Graded Modalities in Strategy Logic

Benjamin Aminof[a], Vadim Malvone[b], Aniello Murano[b], Sasha Rubin[b]

[a] *Technische Universitat Wien, Austria*
[b] *Università degli Studi di Napoli Federico II, Italy*

Abstract

Strategy Logic (SL) is a logical formalism for strategic reasoning in multi-agent systems. Its main feature is that it has variables for strategies that are associated to specific agents using a binding operator. In this paper we introduce Graded Strategy Logic (GradedSL), an extension of SL by graded quantifiers over tuples of strategy variables, i.e., "there exist at least g different tuples $(x_1, ..., x_n)$ of strategies" where g is a cardinal from the set $\mathbb{N} \cup \{\aleph_0, \aleph_1, 2^{\aleph_0}\}$. We prove that the model-checking problem of GradedSL is decidable. We then turn to the complexity of fragments of GradedSL. When the g's are restricted to finite cardinals, written Graded$_{\mathbb{N}}$SL, the complexity of model-checking is no harder than for SL, i.e., it is non-elementary in the quantifier-block rank. We illustrate our formalism by showing how to count the number of different strategy profiles that are Nash equilibria (NE). By analysing the structure of the specific formulas involved, we conclude that the important problem of checking for the existence of a unique NE can be solved in 2ExpTime, which is not harder than merely checking for the existence of such an equilibrium.

Keywords: Strategic logics; Graded modalities; Nash equilibria.

1. Introduction

Strategy Logic (SL) is a powerful formalism for reasoning about strategies in multi-agent systems [1, 2]. Strategies tell an agent what to do — they are functions that prescribe an action based on the history. The key 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 [4, 2, 5, 6, 7] among which that a strategy profile $\overline{x}$ is a Nash equilibrium, and thus also the existence of a Nash equilibrium (NE).

The 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 [8, 9, 10]. This problem is relevant to the predictive power of NE since, in case there are multiple equilibria, the outcome of the game cannot be uniquely pinned down [11, 12, 13]. Unfortunately, uniqueness has mainly been established either for special cost functions [8], or for very restrictive game topologies [14]. Moreover, there is no general theory of when games have unique equilibria that can be applied to different application areas [8].

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, by introducing an extension of SL called GradedSL. 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 $[[x_1, \ldots, x_n]]^{<g}$, where $g \in \mathbb{N} \cup \{\aleph_0, \aleph_1, 2^{\aleph_0}\}$. Here, two tuples are different if they are different in some component, and two strategies are different if they disagree on some history. The key is being able to express uniqueness of NE is the combination of quantifying over tuples (instead of singleton variables), and adding counting (in the form of graded modalities).

As far as the expressive power of GradedSL concerns, we prove that counting strategies in SL is not possible in general (see Theorem 2.1). On the other hand, every formula of SL has an equivalent formula of GradedSL formed by replacing every quantifier $\langle\langle x\rangle\rangle$ with $\langle\langle x\rangle\rangle^{\geq 1}$. Additionally, the possibility of quantifying over tuples of strategy variables (rather than single strategies) makes the logic quite expressive.

We address the model-checking problem for GradedSL and prove that it is decidable. We also address the complexity of several fragments of GradedSL. First we consider the case in which the g's are restricted to finite cardinals, written $\text{Graded}_{\mathbb{N}}\text{SL}$. Then we investigate the graded extension of classic fragments of SL, such as Nested-Goal SL and one-goal SL [2], while maintaining the restriction of grades over finite cardinals. Roughly speaking, 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; further, the one-goal restriction is obtained by forbidding any nesting and Boolean operation over bindings (see Section 2.4 for details).

We show that the complexity of the model-checking problem for $\text{Graded}_{\mathbb{N}}\text{SL}$ is no harder than for SL, i.e., it is non-elementary in the nesting depth of quantifiers. In particular, we show that model checking $\text{Graded}_{\mathbb{N}}\text{SL}$ 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 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[4, 2, 5, 6, 7]. Since the existence of a NE, and the fact that there is at most one NE, can be expressed in Graded$_{\mathbb{N}}$SL using simple formulas (assuming that the agents' goals are given as LTL formulas) we are able to conclude that the problem of checking the uniqueness of a NE can be solved in 2ExpTime. Previously, it was known that existence of NE can be checked in 2ExpTime [2, 5]. Thus, GradedSL is the first logic that can solve the existence and uniqueness of NE (as well as many other solution concepts) in 2ExpTime.

