Büchi automaton still exponential in $|\psi|^2$, which can be seen as a parity automaton with index 2. Then $\mathcal{N}^{E=v}$ and $\mathcal{U}^{A\leq v}$ both also have sizes exponential in $|\psi|^2$, and index 2. Finally, $\mathcal{A}^P$, which combines an exponential number of the automata above, has size exponential in $|\psi|^2$ and index 2. The final automaton $\mathcal{A}^{V,P}_{\mathsf{E}\psi}$ is obtained from that automaton by plugging the automata for $\mathcal{A}^{V,\bullet}_{\varphi_i}$, whose sizes and indices are dominating the size and index of $\mathcal{A}^P$. It follows that, for $\varphi = \mathsf{E}\psi$, the size of $\mathcal{A}^{V,P}_{\varphi}$ also is $(\mathrm{nd}(\varphi) + 1)$-exponential, and its index is $\mathrm{nd}(\varphi)$-exponential. ◀

To see that Theorem 12 follows from Proposition 14, recall that by definition $[\![\varphi]\!]^{\mathcal{K}} = [\![\varphi]\!]^{t_{\mathcal{K}}}$. Thus to check whether $[\![\varphi]\!]^{\mathcal{K}} \in P$, where atoms in $\mathcal{K}$ takes values in V, it is enough to build $\mathcal{A}^{V,P}_{\varphi}$ as in Proposition 14, 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.

[3] We take $\mathcal{W}^{\psi}_{[0,v]}$ universal because it is not possible to simulate a nondeterministic word automaton on all branches of a tree, but it is possible for a universal one. Note that we could also determinise $\mathcal{W}^{\psi}_{[0,v]}$, but it would cost one more exponential.

For the lower bounds, consider the fragment $\mathsf{EQ}^k\mathsf{CTL}^\star$ of $\mathsf{QCTL}^\star$ which consists in formulas in prenex normal form, i.e. with all quantifications on atomic propositions at the beginning, with at most k alternations between existential and universal quantifiers, counting the first quantifier as one alternation (see [53, p.8] for a formal definition). 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 placing function $\neg$ between quantifiers). It is proved in [53] that model checking $\mathsf{EQ}^k\mathsf{CTL}^\star$ is $(k+1)$-EXPTIME-hard.

5 Model checking quantitative strategic logics

In this section we show how to reduce the model-checking problem for $\mathsf{SL}[\mathcal{F}]$ to that of $\mathsf{QCTL}^\star[\mathcal{F}]$. This reduction is a rather straightforward adaptation of the usual one for qualitative variants of SL (see e.g. [54, 11, 41]). We essentially observe that it can be lifted to the quantitative setting.

We let Agt be a finite set of agents, and AP be a finite set of atomic propositions.

Models transformation. We first define for every WCGS $\mathcal{G} = (\mathsf{Act}, \mathsf{V}, \mathsf{v}_\iota, \Delta, \mathsf{w})$ over Agt and AP a WKS $\mathcal{K}_\mathcal{G} = (\mathsf{S}, \mathsf{s}_\iota, R, \mathsf{w})$ over some set AP' and a bijection $\rho \mapsto u_\rho$ between the set of histories starting in the initial state v_ι of $\mathcal{G}$ and the set of nodes in $t_{\mathcal{K}_\mathcal{G}}$. We consider propositions $\mathsf{AP}_\mathsf{V} = \{p_\mathsf{v} \mid \mathsf{v} \in \mathsf{V}\}$, that we assume to be disjoint from AP. We let $\mathsf{AP}' = \mathsf{AP} \cup \mathsf{AP}_\mathsf{V}$. Define the Kripke structure $\mathcal{K}_\mathcal{G} = (\mathsf{S}, \mathsf{s}_\iota, R, \mathsf{w})$ where

- $\mathsf{S} = \{\mathsf{s}_\mathsf{v} \mid \mathsf{v} \in \mathsf{V}\}$,
- $\mathsf{s}_\iota = \mathsf{s}_{\mathsf{v}_\iota}$,
- $R = \{(\mathsf{s}_\mathsf{v}, \mathsf{s}_{\mathsf{v}'}) \in \mathsf{S}^2 \mid \exists \vec{c} \in \mathsf{Act}^{\mathsf{Agt}} \text{ s.t. } \Delta(\mathsf{v}, \vec{c}) = \mathsf{v}'\}$, and
- $\mathsf{w}(p)(\mathsf{s}_\mathsf{v}) = \begin{cases} 1 & \text{if } p \in \mathsf{AP}_\mathsf{V} \text{ and } p = p_\mathsf{v} \\ 0 & \text{if } p \in \mathsf{AP}_\mathsf{V} \text{ and } p \neq p_\mathsf{v}. \\ \mathsf{w}(p)(\mathsf{v}) & \text{otherwise} \end{cases}$

For every history $\rho = \mathsf{v}_0 \dots \mathsf{v}_k$, define the node $u_\rho = \mathsf{s}_{\mathsf{v}_0} \dots \mathsf{s}_{\mathsf{v}_k}$ in $t_{\mathcal{K}_\mathcal{G}}$ (which exists, by definition of $\mathcal{K}_\mathcal{G}$ and of tree unfoldings). Note that the mapping $\rho \mapsto u_\rho$ defines a bijection between the set of histories from v_ι and the set of nodes in $t_{\mathcal{K}_\mathcal{G}}$.

