Graded Strategy Logic: Reasoning about Uniqueness of Nash Equilibria

Benjamin Aminof
Technische Universitat Wien, Austria
benj@forsyte.tuwien.ac.at

Vadim Malvone, Aniello Murano, Sasha Rubin
Università degli Studi di Napoli Federico II, Italy
{vadim.malvone, aniello.murano, sasha.rubin}@unina.it

ABSTRACT

Strategy Logic (SL) is a well established formalism for strategic reasoning in multi-agent systems. In a nutshell, SL is built over LTL and treats strategies as first-order objects that can be associated with agents by means of a binding operator. In this work we introduce Graded Strategy Logic (GRADED-SL), an extension of SL by graded quantifiers over tuples of strategy variables such as "there exist at least g different tuples $(x_1, ..., x_n)$ of strategies". We study the model-checking problem of GRADED-SL and prove that it is no harder than for SL, i.e., it is non-elementary in the quantifier rank.

We show that GRADED-SL allows one to count the number of different strategy profiles that are Nash equilibria (NE), or subgame-perfect equilibria (SPE). By analyzing the structure of the specific formulas involved, we conclude that the important problems of checking for the existence of a unique NE or SPE can both be solved in 2EXPTIME, which is not harder than merely checking for the existence of such equilibria.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence —*Multiagent Systems*

General Terms

Theory, Verification

Keywords

Strategic logics; Graded modalities; Nash equilibrium

1. INTRODUCTION

Strategy Logic (SL) is a powerful formalism for reasoning about strategies in multi-agent systems [43, 44]. Strategies tell an agent what to do — they are functions that prescribe an action based on the history. The fundamental idea in SL is to treat strategies as first-order object. A strategy x can be quantified existentially $\langle\langle x\rangle\rangle$ (read: there exists a strategy x) and universally $[[x]]$ (read: for all strategies x). Furthermore, strategies are not intrinsically glued to specific agents: the *binding* operator (α, x) allows one to bind an agent α to the strategy x. SL strictly subsumes several other logics for strategic reasoning including the well known ATL and ATL* [3]. Being a very powerful logic, SL can directly express many solution concepts [9, 19, 32, 35, 43] among which that a strategy profile $\overline{x}$ is a Nash equilibrium, and thus also the existence of a Nash equilibrium (NE).

Nash equilibrium is one of the most important concepts in game theory, forming the basis of much of the recent fundamental work in multi-agent decision making. A challenging and important aspect is to establish whether a game admits a *unique* NE [2, 20, 48]. This problem impacts on the predictive power of NE since, in case there are multiple equilibria, the outcome of the game cannot be uniquely pinned down [21, 49, 57]. Unfortunately, uniqueness has mainly been established either for special cost functions [2], or for very restrictive game topologies [47]. More specifically, uniqueness of NE in game theory is proved with procedures that are separated from the ones that check for its existence [2]; these procedures require various transformations of the best response functions of individual contributors [28, 29, 53], and there is no general theory that can be applied to different application areas [2].

In this paper, we address and solve the problem of expressing the uniqueness of certain solution concepts (and NE in particular) in a principled and elegant way. We introduce an extension of SL called GRADED-SL and study its model-checking problem. More specifically, we extend SL by replacing the quantification $\langle\langle x\rangle\rangle$ and $[[x]]$ over strategy variables with *graded quantification over tuples of strategy variables*: $\langle\langle x_1, \ldots, x_n\rangle\rangle^{\geq g}$ (read $\langle\langle x_1, \ldots, x_n\rangle\rangle^{\geq g}$ as "there exist at least g different tuples $(x_1, \ldots, x_n)$ of strategies") and its dual $[[x_1, \ldots, x_n]]^{<g}$ ($g \in \mathbb{N}$). Here, two strategies are different if they disagree on some history. That is, we count strategies syntactically as is usual in graded extensions of modal and description logics [6, 12, 15, 30, 36]. The key for expressing uniqueness of NE is the combination of quantifying over tuples (instead of singleton variables), and adding counting (in the form of graded modalities).

We address the model-checking problem for GRADED-SL and prove that it has the same complexity as SL, i.e., it is non-elementary in the nesting depth of quantifiers. In particular, we show that model checking GRADED-SL formulas with a nesting depth $k > 0$ of blocks of quantifiers (a block of quantifiers is a maximally-consecutive sequence of quantifiers of the same type, i.e., either all existential, or all universal) is in $(k+1)$EXPTIME, and that for the special case where the formula starts with a block of quantifiers, it is in kEXPTIME. Since many natural formulas contain a very small number of

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

quantifiers, the complexity of the model-checking problem is not as bad as it seems. Specifically, several solution concepts can be expressed as SL formulas with a small number of quantifiers [9, 19, 32, 35, 43]. Since the existence of a NE, and the fact that there is at most one NE, can be expressed in GRADED-SL using simple formulas we are able to conclude that the problem of checking the uniqueness of a NE can be solved in 2EXPTIME. Previously, it was only known that existence of NE can be checked in 2EXPTIME [32, 43], and indeed it is 2EXPTIME-complete [4, 32, 50]. Thus, GRADED-SL is the first logic that can solve the existence and uniqueness of NE (as well as many other solution concepts) in a uniform way.

SL has a few natural syntactic fragments, the most powerful of which is called Nested-Goal SL which can express NE [43]. Similarly, we define GRADED-SL *Nested-Goal*, a syntactic fragment of GRADED-SL. The Nested-Goal restriction encompasses formulas in a special prenex normal form with a particular nested temporal structure that restricts the application of both strategy quantifiers and agent bindings (see Section 2 for details). We show that the graded extension of Nested-Goal SL has the same model-checking complexity, i.e., non-elementary in the *alternation number* of the quantifiers appearing in the formula (the alternation number is, roughly speaking, the maximum number of existential/universal quantifier switches [43]). Since uniqueness of NE can be expressed in Nested-Goal GRADED-SL using alternation one, we get an alternative proof for checking the NE uniqueness in 2EXPTIME.

We exemplify our definition and our automata-theoretic algorithm for the model-checking problem with a large number of applications. In particular, we use it for reasoning about repeated one-shot games such as the *iterated prisoner's dilemma* (IPD) [51]. The prisoner's dilemma game is a popular metaphor for the problem of stable cooperation and has been widely used in several application domains [8]. In the classic definition, it consists of two players, each of them has an option to defect or collaborate. More involved is the IPD in which the process is repeated and one can model reactive strategies in continuous play. The IPD has become a standard model for the evolution of cooperative behaviour within a community of egoistic agents, frequently cited for implications in both sociology and biology (see [8]). Evaluating the existence of NE in an IPD and, even more, its uniqueness, is a very challenging and complicated question due to the rich set of strategies such a game can admit [8, 16, 17]. In particular, such infinite-duration games need to be supported by more complex solution concepts such as *subgame-perfect* equilibrium [27, 41, 55]. Thanks to GRADED-SL and the related model-checking result, we get that checking the uniqueness of a NE in an IPD can be solved in 2EXPTIME.

Related work. The importance of solution concepts, verifying a unique equilibrium, and the relationship with logics for strategic reasoning is discussed above. We now give some highlights from the long and active investigation of graded modalities in the formal verification community.

Graded modalities were first studied in modal logic [26] and then exported to the field of *knowledge representation* to allow quantitative bounds on the set of individuals satisfying a given property. Specifically, they were considered as *counting quantifiers* in first-order logics [31] and *number restrictions* in *description logics* [34]. *Graded μ-calculus*, in which immediate-successor accessible worlds are counted, was introduced to reason about graded modal logic with fixed-point operators [36]. Recently, the notion of graded modalities was extended to count the number of paths in the branching-time temporal logic formulas CTL and CTL* [7, 10]. In the verification of reactive systems, we mention two orthogonal approaches: module checking for graded μ-calculus [6, 25] and an extension of ATL by graded path modalities [24].

The work closest to ours is [40]: also motivated by counting NE, it introduces a graded extension of SL, called GSL. In contrast with our work, GSL has a very intricate way of counting strategies: it makes use of a semantic definition of strategies being equivalent, and counting in which equivalent strategies are counted as a single strategy. While this approach has been proved to be sound and general, it heavily complicates the model-checking problem. Indeed, only a very weak fragment of GSL has been solved in [40] by exploiting an *ad hoc* solution that does not seem to be easily scalable to (all of) GSL. Precisely, the fragment investigated there is the vanilla restriction of the graded version of one-goal SL [42]. There is a common belief that the one-goal fragment is not powerful enough to express the existence of a Nash Equilibrium in concurrent games. The smallest fragment that is know to be able to represent this is the so called Boolean-goal Strategy Logic, whose graded extension (in the GSL sense) has no known solution.[1]

Outline. The sequel of the paper is structured as follows. In Section 2 we introduce GRADED-SL and provide some preliminary related concepts. In Section 3 we address the model-checking problem for GRADED-SL and its fragments. We conclude with Section 4 in which we have a discussion and suggestions for future work.