(and is 2ExpTime-**complete).**

Concerning the graded Nested-Goal fragment, namely GradedSL[ng], we show that, in case the g's are restricted to finite cardinals, it has the same model-checking complexity as Nested-Goal SL, 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 [2]). For the one-goal fragment, namely Graded$_{\mathbb{N}}$SL[1g], the model checking problem is instead 2-ExpTime-complete. All model checking complexities reported so far refer to the size of formula. Instead, with respect to the size of the model, the model-checking problem is PTime-complete for all these cases.

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 [15] 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 [16] and *number restrictions* in *description logics* [17]. *Graded μ-calculus*, in which immediate-successor accessible worlds are counted, was introduced to reason about graded modal logic with fixed-point operators [18]. Recently, the notion of graded modalities was extended to count the number of paths in the branching-time temporal logic formulas CTL and CTL* [19, 20]. In the verification of reactive systems, we mention two orthogonal approaches: module checking for graded μ-calculus [21, 22] and an extension of ATL by graded path modalities [23].

The work closest to ours is [24]: 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 gives a semantic definition for when two strategies should be considered equal, and counts the number of equivalence classes. While this approach is justified, it leads to a complicated model-checking problem. Indeed, only a very weak fragment of GSL has been solved in [24] 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 [25]. 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 known 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 GradedSL and provide some preliminary related concepts. In Section 3 we address the model-checking problem for GradedSL and its fragments. In Section 4 we illustrate our logic by expressing the uniqueness of various solution concepts. We conclude with Section 5 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 GradedSL for short.

In the following we use a *finite set of variables* Vr, a *finite set of agents* Ag, and a *finite set of atomic propositions* AP. We denote variables by x_i, x_j, etc., agents by α_i, α_j, etc., and atomic propositions by p, q, etc. The assumption that these sets are finite is simply a technical convenience: the model-checking problem (Definition 3.1) takes as input formulas and arenas with any number of variables, agents, and atoms.

2.1. Syntax

GradedSL extends SL by replacing the 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 $g \in \mathbb{N} \cup \{\aleph_0, \aleph_1, 2^{\aleph_0}\}$ is called the *grade* 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.

Definition 2.1. GradedSL 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* $n \in \mathbb{N}$*, and* $g \in \mathbb{N} \cup \{\aleph_0, \aleph_1, 2^{\aleph_0}\}$*:*

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

Note that GradedSL formulas are defined *w.r.t.* fixed finite sets of atomic propositions AP, agents Ag, and variables Vr.

Notation. Whenever we write $\langle\langle x_1,\ldots,x_n\rangle\rangle^{\geq g}$ we mean that $x_i \neq x_j$ for $i \neq j$, i.e., the variables in a tuple are distinct (note that this does not mean

[1] In [26] 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* [27], 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 [26] state, it is not clear how to extend this result to the concurrent setting, even in the two player case.

that the strategies the variables represent are distinct). 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 $\mathsf{G}\,\varphi \triangleq \neg\,\mathsf{F}\,\neg\varphi$. Also, we have that $[\![x_1,\ldots,x_n]\!]^{<g}\varphi \triangleq \neg\langle\!\langle x_1,\ldots,x_n\rangle\!\rangle^{\geq g}\neg\varphi$. The operators $\langle\!\langle x_1,\ldots,x_n\rangle\!\rangle^{\geq g}$ (resp. $[\![x_1,\ldots,x_n]\!]^{<g}$) are called *existential (resp. universal) strategy quantifiers*.

In order to define the semantics, we first define the concept of *free placeholders* in a formula, which refer to agents and variables. 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 [2]. It is important for defining the model-checking procedure, in particular for the encoding of strategies as trees (Definition 3.3).