Formulas translation. Given a game $\mathcal{G} = (\mathsf{Act}, \mathsf{V}, \mathsf{v}_\iota, \Delta, \mathsf{w})$ and a formula $\varphi \in \mathsf{SL}[\mathcal{F}]$, we define a $\mathsf{QCTL}^\star[\mathcal{F}]$ formula φ' such that $[\![\varphi]\!]^\mathcal{G} = [\![\varphi']\!]^{\mathcal{K}_\mathcal{G}}$. More precisely, this translation is parameterised with a partial function $g : \mathsf{Agt} \rightharpoonup \mathsf{Var}$ which records bindings of agents to strategy variables. Suppose that $\mathsf{Act} = \{c_1, \dots, c_m\}$. We define the translation $(\cdot)^g$ by induction on state formulas φ and path formulas ψ. Here is the definition of $(\cdot)^g$ for state formulas:

$$(p)^g = p \qquad\qquad ((a,x)\varphi)^g = (\varphi)^{g[a\mapsto x]}$$

$$(\langle\!\langle x\rangle\!\rangle\varphi)^g = \exists p^x_{c_1} \dots \exists p^x_{c_m}. \Big(\varphi_{\text{str}}(x) \wedge (\varphi)^g\Big),$$

$$\text{where } \varphi_{\text{str}}(x) = \mathsf{AG}\Big(\bigvee\nolimits_{c\in\mathsf{Act}} (p^x_c \wedge \bigwedge\nolimits_{c'\neq c} \neg p^x_{c'})\Big)$$

$$(\mathsf{A}\psi)^g = \mathsf{A}(\psi_{\text{out}}(g) \to (\psi)^g)$$

$$\text{where } \psi_{\text{out}}(g) = \mathsf{G}\left(\bigwedge\nolimits_{\mathsf{v}\in\mathsf{V}} \left(p_\mathsf{v} \to \bigvee\nolimits_{\vec{c}\in\mathsf{Act}^{\mathsf{Agt}}} \left(\bigwedge\nolimits_{a\in\mathsf{dom}(f)} p^{f(a)}_{\vec{c}(a)} \wedge \mathsf{X} p_{\Delta(\mathsf{v},\vec{c})}\right)\right)\right)$$

$$(f(\varphi_1, \dots, \varphi_n))^g = f((\varphi_1)^g, \dots, (\varphi_n)^g)$$

and for path formulas:

$$(\varphi)^g = (\varphi)^g \qquad (\mathsf{X}\psi)^g = \mathsf{X}(\psi)^g$$
$$(\psi\mathsf{U}\psi')^g = (\psi)^g\mathsf{U}(\psi')^g \qquad (f(\psi_1,\ldots,\psi_n))^g = f((\psi_1)^g,\ldots,(\psi_n)^g)$$

This translation is identical to that from branching-time SL to QCTL* in all cases, except for the case of functions which is straightforward. To see that it can be safely lifted to the quantitative setting, it suffices to observe the following: since quantification on atomic propositions is restricted in BQCTL$^\star[\mathcal{F}]$ to Boolean values, and atoms in AP_V also have Boolean values, $\varphi_{\text{str}}(x)$ and $\psi_{\text{out}}(\chi)$ always have value 0 or 1 and thus they can play exactly the same role as in the qualitative setting: $\varphi_{\text{str}}(x)$ holds if and only if the atomic propositions $p^x_{c_1},\ldots,p^x_{c_m}$ indeed code a strategy from the current state, and $\psi_{\text{out}}(\chi)$ holds on a branch of $t_{\mathcal{K}_\mathcal{G}}$ if and only if in this branch each agent $a \in \mathsf{dom}(g)$ follows the strategies coded by atoms p^x_c. As a result $\exists p^x_{c_1}\ldots\exists p^x_{c_m}.\Big(\varphi_{\text{str}}(x)\wedge(\varphi)^g\Big)$ maximises over those valuations for the $p^x_{c_i}$ that code for strategies, other valuations yielding value 0. Similarly, formula $\mathsf{A}(\psi_{\text{out}}(g)\to(\psi)^g)$ minimises over branches that represent outcomes of strategies in g, as others yield value 1.

One can now see that the following holds, where φ is an SL$[\mathcal{F}]$ formula.

▶ **Lemma 15.** *Let χ be an assignment and $g : \mathsf{Agt} \rightharpoonup \mathsf{Var}$ such that $\mathsf{dom}(g) = \mathsf{dom}(\chi) \cap \mathsf{Agt}$ and for all $a \in \mathsf{dom}(g)$, $g(a) = x$ implies $\chi(a) = \chi(x)$. Then*

$$[\![\varphi]\!]^{\mathcal{G}}_{\chi}(\rho) = [\![(\varphi)^g]\!]^{t_{\mathcal{K}_\mathcal{G}}}(u_\rho)$$

As a result, the quantitative model-checking problem for an SL$[\mathcal{F}]$ formula φ, a weighted CGS $\mathcal{G}$ and a predicate $P \subseteq [0,1]$ can be solved by computing the BQCTL$^\star[\mathcal{F}]$ formula $\varphi' = (\varphi)^\emptyset$ and the weighted Kripke structure $\mathcal{K}_\mathcal{G}$, and deciding whether $[\![\varphi']\!]^{\mathcal{K}_\mathcal{G}} \in P$, which can be done by Theorem 12. This establishes the upper-bounds in Theorem 6. As in the case of BQCTL$^\star[\mathcal{F}]$, the lower-bounds are obtained by reduction from the model-checking problem for EQ^{k}CTL*. This reduction is an adaptation of the one from QCTL* to ATL with strategy context in [54], and that preserves nesting depth.