2. GRADED STRATEGY LOGIC

In this section we introduce Graded Strategy Logic, which we call GRADED-SL for short.

2.1 Models

Sentences of GRADED-SL are interpreted over *concurrent game structures*, just as for ATL and SL [3, 43].

DEFINITION 2.1. *A* concurrent game structure *(*CGS*) is a tuple* $\mathcal{G} \triangleq \langle \mathrm{AP}, \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, s_I, \mathsf{ap}, \mathsf{tr}\rangle$, *where* AP, Ag, Ac, *and* St *are the sets of* atomic propositions, agents, actions *and* states, *respectively,* $s_I \in \mathrm{St}$ *is the* initial state, *and* $\mathsf{ap} : \mathrm{St} \to 2^{\mathrm{AP}}$ *is the* labeling function *mapping each state to the set of atomic propositions true in that state. Let* $\mathrm{Dc} \triangleq \mathrm{Ag} \to \mathrm{Ac}$ *be the set of* decisions, *i.e., functions describing the choice of an action by every agent. Then,* $\mathsf{tr} : \mathrm{Dc} \to (\mathrm{St} \to \mathrm{St})$ *denotes the* transition function *mapping every decision* $\delta \in \mathrm{Dc}$ *to a function* $\mathsf{tr}(\delta) : \mathrm{St} \to \mathrm{St}$.

We will usually take the set Ag of agents to be $\{\alpha_1, \ldots, \alpha_n\}$.

A *path (from s)* is a finite or infinite non-empty sequence of states $s_1 s_2 \ldots$ such that $s = s_1$ and for every i there exists a decision δ with $\mathsf{tr}(\delta)(s_i) = s_{i+1}$. The set of paths starting

[1] In [33] it has been shown that, in the restricted case of turn-based structures it is possible to express the existence of Nash equilibria in m^{-}ATL* [45], a memory-full variant of ATL* (hence included in one-goal SL), but exponentially more succinct — and thus with a much more expensive model-checking algorithm. As also the authors in [33] state, it is not clear how to extend this result to the concurrent setting, even in the two player case.

with s is denoted $\mathrm{Pth}(s)$. The set of finite paths from s, called the *histories (from s)*, is denoted $\mathrm{Hst}(s)$. A *strategy (from s)* is a function $\sigma \in \mathrm{Str}(s) \triangleq \mathrm{Hst}(s) \to \mathrm{Ac}$ that prescribes which action has to be performed given a certain history. We write $\mathrm{Pth}, \mathrm{Hst}, \mathrm{Str}$ for the set of all paths, histories, and strategies (no matter where they start).

We use the standard notion of equality between strategies, [38], i.e., $\sigma_1 = \sigma_2$ iff for all $\rho \in \mathrm{Hst}$, $\sigma_1(\rho) = \sigma_2(\rho)$. This extends to equality between two n-tuples of strategies in the natural way, i.e., coordinate-wise.

2.2 Syntax

We describe the syntax as well as related basic concepts. GRADED-SL extends SL by replacing the classic singleton strategy quantifiers $\langle\langle x \rangle\rangle$ and $[[x]]$ with the graded (tupled) quantifiers $\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g}$ and $[[x_1, \ldots, x_n]]^{<g}$, respectively, where each x_i belongs to a countable set of variables Vr and $g \in \mathbb{N}$ is called the *degree* of the quantifier. Intuitively, these are read as "there exist at least g tuples of strategies $(x_1, \ldots, x_n)$" and "all but less than g many tuples of strategies", respectively. The syntax (α, x) denotes a *binding* of the agent α to the strategy x. The syntax of GRADED-SL is:

DEFINITION 2.2. GRADED-SL formulas *are built inductively by means of the following grammar, where* $p \in \mathrm{AP}$, $\alpha \in \mathrm{Ag}$, $x, x_1, \ldots x_n \in \mathrm{Vr}$ *such that* $x_i \neq x_j$ *for* $i \neq j$, *and* $g, n \in \mathbb{N}$*:*

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g}\varphi \mid (\alpha, x)\varphi.$$

Notation. For the rest of the paper, whenever we write $\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g}$ we also mean that $x_i \neq x_j$ for $i \neq j$.

Shorthands are derived as usual. Specifically, $\mathsf{true} \triangleq p \vee \neg p$, $\mathsf{false} \triangleq \neg\mathsf{true}$, $\mathsf{F}\,\varphi \triangleq \mathsf{true}\,\mathsf{U}\,\varphi$, and $\mathsf{G}\,\varphi \triangleq \neg\,\mathsf{F}\,\neg\varphi$. Moreover, the graded universal operator corresponds to the dual of the existential one, i.e., $[[x_1, \ldots, x_n]]^{<g}\varphi \triangleq \neg\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g}\neg\varphi$. A *placeholder* refers to an agent or a variable. In order to define the semantics, we first define the concept of free placeholders in a formula. Intuitively, an agent or variable is free in φ if it does not have a strategy associated with it (either by quantification or binding) but one is required in order to ascertain if φ is true or not. The definition mimics that for SL [43]. It is included here both for completeness and because it is important for defining the model-checking procedure, in particular for the encoding of strategies as trees (Definition 3.2).

DEFINITION 2.3. *The set of* free agents *and* free variables *of a* GRADED-SL *formula* φ *is given by the function* $\mathsf{free} : \text{GRADED-SL} \to 2^{\mathrm{Ag} \cup \mathrm{Vr}}$ *defined as follows:*

- $\mathsf{free}(p) \triangleq \emptyset$, *where* $p \in \mathrm{AP}$*;*
- $\mathsf{free}(\neg\varphi) \triangleq \mathsf{free}(\varphi)$*;*
- $\mathsf{free}(\varphi_1 \vee \varphi_2) \triangleq \mathsf{free}(\varphi_1) \cup \mathsf{free}(\varphi_2)$*;*
- $\mathsf{free}(\mathsf{X}\,\varphi) \triangleq \mathrm{Ag} \cup \mathsf{free}(\varphi)$*;*
- $\mathsf{free}(\varphi_1\,\mathsf{U}\,\varphi_2) \triangleq \mathrm{Ag} \cup \mathsf{free}(\varphi_1) \cup \mathsf{free}(\varphi_2)$*;*
- $\mathsf{free}(\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g}\varphi) \triangleq \mathsf{free}(\varphi) \setminus \{x_1, \ldots, x_n\}$*;*
- $\mathsf{free}((\alpha, x)\varphi) \triangleq \mathsf{free}(\varphi)$, *if* $\alpha \notin \mathsf{free}(\varphi)$, *where* $\alpha \in \mathrm{Ag}$ *and* $x \in \mathrm{Vr}$*;*
- $\mathsf{free}((\alpha, x)\varphi) \triangleq (\mathsf{free}(\varphi) \setminus \{\alpha\}) \cup \{x\}$, *if* $\alpha \in \mathsf{free}(\varphi)$, *where* $\alpha \in \mathrm{Ag}$ *and* $x \in \mathrm{Vr}$.

A formula φ *without free agents (resp., variables), i.e., with* $\mathsf{free}(\varphi) \cap \mathrm{Ag} = \emptyset$ *(resp.,* $\mathsf{free}(\varphi) \cap \mathrm{Vr} = \emptyset$*), is called* agent-closed *(resp.,* variable-closed*). If* φ *is both agent- and variable-closed, it is called a* sentence.

Another important concept that characterizes the syntax of GRADED-SL is the *alternation number* of quantifiers, i.e., the maximum number of quantifier switches $\langle\langle\cdot\rangle\rangle[[\cdot]]$, $[[\cdot]]\langle\langle\cdot\rangle\rangle$, $\langle\langle\cdot\rangle\rangle\neg\langle\langle\cdot\rangle\rangle$, or $[[\cdot]]\neg[[\cdot]]$ that binds a tuple of variables in a subformula that is not a sentence. We denote by $\mathsf{alt}(\varphi)$ the alternation number of a GRADED-SL formula φ. The *quantifier rank* of φ is the maximum nesting of quantifiers in φ, e.g., $\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g}(\alpha_1, x_1)\ldots(\alpha_n, x_n)\bigwedge_{i=1}^{n}(\langle\langle \mathsf{y} \rangle\rangle(\alpha_i, \mathsf{y})\psi_i) \to \psi_i$ has quantifier rank 2 if each ψ_i is quantifier free. Moreover, a *quantifier-block of* φ is a maximally-consecutive sequence of quantifiers in φ of the same type (*i.e.*, either all existential, or all universal). The *quantifier-block rank of* φ is exactly like the quantifier rank except that a quantifier block of j quantifiers contributes 1 instead of j to the count.