Definition 2.2. *The set of* free agents *and* free variables *of a* GRADEDSL *formula φ is given by the function* $\mathsf{free} : \text{GRADEDSL} \to 2^{\mathrm{Ag}\cup\mathrm{Vr}}$ *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*.*

Roughly, the *quantifier rank* of φ is the maximum, over all paths in the parse-tree of φ, of the number of strategy quantifiers that appear on the path, 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.

Definition 2.3. *The* quantifier rank *of a* GRADEDSL *formula is inductively defined as follows:*

- $\mathsf{qr}(p) \triangleq 0$*, where* $p \in \mathrm{AP}$*;*
- $\mathsf{qr}(\mathsf{OP}\varphi) \triangleq \mathsf{qr}(\varphi)$*, where* $\mathsf{OP} \in \{\neg, \mathsf{X}, \flat\}$*;*
- $\mathsf{qr}(\varphi_1 \mathsf{OP} \varphi_2) \triangleq \mathsf{max}(\mathsf{qr}(\varphi_1), \mathsf{qr}(\varphi_2))$ *where* $\mathsf{OP} \in \{\vee, \mathsf{U}\}$*;*
- $\mathsf{qr}(\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g} \varphi) \triangleq \mathsf{qr}(\varphi) + 1.$

Roughly, 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 a formula is like the quantifier rank except that a quantifier block of j quantifiers contributes 1 instead of j to the count. The formal definition follows:

Definition 2.4. *The* quantifier-block rank *of a* GRADEDSL *formula that uses the shorthand for universal strategy quantifiers, is inductively defined as follows:*

- $\mathsf{qbr}(p) \triangleq 0$*, where* $p \in \mathrm{AP}$*;*
- $\mathsf{qbr}(\mathsf{OP}\varphi) \triangleq \mathsf{qbr}(\varphi)$*, where* $\mathsf{OP} \in \{\neg, \mathsf{X}, \flat\}$*;*
- $\mathsf{qbr}(\varphi_1 \mathsf{OP} \varphi_2) \triangleq \mathsf{max}(\mathsf{qbr}(\varphi_1), \mathsf{qbr}(\varphi_2))$ *where* $\mathsf{OP} \in \{\vee, \mathsf{U}\}$*;*
- $\mathsf{qbr}(\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g} \varphi) \triangleq \mathsf{qbr}(\varphi)$ *if* φ *begins with an existential strategy quantifier, and* $\mathsf{qbr}(\varphi) + 1$ *otherwise.*
- $\mathsf{qbr}([[x_1, \ldots, x_n]]^{<g} \varphi) \triangleq \mathsf{qbr}(\varphi)$ *if* φ *begins with a universal strategy quantifier, and* $\mathsf{qbr}(\varphi) + 1$ *otherwise.*

Note that we treat $\neg\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g} \neg\varphi$ differently to $[[x_1, \ldots, x_n]]^{<g} \varphi$. Thus, one should choose the existential and universal quantifiers judiciously in order to obtain a low quantifier-block rank.

2.2. Models

Sentences of GRADEDSL are interpreted over *arenas* [2], just as for ATL and SL [3, 2].

[2]This is sometimes called a Concurrent Game Structure.

Definition 2.5. *An* arena *over fixed sets of* AP *and* Ag*, is a tuple* $\mathcal{A} \triangleq \langle \mathrm{Ac}, \mathrm{St}, s_I, \mathsf{ap}, \mathsf{tr} \rangle$*, where:*

- Ac *is a finite set of* actions*,*
- St *is a finite set of* states*,*
- $s_I \in \mathrm{St}$ *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,* $\mathsf{tr} : \mathrm{Dc} \to (\mathrm{St} \to \mathrm{St})$*, a* transition function*, maps 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 $\mathsf{tr}(\delta)(s_i) = s_{i+1}$. Given a path $\pi = s_1 s_2 \ldots$, with $\lambda(\pi)$ we denote the *label of* π as a sequence of sets of atomic propositions $\pi_1, \pi_2, \ldots$ where $\mathsf{ap}(s_1) = \pi_1$, $\mathsf{ap}(s_2) = \pi_2$, and so on. The set of paths starting 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 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, [28], 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.