▶ Remark 16. Lemma 15 together with Lemma 13 imply that SL$[\mathcal{F}]$ formulas also can take only exponentially many values when a finite domain is fixed for atomic propositions. This justifies the observation of Remark 4 that supremum and infimum in the semantics of SL$[\mathcal{F}]$ can be replaced with maximum and minimum.

6 The case of SL$_{1G}[\mathcal{F}]$

We now study the fragment SL$_{1G}[\mathcal{F}]$, which is the extension to the quantitative setting of the one-goal fragment SL$_{1G}$ of SL [57].

In order to define the syntax, we need to introduce the notions of *quantification prefix* and *binding prefix*. A quantification prefix is a sequence $\wp = \langle\!\langle x_1]\!] \ldots \langle\!\langle x_n]\!]$, where $\langle\!\langle x_i]\!] \in \{\langle\!\langle x_i\rangle\!\rangle, [\![x_i]\!]\}$ is either an existential or universal quantification. For a fixed set of agents $\mathsf{Agt} = \{a_1,\ldots,a_n\}$ a binding prefix is a sequence $\flat = (a_1,x_1),\ldots(a_n,x_n)$, where every agent in Agt occurs exactly once. A combination $\wp\flat$ is *closed* if every variable occurring in $\flat$ occurs in some quantifier of $\wp$. We can now give the definition of SL$_{1G}[\mathcal{F}]$ syntax.

▶ **Definition 17** (SL$_{1G}[\mathcal{F}]$ Syntax)**.** *Let AP be a set of Boolean atomic propositions, and let $\mathcal{F} \subseteq \{f : [0,1]^m \to [0,1] \mid m \in \mathbb{N}\}$ be a set of functions over $[0,1]$. The set of SL$_{1G}[\mathcal{F}]$ formulas is defined by the following grammar:*

$$\varphi := p \mid f(\varphi,\ldots,\varphi) \mid \varphi \mid \mathsf{X}\varphi \mid \varphi\mathsf{U}\varphi \mid \wp\flat\varphi,$$

where $p \in \mathsf{AP}$, $f \in \mathcal{F}$, and $\wp\flat$ is a closed combination of a quantification prefix and of a binding prefix.

Note that all $\mathsf{SL}[\mathcal{F}]$ formulas are sentences, as all strategy variables are quantified immediately before being bound to some agent. The *sentence nesting depth* of an $\mathsf{SL}_{\mathsf{1G}}[\mathcal{F}]$ formula is defined as follows:

- $\mathsf{SntNest}(p) = 0$ for every $p \in \mathsf{AP}$;
- $\mathsf{SntNest}(f(\varphi_1, \dots, \varphi_n)) = \max_{1 \leq i \leq n}\{\mathsf{SntNest}(\varphi_i)\}$;
- $\mathsf{SntNest}(\mathsf{X}\psi) = \mathsf{SntNest}(\psi)$;
- $\mathsf{SntNest}(\psi_1 \mathsf{U} \psi_2) = \max\{\mathsf{SntNest}(\psi_1), \mathsf{SntNest}(\psi_2)\}$.
- $\mathsf{SntNest}(\wp\flat\varphi) = \mathsf{SntNest}(\varphi) + 1$;

Intuitively, the sentence nesting depth measures the number of sentences, i.e., formulas with no free agent or variable, that are nested into each other in the formula.

In order to solve the model-checking problem for $\mathsf{SL}_{\mathsf{1G}}[\mathcal{F}]$, we need the technical notion of *concurrent multi-player parity game* introduced in [56].

▶ **Definition 18.** *A* concurrent multi-player parity game *(CMPG) is a tuple $\mathcal{P} = (\mathsf{Agt}, \mathsf{Act}, S, \mathsf{s}_\iota, \mathsf{p}, \Delta)$, where $\mathsf{Agt} = 0, \dots, n$ is a set of agents indexed with natural numbers, S is a set of states, s_ι is a designated initial state, $\mathsf{p} : S \to \mathbb{N}$ is a priority function, and $\Delta : S \times \mathsf{Act}^{\mathsf{Agt}} \to S$ is a transition function determining the evolution of the game according to the joint actions of the players.*

A CMPG is a game played by players $\mathsf{Agt} = 0, \dots, n$ for an infinite number of rounds. In each round, the players concurrently and independently choose moves, and the current state and the action chosen for each player determine the successor state. In details we have that each player i, with $i \mod 2 = 0$ is part of the existential (even) team; the other players are instead part of the universal (odd) team. Informally, the goal in a CMPG is to check whether there exists a strategy for 0 such that, for each strategy for 1, there exists a strategy for 2, and so forth, such that the induced plays satisfy the parity condition. Then, we say that the existential team wins the game. Otherwise the universal team wins the game.

As shown in [56, Theorem 4.1, Corollary 4.1], one can decide the winners of a CMPG $\mathcal{P} = (\mathsf{Agt}, \mathsf{Act}, \mathsf{S}, \mathsf{s}_\iota, \mathsf{p}, \Delta)$ in time polynomial w.r.t. $|\mathsf{S}|$ and $|\mathsf{Act}|$, and exponential w.r.t. $|\mathsf{Agt}|$ and $k = \max \mathsf{p}$ (the maximal priority).