SL has a few natural syntactic fragments, the most powerful of which is called Nested-Goal SL. Similarly, we define GRADED-SL *Nested-Goal* (abbreviated GRADED-SL[NG]), as a syntactic fragment of GRADED-SL. As in Nested-Goal SL, in GRADED-SL[NG] we require that bindings and quantifications appear in exhaustive blocks. I.e., whenever there is a quantification over a variable in a formula ψ it is part of a consecutive sequence of quantifiers that covers all of the free variables that appear in ψ, and whenever an agent is bound to a strategy then it is part of a consecutive sequence of bindings of all agents to strategies. Finally, formulas with free agents are not allowed. To formalize GRADED-SL[NG] we first introduce some notions. A *quantification prefix* over a finite set $\mathrm{V} \subseteq \mathrm{Vr}$ of variables is a sequence $\wp \in \{\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g}, [[x_1, \ldots, x_n]]^{<g} : x_1, \ldots, x_n \in \mathrm{V} \wedge g \in \mathbb{N}\}^*$ such that each $x \in \mathrm{V}$ occurs exactly once in $\wp$. A *binding prefix* is a sequence $\flat \in \{(a, x) : \alpha \in \mathrm{Ag} \wedge x \in \mathrm{Vr}\}^{|\mathrm{Ag}|}$ such that each $\alpha \in \mathrm{Ag}$ occurs exactly once in $\flat$. We denote the set of binding prefixes by Bn, and the set of quantification prefixes over V by Qn(V).

DEFINITION 2.4. GRADED-SL[NG] *formulas are built inductively using the following grammar, with* $p \in \mathrm{AP}$, $\wp \in \mathrm{Qn}(\mathrm{V})$ *(*$\mathrm{V} \subseteq \mathrm{Vr}$*), and* $\flat \in \mathrm{Bn}$*:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \wp\varphi \mid \flat\varphi,$$

where in the rule $\wp\varphi$ *we require that* φ *is agent-closed and* $\wp \in \mathrm{Qn}(\mathsf{free}(\varphi))$*.*

We conclude this subsection by introducing GRADED-SL[1G], another graded extension of fragment of SL, namely the one-goal sub-logic. As the name says, this fragment is obtained by restricting GRADED-SL[NG] to encompass formulas with just one nested goal. The importance of this fragment in SL stems from the fact that it strictly includes ATL* while maintaining the same complexity for both the model checking and the satisfiability problems, *i.e.* 2EXPTIME-COMPLETE [42,43]. However, it is commonly believed that Nash Equilibrium cannot be expressed in this fragment. The definition of GRADED-SL[1G] follows.

DEFINITION 2.5. GRADED-SL[1G] *formulas are built inductively using the following grammar, with* $p \in \mathrm{AP}$, $\wp \in \mathrm{Qn}(\mathrm{V})$ *(*$\mathrm{V} \subseteq \mathrm{Vr}$*), and* $\flat \in \mathrm{Bn}$*:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \wp\flat\varphi,$$

with $\wp$ *quantification prefix over* $\mathsf{free}(\flat\varphi)$*.*

2.3 Semantics

As for SL, the interpretation of a GRADED-SL formula requires a valuation of its free placeholders. This is done via *assignments (from s)*, *i.e.*, functions $\chi \in \mathrm{Asg}(s) \triangleq (\mathrm{Vr} \cup \mathrm{Ag}) \to \mathrm{Str}(s)$ mapping variables/agents to strategies. We denote by $\chi[e \mapsto \sigma]$, with $e \in \mathrm{Vr} \cup \mathrm{Ag}$ and $\sigma \in \mathrm{Str}(s)$, the assignment that differs from χ only in the fact that e maps to σ. Extend this definition to tuples: for $\overline{e} = (e_1, \ldots, e_n)$ with $e_i \neq e_j$ for $i \neq j$, define $\chi[\overline{e} \mapsto \overline{\sigma}]$ to be the assignment that differs from χ only in the fact that e_i maps to σ_i (for each i).

Since assignments are total functions, each assignment from s determines a unique path from s, called a play:

DEFINITION 2.6. *For an assignment $\chi \in \mathrm{Asg}(s)$ the (χ, s)-play denotes the path $\pi \in \mathrm{Pth}(s)$ such that for all $i \in \mathbb{N}$, it holds that $\pi_{i+1} = \mathsf{tr}(\mathsf{dc})(\pi_i)$, where $\mathsf{dc}(\alpha) \triangleq \chi(\alpha)(\pi_{\leq i})$ for $\alpha \in \mathrm{Ag}$. The function $\mathsf{play} : \mathrm{Asg} \times \mathrm{St} \to \mathrm{Pth}$, with $\mathsf{dom}(\mathsf{play}) \triangleq \{(\chi, s) : \chi \in \mathrm{Asg}(s)\}$, maps (χ, s) to the (χ, s)-play $\mathsf{play}(\chi, s) \in \mathrm{Pth}(s)$.*

The notation $\pi_{\leq i}$ (resp. $\pi_{<i}$) denotes the prefix of the sequence π of length i (resp. $i-1$). Similarly, the notation π_i denotes the ith symbol of π. Thus, $\mathsf{play}(\chi, s)_i$ is the ith state on the play determined by χ from s.

The following definition of χ_i says how to interpret an assignment χ starting from a point i along the play, i.e., for each placeholder e, take the action the strategy $\chi(e)$ would do if it were given the prefix of the play up to i followed by the current history.

DEFINITION 2.7. *For $\chi \in \mathrm{Asg}(s)$ and $i \in \mathbb{N}$, writing $\rho \triangleq \mathsf{play}(\chi, s)_{\leq i}$ (the prefix of the play up to i) and $t \triangleq \mathsf{play}(\chi, s)_i$ (the last state of ρ) define $\chi_i \in \mathrm{Asg}(t)$ to be the assignment from t that maps $e \in \mathrm{Vr} \cup \mathrm{Ag}$ to the strategy that maps $h \in \mathrm{Hst}(t)$ to the action $\chi(e)(\rho_{<i} \cdot h)$.*

We now define the semantics of GRADED-SL. In particular, we define $\mathcal{G}, \chi, s \models \varphi$ and say that φ *holds at s in $\mathcal{G}$ under χ*.

DEFINITION 2.8. *Fix a CGS $\mathcal{G}$. For all states $s \in \mathrm{St}$ and assignments $\chi \in \mathrm{Asg}(s)$, the relation $\mathcal{G}, \chi, s \models \varphi$ is defined inductively on the structure of φ:*

- $\mathcal{G}, \chi, s \models p$ *iff* $p \in \mathsf{ap}(s)$;
- $\mathcal{G}, \chi, s \models \neg\varphi$ *iff* $\mathcal{G}, \chi, s \not\models \varphi$;
- $\mathcal{G}, \chi, s \models \varphi_1 \vee \varphi_2$ *iff* $\mathcal{G}, \chi, s \models \varphi_1$ *or* $\mathcal{G}, \chi, s \models \varphi_2$;
- $\mathcal{G}, \chi, s \models \mathsf{X}\,\varphi$ *iff* $\mathcal{G}, \chi_1, \mathsf{play}(\chi, s)_1 \models \varphi$;
- $\mathcal{G}, \chi, s \models \varphi_1 \,\mathsf{U}\, \varphi_2$ *iff there is an index $i \in \mathbb{N}$ such that $\mathcal{G}, \chi_i, \mathsf{play}(\chi, s)_i \models \varphi_2$ and, for all indexes $j \in \mathbb{N}$ with $j < i$, it holds that $\mathcal{G}, \chi_j, \mathsf{play}(\chi, s)_j \models \varphi_1$;*
- $\mathcal{G}, \chi, s \models (\alpha, x)\varphi$ *iff* $\mathcal{G}, \chi[\alpha \mapsto \chi(x)], s \models \varphi$;
- $\mathcal{G}, \chi, s \models \langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g} \varphi$ *iff there exist g many tuples $\overline{\sigma_1}, \ldots, \overline{\sigma_g}$ of strategies such that:*
  - $\overline{\sigma_i} \neq \overline{\sigma_j}$ *for* $i \neq j$, *and*
  - $\mathcal{G}, \chi[\overline{x} \mapsto \overline{\sigma_i}], s \models \varphi$ *for* $1 \leq i \leq g$.

Intuitively, the existential quantifier $\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g} \varphi$ allows us to count the number of distinct tuples of strategies that satisfy φ.

As usual, if χ and χ' agree on $\mathsf{free}(\varphi)$, then $\mathcal{G}, \chi, s \models \varphi$ if and only if $\mathcal{G}, \chi', s \models \varphi$, i.e., the truth of φ does not depend on the values the assignment takes on placeholders that are not free. Thus, for a sentence φ, we write $\mathcal{G} \models \varphi$ to mean that $\mathcal{G}, \chi, s_I \models \varphi$ for some (equivalently, for all) assignments χ, and where s_I is the initial state of $\mathcal{G}$.