2.3. Semantics

As for SL, the interpretation of a GradedSL formula requires a valuation of its free placeholders.

Definition 2.6. *An* assignment (from s) *is a function* $\chi \in \mathrm{Asg}(s) \triangleq (\mathrm{Vr} \cup \mathrm{Ag}) \to \mathrm{Str}(s)$ *mapping variables and 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 an assignment ensures that all free variables are associated with strategies, it induces a play.

Definition 2.7. *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 2.8. *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 $\chi(e)(\rho_{<i} \cdot h)$.*

The semantics of GradedSL mimics the one for SL as given in [2]. Given an arena $\mathcal{A}$, for all states $s \in \mathrm{St}$ and assignments $\chi \in \mathrm{Asg}(s)$, we now define the relation $\mathcal{A}, \chi, s \models \varphi$, read φ *holds at s in $\mathcal{A}$ under χ*.

Definition 2.9. *Fix an arena $\mathcal{A}$. For all states $s \in \mathrm{St}$ and assignments $\chi \in \mathrm{Asg}(s)$, the relation $\mathcal{A}, \chi, s \models \varphi$ is defined inductively on the structure of φ:*

- $\mathcal{A}, \chi, s \models p$ *iff* $p \in \mathsf{ap}(s)$*;*
- $\mathcal{A}, \chi, s \models \neg\varphi$ *iff* $\mathcal{A}, \chi, s \not\models \varphi$*;*
- $\mathcal{A}, \chi, s \models \varphi_1 \vee \varphi_2$ *iff* $\mathcal{A}, \chi, s \models \varphi_1$ *or* $\mathcal{A}, \chi, s \models \varphi_2$*;*
- $\mathcal{A}, \chi, s \models \mathsf{X}\,\varphi$ *iff* $\mathcal{A}, \chi_1, \mathsf{play}(\chi, s)_1 \models \varphi$*;*
- $\mathcal{A}, \chi, s \models \varphi_1 \,\mathsf{U}\, \varphi_2$ *iff there is an index $i \in \mathbb{N}$ such that $\mathcal{A}, \chi_i, \mathsf{play}(\chi, s)_i \models \varphi_2$ and, for all indexes $j \in \mathbb{N}$ with $j < i$, it holds that $\mathcal{A}, \chi_j, \mathsf{play}(\chi, s)_j \models \varphi_1$;*
- $\mathcal{A}, \chi, s \models (\alpha, x)\varphi$ *iff* $\mathcal{A}, \chi[\alpha \mapsto \chi(x)], s \models \varphi$*;*

- $\mathcal{A}, \chi, s \models \langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g} \varphi$ *iff there exist g many n-tuples of strategies $\overline{\sigma_i}$ ($0 \leq i < g$) such that:*
  - $\overline{\sigma_i} \neq \overline{\sigma_j}$ *for* $i \neq j$, *and*
  - $\mathcal{A}, \chi[\overline{x} \mapsto \overline{\sigma_i}], s \models \varphi$ *for* $0 \leq i < g$.

Intuitively, $\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g} \varphi$ expresses that the number of distinct tuples of strategies that satisfy φ is at least g.

As usual, if χ and χ' agree on $\mathsf{free}(\varphi)$, then $\mathcal{A}, \chi, s \models \varphi$ if and only if $\mathcal{A}, \chi', s \models \varphi$, 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 $\mathcal{A} \models \varphi$ to mean that $\mathcal{A}, \chi, s_I \models \varphi$ for some (equivalently, for all) assignments χ, and where s_I is the initial state of $\mathcal{A}$.

2.4. Fragments of GradedSL

In this section we introduce various syntactic fragments of GradedSL. Obviously SL can be considered a fragment of GradedSL: note that the GradedSL quantifier $\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g}$ in case $g = 1$ and $n = 1$ has the same semantics as the SL quantifier $\langle\langle x_1 \rangle\rangle$. The next result shows that GradedSL is strictly more expressive than SL, i.e., there is a GradedSL sentence whose models are not the set of models of any SL sentence.