▶ **Theorem 19.** *The model-checking problem for closed formulas of $\mathsf{SL}_{\mathsf{1G}}[\mathcal{F}]$ is decidable, and 2-EXPTIME-complete.*

Proof. We let $\mathcal{G} = (\mathsf{AP}, \mathsf{Agt}, \mathsf{Act}, \mathsf{V}, \mathsf{v}_\iota, \Delta, \mathsf{w})$ be a WCGS and we consider a formula of the form $\wp\flat\varphi$. We also assume, for simplicity, that $\wp = \langle\langle x_0 \rangle\rangle [\![x_1]\!], \dots, \langle\langle x_k]\!]$, that is, quantifiers perfectly alternate between existential and universal.[4] Note that the formula $\wp\flat\varphi$ is a sentence, therefore the choice of an assignment is useless. Moreover, recall that, by Lemma 15 and in particular Remark 16, the set $V(\wp\flat\varphi)$ of possible values is bounded by $2^{|\wp\flat\varphi|}$.

We proceed by induction on the sentence nesting depth. As base case let $\mathsf{SntNest}(\wp\flat\varphi) = 1$, i.e., there is no occurrence of neither quantifiers nor bindings in φ. Then, φ can be regarded as an $\mathsf{LTL}[\mathcal{F}]$ formula that can be interpreted over paths of the WKS $\mathcal{K} = (\mathsf{AP}, \mathsf{V}, \mathsf{v}_\iota, R, \mathsf{w})$ where $R = \{(\mathsf{v}_1, \mathsf{v}_2) \mid \exists \vec{c} \in \mathsf{Act}^{\mathsf{Agt}}.\ \mathsf{v}_2 = \Delta(\mathsf{v}_1, \vec{c})\}$. Now, thanks to [3, Theorem 3.1], for

[4] To reduce to this case, one can either collapse the agents occurring in the same quantification block, or interleave them with dummy agents quantified with the other modality.

every value $v \in V(\wp\flat\varphi)$, there exists a nondeterministic generalised Büchi word automaton $\mathcal{B}_{\mathcal{K},\varphi,P_v}$, with $P_v = [v, 1]$ that accepts all and only the infinite paths π of $\mathcal{K}$ such that $[\![\varphi]\!]^{\mathcal{K}}(\pi) \in P_v$. Following [61], we can convert $\mathcal{B}_{\mathcal{K},\varphi,P_v}$ into a deterministic parity word automaton $\mathcal{A}_{\mathcal{K},\varphi,P_v} = (\mathsf{V}, Q, q_\iota, \delta, \mathsf{p})$ of size doubly-exponential in the size of φ and index bounded by $2^{|\varphi|}$.

At this point, define the following CMPG $\mathcal{P} = (\mathsf{Agt}', \mathsf{Act}, \mathsf{S}, \mathsf{s}_\iota, \mathsf{p}', \Delta')$ such that

- $\mathsf{Agt}' = \{0, \ldots, k\}$ is a set of agents, one for every variable occurring in $\wp$, ordered in the same way as in $\wp$ itself;
- Act is the set of actions in $\mathcal{G}$;
- $\mathsf{S} = \mathsf{V} \times Q$ is the product of the states of $\mathcal{G}$ and the automaton $\mathcal{A}_{\mathcal{K},\varphi,P_v}$;
- $\mathsf{s}_\iota = (\mathsf{v}_\iota, q_\iota)$ is the pair given by the initial states of $\mathcal{G}$ and $\mathcal{A}_{\mathcal{K},\varphi,t}$, respectively;
- $\mathsf{p}'(\mathsf{v}, q) = \mathsf{p}(q)$ mimics the parity function of $\mathcal{A}_{\mathcal{K},\varphi,P_v}$;
- if $\vec{c} \in \mathsf{Act}^{\mathsf{Agt}'}$, $\Delta'((\mathsf{v}, q), \vec{c}) = (\Delta(\mathsf{v}, \flat(\vec{c})), \delta(q, \mathsf{v}))$ mimics the execution of both $\mathcal{G}$ and $\mathcal{A}_{\mathcal{K},\varphi,P_v}$.

The game emulates two things, one per each component of its state-space. In the first, it emulates a path π generated in $\mathcal{G}$. In the second, it emulates the execution of the automaton $\mathcal{A}_{\mathcal{K},\varphi,P_v}$ when it reads the path π generated in the first component. By construction, it results that every execution $(\pi, \eta) \in \mathsf{V}^\omega \times Q^\omega$ in $\mathcal{P}$ satisfies the parity condition determined by p' if, and only if, $[\![\varphi]\!]^{\mathcal{K}}(\pi) \in P_v$. Moreover, observe that, since $\mathcal{A}_{\mathcal{K},\varphi,P_v}$ is deterministic, for every possible history ρ in $\mathcal{G}$, there is a unique partial run η_ρ that makes the partial execution (ρ, η_ρ) possible in $\mathcal{P}$. This makes the sets of possible strategies $\mathrm{Str}_{\mathcal{G}}(\mathsf{v}_\iota)$ and $\mathrm{Str}_{\mathcal{P}}(\mathsf{s}_\iota)$ in perfect bijection. $\mathcal{P}$ has a winning strategy if and only if $[\![\wp\flat\varphi]\!]^{\mathcal{G}}(\mathsf{v}_\iota) \in P_v$. In order to compute the exact value of $[\![\wp\flat\varphi]\!]^{\mathcal{G}}(\mathsf{v}_\iota)$, we repeat the procedure described above for every $v \in V(\wp\flat\varphi)$ and take the maximum v of those for which $[\![\wp\flat\varphi]\!]^{\mathcal{G}}(\mathsf{v}_\iota) \in P_v$.