Comparison with other logics. In the following we give the main intuitions relating GRADED-SL with SL and a *naive* fragment of GRADED-SL in which quantifiers are over single variables (and not tuple of variables).

GRADED-SL extends SL by replacing universal and existential strategy quantifiers $\langle\langle x \rangle\rangle$ and $[[x]]$ with their graded versions over tuples of variables $\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g}$ and $[[x_1, \ldots, x_n]]^{<g}$. Grades allow one to count, which is not possible, a priori, in SL. On the other hand, every formula of SL has an equivalent formula of GRADED-SL: formed by replacing every quantifier $\langle\langle x \rangle\rangle$ with $\langle\langle x \rangle\rangle^{\geq 1}$.

An important power of GRADED-SL is that it can quantify over tuples of strategy variables. Consider the formula $\varphi = \langle\langle x, y \rangle\rangle^{\geq 2}(a, x)(b, y)\psi$ in GRADED-SL that represents the property in which there exist two tuples of strategies that satisfy ψ, and compare it to the following attempt at expressing φ only quantifying over single strategies: $\varphi' \triangleq \langle\langle x \rangle\rangle^{\geq 2}\langle\langle y \rangle\rangle^{\geq 2}(a, x)(b, y)\psi$. Observe that φ' says that there are two different strategies σ_1 and σ_2 for agent α_1, and for each σ_i there are two different strategies δ_1^i and δ_2^i for the agent α_2 that satisfy ψ. Thus, there are four tuples of strategies that satisfy ψ, i.e., $\{\sigma_1, \delta_1^1\}$, $\{\sigma_1, \delta_2^1\}$, $\{\sigma_2, \delta_1^2\}$, $\{\sigma_2, \delta_2^2\}$. Other attempts (such as $\langle\langle x \rangle\rangle^{\geq 2}\langle\langle y \rangle\rangle^{\geq 1}(a, x)(b, y)\psi$), also fail to capture φ as they restrict one of the agents to a single strategy. This demonstrates the inadequacy of quantifying over single strategies for counting solution concepts.

2.4 Games with temporal objectives

In game theory, players have objectives that are summarised in a payoff function they receive depending on the resulting play. In order to specify such payoffs we follow a formalisation from [35] called *objective* LTL. This will allow us to model the Prisoner's Dilemma (PD), probably the most famous game in game-theory, as well as an iterated version (IPD). We then discuss appropriate solution concepts, and show how to express these in GRADED-SL.

Let $\mathcal{G}$ be a CGS with n agents. Let $m \in \mathbb{N}$ and fix, for each agent $\alpha_i \in \mathrm{Ag}$, an *objective* tuple $S_i \triangleq \langle \mathsf{f}_i, \varphi_i^1, \ldots, \varphi_i^m \rangle$, where $\mathsf{f}_i : \{0,1\}^m \to \mathbb{N}$, and each φ_i^j is an LTL formula over AP. If π is a play, then agent α_i receives payoff $\mathsf{f}_i(\overline{h}) \in \mathbb{N}$ where the j'th bit $\overline{h}_j$ of $\overline{h}$ is 1 if and only if $\pi \models \varphi_i^j$. For instance, f may count the number of formulas that are true.

Prisoner's Dilemma – One shot and Iterated

Two people have been arrested for robbing a bank and placed in separate isolation cells. Each has two possible choices, remaining silent or confessing. If a robber confesses and the other remains silent, the former is released and the latter stays in prison for a long time. If both confess they get two convictions, but they will get early parole. If both remain silent, they get a lighter sentence (e.g., on firearms possession charges). The dilemma faced by the prisoners is that, whatever the choice of the other prisoner, each is better off confessing than remaining silent. But the result obtained when both confess is worse than if they both remain silent.

We describe this (one-shot) scenario with the CGS in Figure 1 and, for agent α_i, the objective $S_i \triangleq \langle \mathsf{f}_i, \varphi_i^1, \varphi_i^2, \varphi_i^3, \varphi_i^4 \rangle$

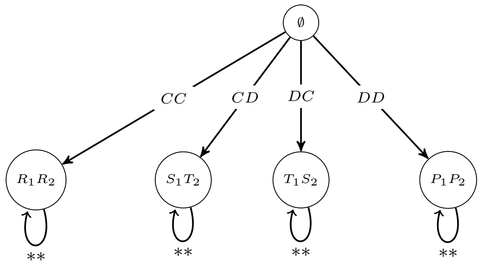

Figure 1: Prisoner's dilemma.

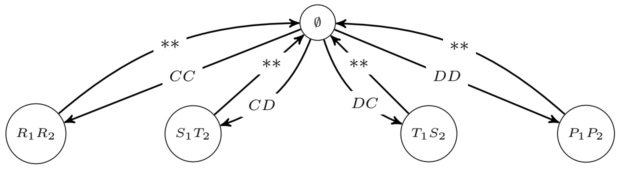

Figure 2: Iterated Prisoner's dilemma.

where $\varphi_i^1 \triangleq \mathsf{X}\,\mathbf{S}_i$, $\varphi_i^2 \triangleq \mathsf{X}\,\mathbf{P}_i$, $\varphi_i^3 \triangleq \mathsf{X}\,\mathbf{R}_i$, and $\varphi_i^4 \triangleq \mathsf{X}\,\mathbf{T}_i$ and f_i returns the value of its input vector interpreted as a binary number, *e.g.*, $\mathsf{f}_i(0100) = 4$.

In words, we have two agents α_1 and α_2. Each agent has two actions: cooperate (C) and defect (D), corresponding, respectively, to the options of remaining silent or confessing. For each possible pair of moves, the game goes in a state whose atomic propositions represent the payoffs: $\mathbf{R}_i$ represents the *reward* payoff that α_i receives if both cooperate; $\mathbf{P}_i$ is the *punishment* that α_i receives if both defect; $\mathbf{T}_i$ is the *temptation* that α_i receives as a sole defector, and $\mathbf{S}_i$ is the *sucker* payoff that α_i receives as a sole cooperator. The payoffs satisfy the following chain of inequalities: $\mathbf{T}_i > \mathbf{R}_i > \mathbf{P}_i > \mathbf{S}_i$.

The Iterated Prisoner's Dilemma (IPD) is used to model a series of interactions. This is like PD except that after the first choice, both agents have another choice and so on. This is modeled in Figure 2. The main difference between the one-shot and iterated PD is that in the latter the agents' actions (may) depend on the past behaviour. Convenient payoff functions for the IPD are the mean-average and discounted-payoff [11]. Such quantitative payoffs can be replaced, in a first approximation, by LTL payoffs such as "maximise the largest payoff I receive infinitely often". This is formalised, for agent α_i, by the objective $S_i \triangleq \langle \mathsf{f}_i, \varphi_i^1, \varphi_i^2, \varphi_i^3, \varphi_i^4 \rangle$ where $\varphi_i^1 \triangleq \mathsf{G}\,\mathsf{F}\,\mathbf{S}_i$, $\varphi_i^2 \triangleq \mathsf{G}\,\mathsf{F}\,\mathbf{P}_i$, $\varphi_i^3 \triangleq \mathsf{G}\,\mathsf{F}\,\mathbf{R}_i$, and $\varphi_i^4 \triangleq \mathsf{G}\,\mathsf{F}\,\mathbf{T}_i$, and f_i is as in the one-shot PD.

Solution Concepts. Solution concepts are criteria by which one captures what rational agents would do. This is especially relevant in case each agent has its own objective. The central solution concept in game theory is the Nash Equilibrium.

A tuple of strategies, one for each player, is called a *strategy profile*. A strategy profile is a *Nash equilibrium (NE)* if no agent can increase his payoff by unilaterally choosing a different strategy. A game may have zero, one, or many NE.

Consider first a CGS $\mathcal{G}$ with n agents, where the objective of the agent $\alpha_i \in \mathrm{Ag}$ contains a single LTL formula φ_i (with a larger payoff if it holds than if it doesn't). It is not hard to see that the following formula of SL expresses that $\overline{x} \triangleq (x_1 \ldots x_n)$ is a Nash Equilibrium:

$$\psi^1_{NE}(\overline{x}) \triangleq (\alpha_1, x_1) \ldots (\alpha_n, x_n) \bigwedge_{i=1}^{n} (\langle\!\langle \mathtt{y} \rangle\!\rangle (\alpha_i, \mathtt{y}) \varphi_i) \rightarrow \varphi_i$$

An alternative (which we will later use to obtain a formula in the fragment Graded-SL[ng] that expresses the existence of a unique NE) is the following:

$$\phi^1_{NE}(\overline{x}) \triangleq [\![\mathtt{y}_1]\!] \ldots [\![\mathtt{y}_n]\!] \bigwedge_{i=1}^{n} (\flat_i \varphi_i) \rightarrow \flat \varphi_i$$

where $\flat = (\alpha_1, x_1) \ldots (\alpha_n, x_n)$, and $\flat_i = (\alpha_1, x_1) \ldots (\alpha_{i-1}, \mathtt{x}_{i-1})(\alpha_i, \mathtt{y}_i)(\alpha_{i+1}, \mathtt{x}_{i+1}) \ldots (\alpha_n, x_n)$.