Theorem 2.1. GradedSL *is strictly more expressive than* SL.

Proof. Fix $\mathrm{Ag} = \{\alpha\}$, $\mathrm{AP} = \{p\}$, $\mathrm{St} = \{s\}$, $s_I = s$, $\mathsf{ap}(s) = \{p\}$. Define $\mathcal{A} = \langle \mathrm{Ac}_A, \mathrm{St}, s_I, \mathsf{ap}, \mathsf{tr}_A \rangle$, where $\mathrm{Ac}_A = \{0\}$, and $\mathsf{tr}_A(\delta)(s) = s$ for every decision δ; and $\mathcal{B} = \langle \mathrm{Ac}_B, \mathrm{St}, s_I, \mathsf{ap}, \mathsf{tr}_B \rangle$, where $\mathrm{Ac}_B = \{0, 1\}$, and $\mathsf{tr}_B(\delta)(s) = s$ for every decision δ. Thus, each arena consists of a single state with self-loops, the difference being that in $\mathcal{B}$ there are two actions while in $\mathcal{A}$ there is only a single action.

Consider the GradedSL formula $\neg\langle\langle x \rangle\rangle^{\geq 2}\mathsf{true}$. Note that $\mathcal{A} \models \neg\langle\langle x \rangle\rangle^{\geq 2}\mathsf{true}$ (since there is only a single strategy in $\mathcal{A}$), while $\mathcal{B} \not\models \neg\langle\langle x \rangle\rangle^{\geq 2}\mathsf{true}$ (since there are at least two, and in fact $2^{\aleph_0}$ many, strategies in $\mathcal{B}$).

Let χ_A be the unique assignment in $\mathcal{A}$, i.e., that maps α and every variable in Vr to the strategy σ defined by $\sigma(h) = 0$ for all histories h. We claim that, for every SL formula φ, if $\mathcal{A}, \chi_A, s \models \varphi$ then, for all assignments χ we have that $\mathcal{B}, \chi, s \models \varphi$. Thus, in particular, no SL sentence can distinguish between $\mathcal{A}$ and $\mathcal{B}$, and the theorem follows.

One can easily prove the claim by induction on the structure of an SL formula. Alternatively, one may note that $\mathcal{A}$ and $\mathcal{B}$ are locally-isomorphic, and thus agree on all SL formulas (see [29, Section 3] for the definition and properties of "local-isomorphism").[3] □

[3]We thank an anonymous reviewer for pointing this out.

Recall that SL has a few natural syntactic fragments, the most powerful of which is SL[NG] (here "NG" stands for Nested-Goal). Recall that in 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. Also, formulas with free agents are not allowed. We define GRADEDSL[NG] in a similar way, as follows.

A *quantification prefix* over a set $\mathrm{V} \subseteq \mathrm{Vr}$ of variables is a sequence $\wp$ from the set

$$\{\langle\langle x_1, \dots, x_n\rangle\rangle^{\geq g}, [[x_1, \dots, x_n]]^{<g} : n \in \mathbb{N}, x_1, \dots, x_n \in \mathrm{V} \wedge g \in \mathbb{N} \cup \{\aleph_0, \aleph_1, 2^{\aleph_0}\}\}^*$$

such that each $x \in \mathrm{V}$ occurs exactly once in $\wp$. A *binding prefix* is a sequence $\flat \in \{(\alpha, x) : \alpha \in \mathrm{Ag} \wedge x \in \mathrm{Vr}\}^*$ 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 2.10. GRADEDSL[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 $\wp \in \mathrm{Qn}(\mathsf{free}(\varphi))$. In this case we call $\wp\varphi$ a principal *formula.*

Formulas of GRADEDSL[NG] can be classified according to their *alternation number*, i.e., the maximum number of quantifier switches in a quantification prefix. [4] Formally:

Definition 2.11. *The* alternation number *of a* GRADEDSL[NG] *formula is given by:*