For the induction case, assume we can compute the satisfaction value of every $\mathsf{SL_{1G}}[\mathcal{F}]$ formula with sentence nesting depth at most n, and let $\mathsf{SntNest}(\wp\flat\varphi) = n + 1$. Observe that, for every subsentence $\wp'\flat'\varphi'$ of $\wp\flat\varphi$, we have that $\mathsf{SntNest}(\wp'\flat'\varphi') \leq n$ and so, by induction hypothesis, we can compute $[\![\wp'\flat'\varphi']\!]^{\mathcal{G}}(\mathsf{v})$ for every $\mathsf{v} \in \mathsf{V}$. Now, introduce a fresh atomic proposition $p_{(\wp'\flat'\varphi')}$ whose weight in $\mathcal{G}$ is defined as $\mathsf{w}(v)(p_{(\wp'\flat'\varphi')}) = [\![\wp'\flat'\varphi']\!]^{\mathcal{G}}(v)$ and a set of fresh atomic propositions p_{v}, one for every $\mathsf{v} \in \mathsf{V}$, whose weights in $\mathcal{G}$ are defined as $\mathsf{w}(\mathsf{v})(p_{\mathsf{v}}) = 1$ and $\mathsf{w}(\mathsf{v})(p_{\mathsf{v}'}) = 0$ if $\mathsf{v} \neq \mathsf{v}'$. Now, consider the Boolean formula

$$\Phi(\wp'\flat'\varphi') = \bigvee_{\mathsf{v} \in \mathsf{V}} (p_{\mathsf{v}} \wedge p_{(\wp'\flat'\varphi')})$$

Observe that every disjunct is a conjunction of the form $p_{\mathsf{v}} \wedge p_{(\wp'\flat'\varphi')}$, whose satisfaction value on a state v' is the minimum among the weights of p_{v} and $p_{(\wp'\flat'\varphi')}$. This can be either 0, if $\mathsf{v} \neq \mathsf{v}'$ or $[\![\wp'\flat'\varphi']\!]^{\mathcal{G}}(\mathsf{v})$, if $\mathsf{v} = \mathsf{v}'$. Now, the big disjunction takes the maximum among them. Therefore, we obtain that $[\![\Phi(\wp'\flat'\varphi')]\!]^{\mathcal{G}}(\mathsf{v}) = [\![\wp'\flat'\varphi']\!]^{\mathcal{G}}(\mathsf{v})$. This allows us to replace every occurrence of $\wp'\flat'\varphi'$ in φ with the Boolean combination $\Phi(\wp'\flat'\varphi')$, making the resulting formula to be of sentence nesting depth 1. Thus, we can apply the procedure described in the base case, to compute $[\![\wp\flat\varphi]\!]^{\mathcal{G}}(\mathsf{v})$.

Regarding the complexity, note that the size of $\mathcal{P}$ is $|\mathsf{V}| \cdot |Q|$ that is in turn linear with respect to the size of $\mathcal{G}$ and doubly exponential in the size of φ. This is due to the fact that the automaton $\mathcal{A}_{\mathcal{K},\varphi,P_v}$ results from the construction of the NGBW $\mathcal{B}_{\mathcal{K},\varphi,P_v}$, of size singly exponential in $|\varphi|$ and the transformation to a DPW, that adds up another exponential to the construction. On the other hand, the number of priorities in $\mathcal{P}$ is only singly exponential in $|\varphi|$, and it is due to the fact that the transformation from NGBW to DPW requires a singly exponential number of priorities. Therefore, the CMPG $\mathcal{P}$ can be solved in time polynomial w.r.t. the size $\mathcal{G}$ and double exponential in $|\varphi|$. Such 2-EXPTIME procedure is executed a number of time exponential in φ, which is still 2-EXPTIME.

Hardness follows from that of SL_{1G} [57]. ◀

7 Future work

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 [31]. 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 [46, 35] or tax increase [33], 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 $[0,1]$.

References

1 Natasha Alechina, Brian Logan, Nguyen Hoang Nga, and Abdur Rakib. Resource-bounded alternating-time temporal logic. In Wiebe van der Hoek, Gal A. Kaminka, Yves Lespérance, Michael Luck, and Sandeep Sen, editors, *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'10)*, pages 481–488. International Foundation for Autonomous Agents and Multiagent Systems, May 2010.

2 Shaull Almagor, Udi Boker, and Orna Kupferman. Discounting in LTL. In Erikz Ábrahám and Klaus Havelund, editors, *Proceedings of the 20th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'14)*, volume 8413 of *Lecture Notes in Computer Science*, pages 424–439. Springer-Verlag, April 2014. doi:10.1007/978-3-642-54862-8_37.

3 Shaull Almagor, Udi Boker, and Orna Kupferman. Formally reasoning about quality. *Journal of the ACM*, 63(3):24:1–24:56, September 2016. doi:10.1145/2875421.