Consider now the general case, where each agent α_i has an objective tuple $S_i \triangleq \langle \mathsf{f}_i, \varphi_i^1, \ldots, \varphi_i^m \rangle$. Given a vector $\overline{h} \in \{0,1\}^m$, let $gd_i(\overline{h}) \triangleq \{\overline{t} \in \{0,1\}^m \mid \mathsf{f}_i(\overline{t}) \geq \mathsf{f}_i(\overline{h})\}$ be the set of vectors $\overline{t}$ for which the payoff for agent α_i is not worse than for $\overline{h}$. Also, let $\eta_i^{\overline{h}}$ be the formula obtained by taking a conjunction of the formulas $\varphi_i^1, \ldots, \varphi_i^m$ or their negations according to $\overline{h}$, i.e., by taking φ_a^j if the j'th bit in $\overline{h}$ is 1, and otherwise taking $\neg\varphi_a^j$. Formally, $\eta_i^{\overline{h}} \triangleq \wedge_{j \in \{1 \leq j \leq m \mid h_j = 1\}} \varphi_i^j \wedge \wedge_{j \in \{1 \leq j \leq m \mid h_j = 0\}} \neg\varphi_i^j$. Observe that the following formula says that $\overline{x} \triangleq (x_1 \ldots x_n)$ is a Nash Equilibrium:

$$\psi_{NE}(\overline{x}) \triangleq (\alpha_1, x_1) \ldots (\alpha_n, x_n) \bigwedge_{i=1}^{n} \bigwedge_{\overline{h} \in \{0,1\}^m} (\langle\!\langle \mathtt{y} \rangle\!\rangle (\alpha_i, \mathtt{y}) \eta_i^{\overline{h}}) \rightarrow \bigvee_{\overline{t} \in gd_i(\overline{h})} \eta_i^{\overline{t}}$$

Alike, one can modify $\phi^1_{NE}(\overline{x})$ to obtain a similar formula $\phi_{NE}(\overline{x})$ expressing that $\overline{x} \triangleq (x_1 \ldots x_n)$ is a Nash Equilibrium:

$$\phi_{NE}(\overline{x}) \triangleq [\![\mathtt{y}_1]\!] \ldots [\![\mathtt{y}_n]\!] \bigwedge_{i=1}^{n} \bigwedge_{\overline{h} \in \{0,1\}^m} (\flat_i \eta_i^{\overline{h}}) \rightarrow \bigvee_{\overline{t} \in gd_i(\overline{h})} \flat \eta_i^{\overline{t}}$$

Going back to the PD example, due to the simplicity of the payoff functions, the formula ψ_{NE} collapses to become:

$$\psi_{PD}(\overline{x}) \triangleq (\alpha_1, x_1) \ldots (\alpha_n, x_n) \bigwedge_{i=1}^{2} \bigwedge_{j=1}^{4} (\langle\!\langle y \rangle\!\rangle (\alpha_i, y) \varphi_i^j) \Rightarrow (\bigvee_{r \geq j} \varphi_i^r)$$

As it turns out (again due to the simplicity of the payoff functions), the formula above is also correct for the IPD.

It has been argued (in [35,55]) that NE may be implausible when used for sequential games (of which iterated one shot games are central examples), and that a more robust notion is subgame-perfect equilibrium [52]. Given a game $\mathcal{G}$, a strategy profile is a *subgame-perfect equilibrium (SPE)* if for every possible history of the game, the strategies are an NE. The following formula expresses that $\overline{x} \triangleq (\alpha_1, x_1) \ldots (\alpha_n, x_n)$ is an SPE:

$$\phi_{SPE}(\overline{x}) \triangleq [\![z_1, \ldots, z_n]\!] (\alpha_1, z_1) \ldots (\alpha_n, z_n)\, \mathsf{G}\, \phi_{NE}(\overline{x})$$

Using graded modalities, we can thus express the uniqueness of a NE using the following Graded-SL formula:

$$\langle\!\langle x_1, \ldots, x_n \rangle\!\rangle^{\geq 1} \psi_{NE}(\overline{x}) \wedge \neg \langle\!\langle x_1, \ldots, x_n \rangle\!\rangle^{\geq 2} \psi_{NE}(\overline{x})$$

By replacing ψ_{NE} with ϕ_{NE} (resp. by ϕ_{SPE}) in the formula above, we can express the uniqueness of a NE (resp. SPE) in Graded-SL[ng].

3. THE MODEL-CHECKING PROCEDURE

In this section we study the model-checking problem for GRADED-SL and show that it is decidable with a time-complexity that is non-elementary (i.e., not bounded by any fixed tower of exponentials). However, it is elementary if the number of blocks of quantifiers is fixed. For the algorithmic procedures, we follow an *automata-theoretic approach* [37], reducing the decision problem for the logic to the emptiness problem of an automaton. The procedure we propose here extends that used for SL in [43]. The only case that is different is the new graded quantifier over tuples of strategies.

We start with the central notions of automata theory, and then show how to convert a GRADED-SL sentence φ into an automaton that accepts exactly the (tree encodings) of the concurrent game structures that satisfy φ. This is used to prove the main result about GRADED-SL model checking.

3.1 Automata Theory

A Σ-*labeled* Υ-*tree* T is a pair $\langle T, V\rangle$ where $T \subseteq \Upsilon^+$ is prefix-closed (i.e., if $t \in T$ and $s \in \Upsilon^+$ is a prefix of t then also $s \in T$), and $V : T \to \Sigma$ is a labeling function. Note that every word $w \in \Upsilon^+ \cup \Upsilon^\omega$ with the property that every prefix of w is in T, can be thought of as a path in T. Infinite paths are called *branches.*

Nondeterministic tree automata (NTA) are a generalization to infinite trees of the classical automata on words [54]. *Alternating tree automata* (ATA) are a further generalization of nondeterministic tree automata [23]. Intuitively, on visiting a node of the input tree, while an NTA sends exactly one copy of itself to each of the successors of the node, an ATA can send several copies to the same successor. We use the parity acceptance condition [37].

For a set X, let $\mathtt{B}^+(X)$ be the set of positive Boolean formulas over X, including the constants **true** and **false**. A set $Y \subseteq X$ satisfies a formula $\theta \in \mathtt{B}^+(X)$, written $Y \models \theta$, if assigning **true** to elements in Y and **false** to elements in $X \setminus Y$ makes θ true.

DEFINITION 3.1. *An Alternating Parity Tree-Automaton (*APT*) is a tuple* $\mathcal{A} \triangleq \langle \Sigma, \Delta, \mathrm{Q}, \delta, q_0, \aleph\rangle$*, where* Σ *is the* input alphabet*,* Δ *is a set of* directions*,* Q *is a finite set of* states*,* $q_0 \in \mathrm{Q}$ *is an* initial state*,* $\delta : \mathrm{Q} \times \Sigma \to \mathtt{B}^+(\Delta \times \mathrm{Q})$ *is an* alternating transition function*, and* $\aleph$*, an* acceptance condition*, is of the form* $(\mathrm{F}_1, \ldots, \mathrm{F}_k) \in (2^{\mathrm{Q}})^+$ *where* $\mathrm{F}_1 \subseteq \mathrm{F}_2 \ldots \subseteq \mathrm{F}_k = \mathrm{Q}$*.*

The set $\Delta \times \mathrm{Q}$ is called the set of *moves*. An NTA is an ATA in which each conjunction in the transition function δ has exactly one move (d, q) associated with each direction d.

An *input tree* for an APT is a Σ-labeled Δ-tree $\mathsf{T} = \langle \mathrm{T}, \mathsf{v}\rangle$. A *run* of an APT on an input tree $\mathsf{T} = \langle \mathrm{T}, \mathsf{v}\rangle$ is a $(\Delta \times \mathrm{Q})$-tree R such that, for all nodes $x \in \mathrm{R}$, where $x = (d_1, q_1) \ldots (d_n, q_n)$ (for some $n \in \mathbb{N}$), it holds that *(i)* $y \triangleq (d_1, \ldots, d_n) \in \mathrm{T}$ and *(ii)* there is a set of moves $\mathrm{S} \subseteq \Delta \times \mathrm{Q}$ with $\mathrm{S} \models \delta(q_n, \mathsf{v}(y))$ such that $x \cdot (d, q) \in \mathrm{R}$ for all $(d, q) \in \mathrm{S}$.