- $\mathsf{alt}(p) \triangleq 0$*, where* $p \in \mathrm{AP}$*;*
- $\mathsf{alt}(\mathsf{OP}\varphi) \triangleq \mathsf{alt}(\varphi)$*, where* $\mathsf{OP} \in \{\neg, \mathsf{X}, \flat\}$*;*
- $\mathsf{alt}(\varphi_1 \mathsf{OP} \varphi_2) \triangleq \mathsf{max}(\mathsf{alt}(\varphi_1), \mathsf{alt}(\varphi_2))$ *where* $\mathsf{OP} \in \{\vee, \mathsf{U}\}$*;*

[4] In [2] the alternation number is described for all formulas of SL, but only used for the Nested-Goal fragment. Thus, here, we only define it for GRADEDSL[NG].

- $\mathsf{alt}(\wp\varphi) \triangleq \mathsf{max}(\mathsf{alt}(\varphi), \mathsf{alt}(\wp))$ *where* $\wp$ *is a quantification prefix and* $\mathsf{alt}(\wp) \triangleq \sum_{i=1}^{|\wp|-1} \mathsf{switch}(\wp_i, \wp_{i+1})$*, where* $\mathsf{switch}(Q, Q') = 0$ *if* Q *and* Q' *are either both universal or both existential quantifiers, and* 1 *otherwise.*

Another important fragment of SL is SL[1G] (here "1G" stands for One-Goal). Intuitively, SL[1G] is the fragment of GradedSL[NG] in which quantification is immediately followed by binding. The importance of this fragment 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 [25, 2]. However, it is commonly believed that Nash Equilibrium cannot be expressed in this fragment. Similarly, we give the following definition of GradedSL[1G]:

Definition 2.12. GradedSL[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* $\flat \in \mathrm{Bn}$*:*

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

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

Finally, an important fragment (in which one can express uniqueness of strategy profiles) is when all grades are in $\mathbb{N}$.

Definition 2.13. *We write* Graded$_{\mathbb{N}}$SL*,* Graded$_{\mathbb{N}}$SL[1G]*, and* Graded$_{\mathbb{N}}$SL[NG] *for the fragments in which all grades are from the set* $\mathbb{N}$*.*

3. Model-checking GradedSL

In this section we study the model-checking problem for GradedSL 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.

Definition 3.1. *The* model-checking problem for GradedSL (resp. GradedSL[NG]) *is the following decision problem: given a formula* φ *from* GradedSL *(resp.* GradedSL[NG]*) over some finite sets of atoms* AP*, agents* Ag*, and variables* Vr*, and given an arena* $\mathcal{A}$ *over the sets* AP *and* Ag*, decide if* $\mathcal{A} \models \varphi$*.*

When measuring computational complexity, the grades in formulas are written in unary.

For the algorithmic procedures, we follow an *automata-theoretic approach* [30], 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 [2]. The only case that is different is the new graded quantifier over tuples of strategies, i.e., we show how to convert a GRADEDSL formula φ into an automaton that accepts exactly the (tree encodings) of the assignments that satisfy φ.

Tree Automata. 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 [31]. *Alternating tree automata* (ATA) are a further generalization of nondeterministic tree automata [32]. 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 [30].

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

Definition 3.2. *An Alternating Parity Tree-Automaton (*APT*) is a tuple* $\mathcal{M} \triangleq \langle \Sigma, \Delta, \mathrm{Q}, \delta, q_0, \mathrm{F} \rangle$*, 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*,*
- $\delta : \mathrm{Q} \times \Sigma \to \mathsf{B}^+(\Delta \times \mathrm{Q})$ *is an* alternating transition function*, and*
- F*, 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 $\mathcal{M}$ 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* $\mathrm{F} = (\mathrm{F}_1, \ldots, \mathrm{F}_k)$ if the least index $i \in [1, k]$ for which $\mathsf{inf}(u) \cap \mathrm{F}_i \neq \emptyset$ is even. A run is *successful* if all its branches satisfy the parity acceptance condition F. An APT *accepts* an input tree T iff there exists a successful run R of $\mathcal{M}$ on T.