4 Shaull Almagor and Orna Kupferman. High-quality synthesis against stochastic environments. In Jean-Marc Talbot and Laurent Regnier, editors, *Proceedings of the 25th EACSL Annual Conference on Computer Science Logic (CSL'16)*, volume 62 of *Leibniz International Proceedings in Informatics*, pages 28:1–28:17. Leibniz-Zentrum für Informatik, September 2016. doi:10.4230/LIPIcs.CSL.2016.28.

5 Shaull Almagor, Orna Kupferman, and Giuseppe Perelli. Synthesis of controllable Nash equilibria in quantitative objective games. In Jérôme Lang, editor, *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, pages 35–41. IJCAI organization, July 2018. doi:10.24963/ijcai.2018/5.

6 Shaull Almagor, Orna Kupferman, Jan Oliver Ringert, and Yaron Velner. High-quality synthesis against stochastic environments. In Rupak and Majumdar, editor, *Proceedings of the 29th International Conference on Computer Aided Verification (CAV'17)*, volume 10427 of *Lecture Notes in Computer Science*, pages 353–374. Springer-Verlag, July 2017. doi:10.1007/978-3-319-63390-9_19.

7 Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, September 2002. doi:10.1145/585265.585270.

8 Benjamin Aminof, Vadim Malvone, Aniello Murano, and Sasha Rubin. Graded modalities in strategy logic. *Information and Computation*, 261(4):634–649, August 2018. doi:10.1016/j.ic.2018.02.021.

9 Eugene Asarin, Oded Maler, Amir Pnueli, and Joseph Sifakis. Controller synthesis for timed automata. In *Proceedings of the 5th IFAC Cconference on System Structure and Control (SSC'98)*, pages 469–474. Elsevier, July 1998.

10 Christel Baier, Tomáš Brázdil, Marcus Größer, and Antonín Kučera. Stochastic game logic. *Acta Informatica*, 49(4):203–224, June 2012. doi:10.1007/s00236-012-0156-0.

11 Raphaël Berthon, Bastien Maubert, Aniello Murano, Sasha Rubin, and Moshe Y. Vardi. Strategy logic with imperfect information. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12, 2017. URL: https://doi.org/10.1109/LICS.2017.8005136, doi:10.1109/LICS.2017.8005136.

12 Dietmar Berwanger. Admissibility in infinite games. In *Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science (STACS'07)*, volume 4393 of *Lecture Notes in Computer Science*, pages 188–199. Springer, 2007. URL: https://doi.org/10.1007/978-3-540-70918-3_17, doi:10.1007/978-3-540-70918-3_17.

13 Udi Boker, Krishnendu Chatterjee, Thomas A. Henzinger, and Orna Kupferman. Temporal specifications with accumulative values. *ACM Transactions on Computational Logic*, 15(4):27:1–27:25, August 2014. doi:10.1145/2629686.

14 Benedikt Bollig, Normann Decker, and Martin Leucker. Frequency linear-time temporal logic. In Tiziana Margaria, Zongyang Qiu, and Hongli Yang, editors, *Proceedings of the 6th International Symposium on Theoretical Aspects of Software Engineering (TASE'12)*, pages 85–92. IEEE Comp. Soc. Press, July 2012. doi:10.1109/TASE.2012.43.

15 Patricia Bouyer, Uli Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, and Jiří Srba. Infinite runs in weighted timed automata with energy constraints. In Franck Cassez and Claude Jard, editors, *Proceedings of the 6th International Conferences on Formal Modelling and Analysis of Timed Systems (FORMATS'08)*, volume 5215 of *Lecture Notes in Computer Science*, pages 33–47. Springer-Verlag, September 2008. doi:10.1007/978-3-540-85778-5_4.

16 Patricia Bouyer, Patrick Gardy, and Nicolas Markey. Weighted strategy logic with boolean goals over one-counter games. In Prahladh Harsha and G. Ramalingam, editors, *Proceedings of the 35th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'15)*, volume 45 of *Leibniz International Proceedings in Informatics*, pages 69–83. Leibniz-Zentrum für Informatik, December 2015. doi:10.4230/LIPIcs.FSTTCS.2015.69.

17 Patricia Bouyer, Orna Kupferman, Nicolas Markey, Bastien Maubert, Nello Murano, and Giuseppe Perelli. Reasoning about quality and fuzziness of strategic behaviours. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*. IJCAI organization, 2019.

18 Patricia Bouyer, Nicolas Markey, and Raj Mohan Matteplackel. Averaging in LTL. In Paolo Baldan and Daniele Gorla, editors, *Proceedings of the 25th International Conference on Concurrency Theory (CONCUR'14)*, volume 8704 of *Lecture Notes in Computer Science*, pages 266–280. Springer-Verlag, September 2014. doi:10.1007/978-3-662-44584-6_19.

19 Romain Brenguier. Robust equilibria in mean-payoff games. In *Proceedings of Foundations of Software Science and Computation Structures - 19th International Conference (FOSSACS'16)*, volume 9634 of *Lecture Notes in Computer Science*, pages 217–233. Springer, 2016. URL: https://doi.org/10.1007/978-3-662-49630-5_13, doi:10.1007/978-3-662-49630-5_13.

20 Romain Brenguier, Guillermo A. Pérez, Jean-François Raskin, and Mathieu Sassolas. Admissibility in quantitative graph games. In S. Akshay, Akash Lal, Saket Saurabh, and Sandeep Sen, editors, *Proceedings of the 36th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'16)*, Leibniz International Proceed-

ings in Informatics, pages 42:1–42:14. Leibniz-Zentrum für Informatik, December 2016. doi:10.4230/LIPIcs.FSTTCS.2016.42.