The acceptance condition allows us to say when a run is successful. Let R be a run of an APT $\mathcal{A}$ on an input tree T and $u \in (\Delta \times \mathrm{Q})^\omega$ one of its branches. Let $\mathsf{inf}(u) \subseteq \mathrm{Q}$ denote the set of states that occur in infinitely many moves of u. Say that u *satisfies the parity acceptance condition* $\aleph = (\mathrm{F}_1, \ldots, \mathrm{F}_k)$ if the least index $i \in [1, k]$ for which $\mathsf{inf}(u) \cap \mathrm{F}_i \neq \emptyset$ is even. An APT *accepts* an input tree T iff there exists a run R of $\mathcal{A}$ on T such that all its branches satisfy the acceptance condition $\aleph$. The *language* $\mathrm{L}(\mathcal{A})$ of the APT $\mathcal{A}$ is the set of trees T accepted by $\mathcal{A}$. Two automata are *equivalent* if they have the same language. The *emptiness problem* for alternating parity tree-automata is to decide, given $\mathcal{A}$, whether $\mathrm{L}(\mathcal{A}) = \emptyset$. The *universality problem* is to decide whether $\mathcal{A}$ accepts all trees.

3.2 From Logic to Automata

Following an automata-theoretic approach, we reduce the model-checking problem of GRADED-SL to the emptiness problem for alternating parity tree automata [43]. The main step is to translate every GRADED-SL formula φ (i.e., φ may have free placeholders), concurrent-game structure $\mathcal{G}$, and state s, into an APT that accepts a tree if and only if the tree encodes an assignment χ such that $\mathcal{G}, \chi, s \models \varphi$.

We first describe the encoding, following [43]. Informally, the CGS $\mathcal{G}$ is encoded by its "tree-unwinding starting from s" whose nodes represent histories, i.e., the St-labeled St-tree $\mathsf{T} \triangleq \langle \mathrm{Hst}(s), \mathsf{u}\rangle$ such that $\mathsf{u}(h)$ is the last symbol of h. Then, every strategy $\chi(e)$ with $e \in \mathsf{free}(\varphi)$ is encoded as an Ac-labelled tree over the unwinding. The unwinding and these strategies $\chi(e)$ are viewed as a single $(\mathrm{VAL} \times \mathrm{St})$-labeled tree where $\mathrm{VAL} \triangleq \mathsf{free}(\varphi) \to \mathrm{Ac}$.

DEFINITION 3.2. *The* encoding of χ (w.r.t. $\varphi, \mathcal{G}, s$) *is the* $(\mathrm{VAL} \times \mathrm{St})$*-labeled* St*-tree* $\mathsf{T} \triangleq \langle \mathrm{T}, \mathsf{u}\rangle$ *such that* T *is the set of histories* h *of* $\mathcal{G}$ *starting with* s *and* $\mathsf{u}(h) \triangleq (\mathsf{f}, q)$ *where* q *is the last symbol in* h *and* $\mathsf{f} : \mathsf{free}(\varphi) \to \mathrm{Ac}$ *is defined by* $\mathsf{f}(e) \triangleq \chi(e)(h)$ *for all* $e \in \mathsf{free}(\varphi)$*.*[2]

LEMMA 3.1. *For every* GRADED-SL *formula* φ*,* CGS $\mathcal{G}$*, and state* $s \in \mathrm{St}$*, there exists an* APT $\mathcal{A}_\varphi$ *such that for all assignments* χ*, if* T *is the encoding of* χ *(w.r.t.* $\varphi, \mathcal{G}, s$*), then* $\mathcal{G}, \chi, s \models \varphi$ *iff* $\mathsf{T} \in \mathrm{L}(\mathcal{A}_\varphi)$*.*

PROOF. As in [43] we induct on the structure of the formula φ to construct the corresponding automaton $\mathcal{A}_\varphi$. The Boolean operations are easily dealt with using the fact that disjunction corresponds to non-determinism, and negation corresponds to dualising the automaton. Note (†) that thus also conjunction is dealt with due to De Morgan's laws. The temporal operators are dealt with by following the unique play (determined by the given assignment) and verifying the required subformulas, e.g., for $\mathsf{X}\,\psi$ the automaton, after taking one step along the play, launches a copy of the automaton for ψ. All of these operations incur a linear blowup in the size of the automaton. The only case that differs from [43] is the quantification, i.e., we need to handle the case that $\varphi = \langle\!\langle x_1, \ldots, x_n\rangle\!\rangle^{\geq g}\psi$. Recall that $\mathcal{G}, \chi, s \models \langle\!\langle x_1, \ldots, x_n\rangle\!\rangle^{\geq g}\psi$ iff there exists g many tuples $\overline{\sigma_1}, \ldots, \overline{\sigma_g}$ of strategies such that: $\overline{\sigma_a} \neq \overline{\sigma_b}$ for $a \neq b$, and $\mathcal{G}, \chi[\overline{x} \mapsto \overline{\sigma_i}], s \models \psi$ for $1 \leq i \leq g$. We show how to build an NPT for φ that mimics this definition: it will be a projection of an APT $\mathcal{D}_\psi$, which itself is the intersection of two automata, one checking that each of the g tuples of strategies satisfies ψ, and the other checking that each pair of the g tuples of strategies is distinct.

In more detail, introduce a set of fresh variables $X \triangleq \{x_i^j \in \mathrm{Vr} : i \leq n, j \leq g\}$, and consider the formulas ψ^j (for $j \leq g$) formed from ψ by renaming x_i (for $i \leq n$) to x_i^j. Define $\psi' \triangleq \wedge_{j \leq g}\psi^j$. Note that, by induction, each ψ^j has a corresponding APT, and thus, using the conjunction-case

[2]In case $\mathsf{free}(\varphi) = \emptyset$, then f is the (unique) empty function. In this case, the encoding of every χ may be viewed as the tree-unwinding from s.

(†) above, there is an APT $\mathcal{B}$ for ψ'. Note that the input alphabet for $\mathcal{B}$ is $(\mathsf{free}(\psi') \to \mathrm{Ac}) \times \mathrm{St}$ and that $X \subseteq \mathsf{free}(\psi')$.

On the other hand, let $\mathcal{C}$ be an APT with input alphabet $(\mathsf{free}(\psi') \to \mathrm{Ac}) \times \mathrm{St}$ that accepts a tree $\mathsf{T} = \langle \mathrm{T}, \mathsf{v} \rangle$ if and only if for every $a \neq b \leq g$ there exists $i \leq n$ and $h \in \mathrm{T}$ such that $\mathsf{v}(h) = (f, q)$ and $f(x_i^a) \neq f(x_i^b)$.

Form the APT $\mathcal{D}_\psi$ for the intersection of $\mathcal{B}$ and $\mathcal{C}$.

Now, using the classic transformation [46], we remove alternation from the APT $\mathcal{D}_\psi$ to get an equivalent NPT $\mathcal{N}$ (note that this step costs an exponential). Finally, use the fact that NPTs are closed under projection (with no blowup) to get an NPT for the language $proj_X(L(\mathcal{N}))$ of trees that encode assignments χ satisfying φ.

For completeness we recall this last step. If L is a language of Σ-labeled trees with $\Sigma \triangleq \mathrm{A} \to \mathrm{B}$, and $X \subset \mathrm{A}$, then the X-*projection of* L, written $proj_{\mathrm{X}}(L)$, is the language of Σ'-labeled trees with $\Sigma' \triangleq \mathrm{A} \setminus \mathrm{X} \to \mathrm{B}$ such that $\mathsf{T} \triangleq \langle \mathrm{T}, \mathsf{v} \rangle \in proj_{\mathrm{X}}(L)$ if and only if there exists an X-labeled tree $\langle \mathrm{T}, \mathsf{w} \rangle$ such that the language L contains the tree $\langle \mathrm{T}, \mathsf{u} \rangle$ where $\mathsf{u} : \mathrm{T} \to (\mathrm{A} \to \mathrm{B})$ maps $t \in \mathrm{T}$ to $\mathsf{v}(t) \cup \mathsf{w}(t)$. Now, if $\mathcal{N}$ is an NPT with input alphabet $\Sigma \triangleq \mathrm{A} \to \mathrm{B}$, and if $\mathrm{X} \subset \mathrm{A}$, then there is an NPT with input alphabet $\Sigma' \triangleq \mathrm{A} \setminus \mathrm{X} \to \mathrm{B}$ with language $proj_{\mathrm{X}}(L(\mathcal{N}))$.