The *language* $\mathrm{L}(\mathcal{M})$ of the APT $\mathcal{M}$ is the set of trees T accepted by $\mathcal{M}$. Two automata are *equivalent* if they have the same language. The *emptiness problem* for alternating parity tree-automata is to decide, given $\mathcal{M}$, whether $\mathrm{L}(\mathcal{M}) = \emptyset$. The *universality problem* is to decide whether $\mathcal{M}$ accepts all trees.

3.1. From Logic to Automata

We reduce the model-checking problem of GRADEDSL to the emptiness problem for alternating parity tree automata [2]. The main step is to translate every GRADEDSL formula φ (i.e., φ may have free placeholders), arena $\mathcal{A}$, and state s, into an APT that accepts a tree if and only if the tree encodes an assignment χ such that $\mathcal{A}, \chi, s \models \varphi$.

We first describe the encoding, following [2]. Informally, the arena $\mathcal{A}$ 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 3.3. *The* encoding of χ (w.r.t. $\varphi, \mathcal{A}, s$) *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* $\mathcal{A}$ *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* $e \in \mathsf{free}(\varphi)$*.*[5]

We now state and prove a lemma that says one can translate every GRADEDSL formula into an APT. It is proved by induction on the structure of the formula φ, as in [2]. The idea for handling the new case, i.e., the graded quantifier $\langle\!\langle x_1, \ldots, x_n \rangle\!\rangle^{\geq g} \psi$, is to build an APT that is a projection of an APT that itself checks that each of the g tuples of strategies satisfies ψ and that each pair of g tuples is distinct. The case that $g \in \mathbb{N}$ directly builds the required automaton (as is done for SL [2]), while the case that $g \in \{\aleph_0, \aleph_1, 2^{\aleph_0}\}$ goes through logic. Write MSOL for monadic second-order logic in the signature of trees. We use the following two results that show how to express counting quantifiers in MSOL. The first is due to Rabin 2.9 and is nicely exposed in [31].

[5] 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.

Theorem 3.1 (MSOL and Automata). *For every MSOL formula $\alpha(Y)$ there exists an* Apt *accepting the set of trees Y such that $\alpha(Y)$ holds. Conversely, for every* Apt *there is an MSOL formula $\alpha(Y)$ that holds on those Y that are accepted by the* Apt*.*

The second is due to [33] and is proved using the composition technique.

Theorem 3.2. *For every MSOL formula $\alpha(\overline{X}, Y)$ and $\kappa \in \{\aleph_0, \aleph_1, 2^{\aleph_0}\}$ there exists an MSOL formula $\beta(\overline{X})$ equivalent to "there exist κ many trees Y such that $\alpha(\overline{X}, Y)$".*

Lemma 3.1. *For every* GradedSL *formula φ, arena $\mathcal{A}$, and state $s \in \mathrm{St}$, there exists an* Apt *$\mathcal{M}$ such that for all assignments χ, if T is the encoding of χ (w.r.t. $\varphi, \mathcal{A}, s$), then $\mathcal{A}, \chi, s \models \varphi$ iff $\mathsf{T} \in \mathrm{L}(\mathcal{M})$.*

Proof. As in [2] we induct on the structure of the formula φ to construct the corresponding automaton $\mathcal{M}$. The Boolean operations are easily dealt with using the fact that disjunction corresponds to non-determinism, and negation corresponds to dualising the automaton. 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 SL 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 $\overline{\sigma_i}$ 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 $0 \leq i < g$.

There are two cases.

Case $g \in \{\aleph_0, \aleph_1, 2^{\aleph_0}\}$**.** Consider $\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g} \psi$. By induction, there is an Apt $\mathcal{D}$ for ψ. Apply Theorem 3.1 to translate $\mathcal{D}$ into an MSOL formula α, then apply Theorem 3.2 to get an MSOL formula β that holds iff "there exist κ many tuples of trees such that α" (recall that a tuple of trees is coded as a single tree). Finally, apply Theorem 3.1 to convert β into the required Apt.

Note that the blowup in the translations (MSOL to Apt, and closure under "there exists κ many trees") is non-elementary.