21 Romain Brenguier, Jean-François Raskin, and Ocan Sankur. Assume-admissible synthesis. *Acta Informatica*, 54(1):41–83, 2017. URL: https://doi.org/10.1007/s00236-016-0273-2, doi:10.1007/s00236-016-0273-2.

22 Thomas Brihaye, Véronique Bruyère, and Julie De Pril. On equilibria in quantitative games with reachability/safety objectives. *Theory of Computing Systems*, 54(2):150–189, February 2014.

23 Thomas Brihaye, François Laroussinie, Nicolas Markey, and Ghassan Oreiby. Timed concurrent game structures. In Luís Caires and Vasco T. Vasconcelos, editors, *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR'07)*, volume 4703 of *Lecture Notes in Computer Science*, pages 445–459. Springer-Verlag, September 2007. doi:10.1007/978-3-540-74407-8_30.

24 Véronique Bruyère, Noëmie Meunier, and Jean-François Raskin. Secure equilibria in weighted games. In *Proceedings of the Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic and the 29th Annual ACM/IEEE Symposium on Logic In Computer Science (CSL/LICS'14)*. ACM Press, July 2014. doi:10.1145/2603088.2603109.

25 Nils Bulling and Valentin Goranko. How to be both rich and happy: Combining quantitative and qualitative strategic reasoning about multi-player games (extended abstract). In *Proceedings of the 1st International Workshop on Strategic Reasoning (SR'13)*, volume 112 of *Electronic Proceedings in Theoretical Computer Science*, pages 33–41, March 2013. doi:10.4204/EPTCS.112.8.

26 Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, and Mariëlle Stoelinga. Resource interfaces. In Rajeev Alur and Insup Lee, editors, *Proceedings of the 3rd International Conference on Embedded Software (EMSOFT'03)*, volume 2855 of *Lecture Notes in Computer Science*, pages 117–133. Springer-Verlag, October 2003.

27 Krishnendu Chatterjee, Laurent Doyen, Emmanuel Filiot, and Jean-François Raskin. Doomsday equilibria for omega-regular games. In Kenneth L. McMillan and Xavier Rival, editors, *Proceedings of the 15th International Workshop on Verification, Model Checking, and Abstract Interpretation (VMCAI'14)*, volume 8318 of *Lecture Notes in Computer Science*, pages 78–97. Springer-Verlag, January 2014. doi:10.1007/978-3-642-54013-4_5.

28 Krishnendu Chatterjee, Thomas A Henzinger, and Marcin Jurdziński. Games with secure equilibria. *Theoretical Computer Science*, 365(1-2):67–82, 2006.

29 Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Strategy logic. *Information and Computation*, 208(6):677–693, June 2010. doi:10.1016/j.ic.2009.07.004.

30 Taolue Chen and Jian Lu. Probabilistic alternating-time temporal logic and model checking algorithm. In J. Lei, editor, *Proceedings of the 4th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'07)*, pages 35–39. IEEE Comp. Soc. Press, August 2007. doi:10.1109/FSKD.2007.458.

31 Hana Chockler, Orna Kupferman, and Moshe Y. Vardi. Coverage metrics for formal verification. *Software Tools for Technology Transfer*, 8(4-5):373–386, 2006.

32 Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In Dexter C. Kozen, editor, *Proceedings of the 3rd Workshop on Logics of Programs (LOP'81)*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer-Verlag, 1982. doi:10.1007/BFb0025774.

33 Richard Cole, Yevgeniy Dodis, and Tim Roughgarden. How much can taxes help selfish routing? *J. Comput. Syst. Sci.*, 72(3):444–467, 2006.

34 Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar, and Mariëlle Stoelinga. Model checking discounted temporal properties. *Theoretical Computer Science*, 345(1):139–170, November 2005.

35 Filipe Costa de Souza and Leandro Chaves Rego. Mixed equilibrium in 2×2 normal form games: when burning money is rational. *Pesquisa Operacional*, 36:81–99, 04 2016.

36 Dario Della Monica and Aniello Murano. Parity-energy ATL for qualitative and quantitative reasoning in MAS. In Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar, editors, *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'18)*, pages 1441–1449. International Foundation for Autonomous Agents and Multiagent Systems, July 2018.

37 Andrzej Ehrenfeucht and Jan Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8(2):109–113, June 1979. doi:10.1007/BF01768705.

38 E. Allen Emerson and Joseph Y. Halpern. "Sometimes" and "not never" revisited: On branching versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178, January 1986.

39 E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science (FOCS'91)*, pages 368–377. IEEE Comp. Soc. Press, October 1991. doi:10.1109/SFCS.1991.185392.

40 E. Allen Emerson and Chin-Laung Lei. Modalities for model checking: Branching time logic strikes back. *Science of Computer Programming*, 8:275–306, 1987.

41 Nathanaël Fijalkow, Bastien Maubert, Aniello Murano, and Sasha Rubin. Quantifying bounds in strategy logic. In *Proceedings of the 27th EACSL Annual Conference on Computer Science Logic (CSL'18)*, volume 119 of *LIPIcs*, pages 23:1–23:23, 2018. URL: https://doi.org/10.4230/LIPIcs.CSL.2018.23, doi:10.4230/LIPIcs.CSL.2018.23.