The proof that the construction is correct is immediate. □

We make some remarks about the proof. First, all the cases in the induction incur a linear blowup except for the quantification case (recall that the translation from an APT to an NPT results in an exponentially larger automaton [37]). Thus, the size of the APT for φ is non-elementary in the quantifier-rank of φ. However, we can say a little more. Note that a block of k identical quantifiers only costs, in the worst case, a single exponential (and not k many exponentials) because we can extend the proof above to deal with a block of quantifiers at once. Thus, we get that the size of the APT for φ is non-elementary in the quantifier-block rank of φ.

Here is the main decidability result.

THEOREM 3.1. *The model-checking problem for* GRADED-SL *is* PTIME-COMPLETE *w.r.t. the size of the model and* $(k+1)$EXPTIME *if* $k \geq 1$ *is the quantifier-block rank of* φ. *Moreover, if* φ *is the form* $\wp\psi$, *where* $\wp$ *is a quantifier-block, and* ψ *is of quantifier-block rank* $k-1$, *then the complexity is* kEXPTIME.

PROOF. The lower-bound w.r.t the size of the model already holds for SL [43]. For the upper bound, use Lemma 3.1 to transform the CGS and φ into an APT and test its emptiness. The complexity of checking emptiness (or indeed, universality) of an APT is in EXPTIME [37]. As discussed after the proof of the Lemma, the size of the APT is a tower of exponentials whose height is the quantifier-block rank of φ. This gives the $(k+1)$EXPTIME upper bound.

Moreover, suppose that $\varphi = \wp\psi$ where $\wp$ consists of, say, n existential quantifiers (resp. universal quantifiers). The quantifier-block rank of ψ is $k-1$. Moreover, in the proof of Lemma 3.1, the APT $\mathcal{D}_\psi$, whose size is non-elementary in $k-1$, has the property that it is non-empty (resp. universal) if and only if the CGS satisfies $\wp\psi$. Conclude that model checking $\wp\psi$ can be done in kEXPTIME. □

THEOREM 3.2. *The model-checking problem for* GRADED-SL[NG] *is* PTIME-COMPLETE *w.r.t. the size of the model and* $(k+1)$-EXPTIME *when restricted to formulas of maximum alternation number* k.

PROOF. The lower bound already holds for SL[NG] [43], and the upper bound is obtained by following the same reasoning for SL[NG] of the singleton existential quantifier [43] but using the automaton construction as in Theorem 3.1. □

Directly from the statements reported above, we get the following results:

THEOREM 3.3. *Checking the uniqueness of NE, and checking the uniqueness of SPE, can be done in* 2EXPTIME.

PROOF. For NE: by Section 2.4, we need to check that $\langle\!\langle x_1, \ldots, x_n \rangle\!\rangle^{\geq 1} \psi_{NE}(\overline{x})$ holds but $\langle\!\langle x_1, \ldots, x_n \rangle\!\rangle^{\geq 2} \psi_{NE}(\overline{x})$ does not; by the second part of Theorem 3.1, each of these two model-checking problems can be decided in 2EXPTIME.

For SPE: apply Theorem 3.2 and use the fact that the formula for SPE in Section 2.4 is in GRADED-SL Nested-Goal and has alternation number 1. □

We conclude this section with the complexity of the model checking problem for GRADED-SL[1G]. Also in this case one can derive the lower bound from the one holding for the corresponding sub-logic in SL (SL[1G]) and the upper bound by using the same algorithm for SL[1G] but plugging a (yet no more complex) different automata construction for the new existential quantifier modality. Indeed the model checking problem for GRADED-SL[1G] is 2EXPTIME-COMPLETE. It is worth recalling that SL[1G] strictly subsumes ATL* [43]. It is quite immediate to see that this also holds in the graded setting (note that ATL* already allows quantifying over tuples of agents' (bound) strategies). As the model checking for ATL* is already 2EXPTIME-HARD, we get that also for the graded extension for this logic, which we name GATL*, the model checking problem is 2EXPTIME-COMPLETE. The model checking results for both GATL* and GRADED-SL[1G] are reported in the following theorem.

THEOREM 3.4. *The model-checking problem for* GATL* *and* GRADED-SL[1G] *is* PTIME-COMPLETE w.r.t. *the size of the model and* 2-EXPTIME-COMPLETE *in the size of formula.*

4. CONCLUSION

The Nash equilibrium is the foundational solution concept in game theory. The last twenty years have witnessed the introduction of many logical formalisms for modeling and reasoning about solution concepts, and NE in particular [9,14, 19,32,39,43,44]. These formalisms are useful for addressing qualitative questions such as *"does the game admit a Nash equilibrium?"*. Among others, Strategy Logic (SL) has come to the fore as a general formalism that can express and solve this question, for LTL objectives, in 2EXPTIME. Contrast this with the fact that this question is 2EXPTIME-complete even for two player zero-sum LTL games [4].

One of the most important questions about NE in computational game theory is *"does the game admit more than one NE?"* [20, 48] — the unique NE problem. This problem is deeply investigated in game theory and is shown to be very challenging [2, 21, 28, 29, 47, 49, 53, 57]. Prior to this work, no logic-based technique, as far as we know, solved this problem.[3] In this paper we introduced GRADED-SL to address

[3]In the related work section we discussed the logic GSL that, although motivated by the need to address the unique NE problem, only supplies a model-checking algorithm for a very small fragment of GSL that, it is assumed, is not able to express the existence of NE.

and solve the unique NE problem. We have demonstrated that GRADED-SL is elegant, simple, and very powerful, and can solve the unique NE problem for LTL objectives in 2EXPTIME, and thus at the same complexity that is required to merely decide if a NE exists. We also instantiate our formalism by considering the well-known prisoner's dilemma and its iterated version. We have also shown that using the same approach one can express (and solve) the uniqueness of other standard solution concepts, e.g., subgame-perfect equilibria, again in 2EXPTIME. Finally, our work gives the first algorithmic solution to the model-checking problem of a graded variant of ATL*, and proves it to be 2EXPTIME-COMPLETE.

The positive results presented in this paper open several directions for future work. We are most excited about extending LTL objectives to quantitative objectives such as mean-payoff or discounted-payoff. These naturally extend classic games with quantitative aspects. That is, the result of a play is a real-valued payoff for each player [58]. In a mean-payoff game, one is interested in the *long-run average* of the edge-weights along a play, called the *value* of the play. In the basic setting, there are two players, one wishing to minimize this value, and the other to maximize it. In the discounted version, the weights associated with edges are "*discounted*" with time. In other words, an edge chosen at time t adds a weight to the long-run average that is greater than the value the same edge would contribute if chosen later on. Because of their applicability to economics these games have been studied from an algorithmic perspective for some time [58]. Also, the connection of mean-payoff and discounted-payoff objectives with NE has been recently investigated in the multi-agent setting (see [13] for a recent work). However, extending our results to the weighted setting may prove challenging since, in this setting, the automata-theoretic approach gives rise to weighted-automata, for which many problems are much harder or undecidable (though not in all cases) [1, 5].

In the multi-agent setting, reasoning about epistemic alternatives plays a key role. Thus, an important extension would be to combine the knowledge operators in SLK [18] with the graded quantifiers we introduced for GRADED-SL. Since strategic reasoning under imperfect information has an undecidable model-checking problem [22], one may restrict to memoryless strategies as was done for SLK. More involved, would be to add grades to the knowledge operators, thus being able to express "there exists at least g equivalent worlds" [56].

Last but not least, another direction is to consider implementing GRADED-SL and its model-checking procedure in a formal verification tool. A reasonable approach would be, for example, to extend the tool SLK-MCMAS [18].

Acknowledgments

We thank Michael Wooldridge for suggesting uniqueness of Nash Equilibria as an application of graded strategy logic. Benjamin Aminof is supported by the Austrian National Research Network S11403-N23 (RiSE) of the Austrian Science Fund (FWF) and by the Vienna Science and Technology Fund (WWTF) through grant ICT12-059. Sasha Rubin is a Marie Curie fellow of the Istituto Nazionale di Alta Matematica. Aniello Murano is partially supported by the GNCS 2016 project: Logica, Automi e Giochi per Sistemi Auto-adattivi.

REFERENCES

[1] S. Almagor, U. Boker, and O. Kupferman. What's decidable about weighted automata? In *ATVA'11*, LNCS 6996, pages 482–491, 2011.

[2] E. Altman, H. Kameda, and Y. Hosokawa. Nash equilibria in load balancing in distributed computer systems. *IGTR*, 4(2):91–100, 2002.

[3] R. Alur, T. Henzinger, and O. Kupferman. Alternating-Time Temporal Logic. *JACM*, 49(5):672–713, 2002.

[4] R. Alur, S. La Torre, and P. Madhusudan. Playing games with boxes and diamonds. In *CONCUR'03*, LNCS 2761, pages 127–141. Springer, 2003.

[5] B. Aminof, O. Kupferman, and R. Lampert. Rigorous approximated determinization of weighted automata. *Theor. Comput. Sci.*, 480:104–117, 2013.

[6] B. Aminof, A. Legay, A. Murano, and O. Serre. μ-calculus pushdown module checking with imperfect state information. In *IFIP-TCS'08*, IFIP 273, pages 333–348. Springer, 2008.

[7] B. Aminof, A. Murano, and S. Rubin. On CTL* with graded path modalities. In *LPAR-20'15*, LNCS 9450, pages 281–296, 2015.

[8] R. Axelrod. The evolution of strategies in the iterated prisoners dilemma. *The dynamics of norms*, pages 1–16, 1987.

[9] F. Belardinelli. A logic of knowledge and strategies with imperfect information. In *LAMAS'15*, 2015.

[10] A. Bianco, F. Mogavero, and A. Murano. Graded Computation Tree Logic. *TOCL*, 13(3):25:1–53, 2012.

[11] K. G. Binmore. *Fun and Games: A Text on Game Theory*. D.C. Heath, 1992.

[12] P. Bonatti, C. Lutz, A. Murano, and M. Vardi. The Complexity of Enriched muCalculi. *LMCS*, 4(3):1–27, 2008.

[13] E. Boros, K. M. Elbassioni, V. Gurvich, and K. Makino. Nested family of cyclic games with k-total effective rewards. *CoRR*, abs/1412.6072, 2014.

[14] T. Brihaye, A. D. C. Lopes, F. Laroussinie, and N. Markey. ATL with Strategy Contexts and Bounded Memory. In *LFCS'09*, LNCS 5407, pages 92–106, 2009.

[15] D. Calvanese, G. De Giacomo, and M. Lenzerini. Reasoning in expressive description logics with fixpoints based on automata on infinite trees. In *IJCAI'99*, pages 84–89, 1999.

[16] V. Capraro. A model of human cooperation in social dilemmas. *CoRR*, abs/1307.4228, 2013.

[17] V. Capraro, M. Venanzi, M. Polukarov, and N. R. Jennings. Cooperative equilibria in iterated social dilemmas. In *SAGT'13*, LNCS 8146, pages 146–158. 2013.

[18] P. Čermák, A. Lomuscio, F. Mogavero, and A. Murano. MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications. In *CAV'14*, LNCS 8559, pages 524–531. Springer, 2014.

[19] K. Chatterjee, T. Henzinger, and N. Piterman. Strategy Logic. *IC*, 208(6):677–693, 2010.

[20] R. Cornes, R. Hartley, and T. Sandler. An elementary proof via contraction. *Journal of Public Economic Theory*, 1(4):499–509, 1999.

[21] J. B. D. Simchi-Levi, X. Chen. *The Logic of Logistics:*

Theory, Algorithms, and Applications for Logistics Management. Science and Business Media. 2013.

[22] C. Dima and F. Tiplea. Model-checking ATL under Imperfect Information and Perfect Recall Semantics is Undecidable. Technical report, arXiv, 2011.

[23] E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *ASFCS'91*, pages 368–377, 1991.

[24] M. Faella, M. Napoli, and M. Parente. Graded Alternating-Time Temporal Logic. *FI*, 105(1-2):189–210, 2010.

[25] A. Ferrante, A. Murano, and M. Parente. Enriched Mu-Calculi Module Checking. *LMCS*, 4(3):1–21, 2008.

[26] K. Fine. In So Many Possible Worlds. *NDJFL*, 13:516–520, 1972.

[27] D. Fisman, O. Kupferman, and Y. Lustig. Rational Synthesis. In *TACAS'10*, LNCS 6015, pages 190–204. Springer, 2010.

[28] C. D. Fraser. The uniqueness of nash equilibrium in the private provision of public goods: an alternative proof. *Journal of Public Economics*, 49(3):389–390, 1992.

[29] A. Glazer and K. A. Konrad. Private provision of public goods, limited tax deducibility, and crowding out. *FinanzArchiv / Public Finance Analysis*, 50(2):203–216, 1993.

[30] E. Grädel. On the restraining power of guards. *J. Symb. Log.*, 64(4):1719–1742, 1999.

[31] E. Grädel, M. Otto, and E. Rosen. Two-Variable Logic with Counting is Decidable. In *LICS'97*, pages 306–317. IEEE Computer Society, 1997.

[32] J. Gutierrez, P. Harrenstein, and M. Wooldridge. Reasoning about equilibria in game-like concurrent systems. In *KR'14*, 2014.

[33] J. Gutierrez, P. Harrenstein, and M. Wooldridge. Expressiveness and complexity results for strategic reasoning. In *CONCUR'15*, LIPIcs 42, pages 268–282, 2015.

[34] B. Hollunder and F. Baader. Qualifying Number Restrictions in Concept Languages. In *KR'91*, pages 335–346. Kaufmann, 1991.

[35] O. Kupferman, G. Perelli, and M. Y. Vardi. Synthesis with rational environments. In *EUMAS'14*, LNCS 8953, pages 219–235, 2014.

[36] O. Kupferman, U. Sattler, and M. Vardi. The Complexity of the Graded muCalculus. In *CADE'02*, LNCS 2392, pages 423–437. Springer, 2002.

[37] O. Kupferman, M. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *JACM*, 47(2):312–360, 2000.

[38] K. Leyton-Brown and Y. Shoham. *Essentials of Game Theory: A Concise, Multidisciplinary Introduction (Synthesis Lectures on Artificial Intelligence and Machine Learning)*. M&C, 2008.

[39] A. Lopes, F. Laroussinie, and N. Markey. ATL with Strategy Contexts: Expressiveness and Model Checking. In *FSTTCS'10*, LIPIcs 8, pages 120–132, 2010.

[40] V. Malvone, F. Mogavero, A. Murano, and L. Sorrentino. On the counting of strategies. In *TIME'15*, pages 170–179, 2015.

[41] J. H. Miller. The coevolution of automata in the repeated prisoner's dilemma. *Journal of Economic Behavior & Organization*, 29(1):87–112, 1996.

[42] F. Mogavero, A. Murano, G. Perelli, and M. Vardi. What Makes ATL* Decidable? A Decidable Fragment of Strategy Logic. In *CONCUR'12*, LNCS 7454, pages 193–208. Springer, 2012.

[43] F. Mogavero, A. Murano, G. Perelli, and M. Vardi. Reasoning About Strategies: On the Model-Checking Problem. *TOCL*, 15(4):34:1–42, 2014.

[44] F. Mogavero, A. Murano, and M. Vardi. Reasoning About Strategies. In *FSTTCS'10*, LIPIcs 8, pages 133–144. Leibniz-Zentrum fuer Informatik, 2010.

[45] F. Mogavero, A. Murano, and M. Vardi. Relentful Strategic Reasoning in Alternating-Time Temporal Logic. In *LPAR'10*, LNAI 6355, pages 371–387, 2010.

[46] D. E. Muller and P. E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of rabin, mcnaughton and safra. *Theor. Comput. Sci.*, 141(1&2):69–107, 1995.

[47] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multiuser communication networks. *IEEE/ACM Trans. Netw.*, 1(5):510–521, 1993.

[48] G. Papavassilopoulos and J. B. Cruz. On the uniqueness of nash strategies for a class of analytic differential games. *Journal of Optimization Theory and Applications*, 27(2):309–314, 1979.

[49] L. Pavel. *Game Theory for Control of Optical Networks*. Science and Business Media. Springer, 2012.

[50] A. Pnueli and R. Rosner. Distributed reactive systems are hard to synthesize. In *FOCS'90*, pages 746–757, 1990.

[51] A. Rubinstein. Finite automata play the repeated prisoner's dilemma. *Journal of Economic Theory*, 39(1):83–96, 1986.

[52] R. Selten. Spieltheoretische behandlung eines oligopolmodells mit nachfragetragheit. *Zeitschrift fur die gesamte Staatswissenschaft*, 121:301–324, 1965.

[53] H. R. V. T. C. Bergstrom, L. E. Blume. On the private provision of public goods. *Journal of Public Economics*, 29(1):25–49, 1986.

[54] W. Thomas. Infinite trees and automaton definable relations over omega-words. In *STACS'90*, pages 263–277, 1990.

[55] M. Ummels. Rational behaviour and strategy construction in infinite multiplayer games. In *FSTTCS'06*, LNCS 4337, pages 212–223, 2006.

[56] W. van der Hoek and J.-J. Meyer. Graded modalities in epistemic logic. In *LFCS'92*, LNCS 620, pages 503–514. 1992.

[57] Y. Zhang and M. Guizani. *Game Theory for Wireless Communications and Networking*. CRC Press, 2011.

[58] U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158(1&2):343–359, 1996.