42 Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer-Verlag, 1997.

43 Dana Fisman, Orna Kupferman, and Yoad Lustig. Rational synthesis. In Javier Esparza and Rupak Majumdar, editors, *Proceedings of the 16th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'10)*, volume 6015 of *Lecture Notes in Computer Science*, pages 190–204. Springer-Verlag, March 2010. doi:10.1007/978-3-642-12002-2_16.

44 Achille Frigeri, Liliana Pasquale, and Paola Spoletini. Fuzzy time in linear temporal logic. *ACM Transactions on Computational Logic (TOCL)*, 15(4):30, 2014.

45 David Harel and Amir Pnueli. On the development of reactive systems. In *Logics and Models of Concurrent Systems*, volume F-13 of *NATO Advanced Summer Institutes*, pages 477–498. Springer, 1985.

46 Jason D. Hartline and Tim Roughgarden. Optimal mechanism design and money burning. In *Proceedings of the 40th ACM Symposium on Theory of Computing (STOC'08)*, pages 75–84, 2008.

47 Thomas A. Henzinger and Vinayak S. Prabhu. Timed alternating-time temporal logic. In Eugene Asarin and Patricia Bouyer, editors, *Proceedings of the 4th International Conferences on Formal Modelling and Analysis of Timed Systems (FORMATS'06)*, volume 4202 of *Lecture Notes in Computer Science*, pages 1–17. Springer-Verlag, September 2006.

48 Wojciech Jamroga, Beata Konikowska, and Wojciech Penczek. Multi-valued verification of strategic ability. In Catholijn M. Jonker, Stacy Marsella, John Thangarajah, and Karl Tuyls, editors, *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'16)*, pages 1180–1189. International Foundation for Autonomous Agents and Multiagent Systems, May 2016.

49 Orna Kupferman, Giuseppe Perelli, and Moshe Y. Vardi. Synthesis with rational environments. *Ann. Math. Artif. Intell.*, 78(1):3–20, 2016. doi:10.1007/s10472-016-9508-8.

50 Orna Kupferman, Moshe Y. Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, 2000.

51 Orna Kupferman, Moshe Y. Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, 2000.

52 Khaled Ben Lamine and Froduald Kabanza. Using fuzzy temporal logic for monitoring behavior-based mobile robots. In *Proc. of IASTED Int. Conf. on Robotics and Applications*, pages 116–121, 2000.

53 François Laroussinie and Nicolas Markey. Quantified CTL: Expressiveness and complexity. *Logical Methods in Computer Science*, 10(4), 2014. URL: http://dx.doi.org/10.2168/LMCS-10(4:17)2014, doi:10.2168/LMCS-10(4:17)2014.

54 François Laroussinie and Nicolas Markey. Augmenting ATL with strategy contexts. *Inf. Comput.*, 245:98–123, 2015. URL: http://dx.doi.org/10.1016/j.ic.2014.12.020, doi:10.1016/j.ic.2014.12.020.

55 François Laroussinie, Nicolas Markey, and Ghassan Oreiby. Model-checking timed ATL for durational concurrent game structures. In Eugene Asarin and Patricia Bouyer, editors, *Proceedings of the 4th International Conferences on Formal Modelling and Analysis of Timed Systems (FORMATS'06)*, volume 4202 of *Lecture Notes in Computer Science*, pages 245–259. Springer-Verlag, September 2006. doi:10.1007/11867340_18.

56 Vadim Malvone, Aniello Murano, and Loredana Sorrentino. Concurrent multi-player parity games. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'16)*, pages 689–697, 2016.

57 Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. What makes ATL* decidable? A decidable fragment of strategy logic. In Maciej Koutny and Irek Ulidowski, editors, *Proceedings of the 23rd International Conference on Concurrency Theory (CONCUR'12)*, volume 7454 of *Lecture Notes in Computer Science*, pages 193–208. Springer-Verlag, September 2012.

58 Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. Reasoning about strategies: On the model-checking problem. *ACM Transactions on Computational Logic*, 15(4):34:1–34:47, August 2014. doi:10.1145/2631917.

59 Noam Nisan and Amir Ronen. Algorithmic mechanism design. In *Proceedings of the 31st ACM Symposium on Theory of Computing*, pages 129–140, 1999.

60 Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

61 Nir Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. *Logical Methods in Computer Science*, 3(3):1–21, 2007.

62 Amir Pnueli. The temporal semantics of concurrent programs. *Theoretical Computer Science*, 13:45–60, 1981.

63 Amir Pnueli and Roni Rosner. On the synthesis of a reactive module. In *Conference Record of the 16th ACM Symposium on Principles of Programming Languages (POPL'89)*, pages 179–190. ACM Press, January 1989.

64 Lloyd S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, October 1953. doi:10.1073/pnas.39.10.1095.

65 Michael Ummels. Stochastic Multiplayer Games: Theory and Algorithms. Master's thesis, RWTH Aachen, Germany, January 2010.

66 Steen Vester. On the complexity of model-checking branching and alternating-time temporal logics in one-counter systems. In Bernd Finkbeiner, Geguang Pu, and Lijun Zhang, editors, *Proceedings of the 13th International Symposium on Automated Technology for Verification and Analysis (ATVA'15)*, volume 9364 of *Lecture Notes in Computer Science*, pages 361–377. Springer-Verlag, October 2015. doi:10.1007/978-3-319-24953-7_27.