Case $g \in \mathbb{N}$**.** Let $\mathcal{M}$ be the Apt for ψ, given by induction. We show how to build an Npt for φ that mimics the definition of φ: it will be a projection of an Apt, 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 $X \triangleq \{x_i^j : i \leq n, j \leq g\}$, 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, there is an APT $\mathcal{B}_{\mathcal{M},X}$ for ψ' (conjunction can be dealt using universal-choice). Note that the input alphabet for $\mathcal{B}_{\mathcal{M},X}$ is $(\mathsf{free}(\psi') \to \mathrm{Ac}) \times \mathrm{St}$ and that $X \subseteq \mathsf{free}(\psi')$.

On the other hand, let $\mathcal{C}_X$ 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 $\mathcal{D}_{\mathcal{M},X}$ for the intersection of $\mathcal{B}_{\mathcal{M},X}$ and $\mathcal{C}_X$ (formed using universal-choice).

Now, using the classic transformation [34], we remove alternation from the APT $\mathcal{D}_{\mathcal{M},X}$ 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. $\square$

We now analyse the number of states of the constructed APT. All the cases in the induction incur at most a linear blowup except for the quantification case. For the quantification case, in case $g \in \{\aleph_0, \aleph_1, 2^{\aleph_0}\}$ the blowup is non-elementary.

In case $g \in \mathbb{N}$ then the translation incurs an exponential blowup. Indeed, the number of states of the APT $\mathcal{B}_{\mathcal{M},X}$ is $g \times n$ times the number of states of the APT for ψ, and since $\mathcal{C}_X$ consists of the conjunction of $g(g-1)$ automata (one for each pair of tuples), and each such automaton has $O(n)$ many states, the number of states of $\mathcal{C}_X$ is $O(ng^2)$. Thus, the number of states of the APT $\mathcal{D}_{\mathcal{M},X}$ is polynomial in the number of states of the APT for ψ. Finally, the translation from an APT to an NPT results in an exponentially larger automaton [30].

In case all grades are from $\mathbb{N}$ and the formulas are written using the universal-strategy quantifier shorthand, we can easily modify the construction to handle quantifier-blocks in one shot as if they were a single quantifier, i.e., with a single exponential blowup. For instance, suppose $\phi = \langle\!\langle y_1, \ldots, y_m \rangle\!\rangle^{\geq h} \langle\!\langle x_1, \ldots, x_n \rangle\!\rangle^{\geq g} \psi$ (additional quantifiers are treated similarly). As in the proof, let $\mathcal{M}$ be the APT for ψ and take $\mathcal{D}_{\mathcal{M},X}$. Now, instead of immediately removing alternation and projecting, build $\mathcal{D}_{\mathcal{M}',Y}$ where $\mathcal{M}'$ is $\mathcal{D}_{\mathcal{M},X}$ and $Y \triangleq \{y_i^j : i \leq m, j \leq h\}$. Finally, remove alternation from $\mathcal{D}_{\mathcal{M}',Y}$ to get an NPT $\mathcal{N}'$, and then apply the $(X \cup Y)$-projection to the language of $\mathcal{N}'$ to get the desired APT for ϕ. Note that the size of $\mathcal{D}_{\mathcal{M}',Y}$ is exponential in the number of states of $\mathcal{M}$ since the costly step of removing alternation is performed only once. Similarly, to deal with a block of universal strategy quantifiers simply use dualisation. For instance, to deal with $\phi = [\![y_1, \ldots, y_m]\!]^{<h} [\![x_1, \ldots, x_n]\!]^{<g} \psi$ apply the previous procedure to the equivalent formula $\neg \langle\!\langle y_1, \ldots, y_m \rangle\!\rangle^{\geq h} \langle\!\langle x_1, \ldots, x_n \rangle\!\rangle^{\geq g} \neg \psi$ (recall that negating an Apt is done by dualisation, which incurs no blowup).

Theorem 3.3. *The model-checking problem for* GradedSL *is decidable. Regarding complexity:*

1. *The complexity is not bounded by any fixed tower of exponentials.*
2. *The complexity is* PTime-complete *w.r.t. the size of the model.*
3. *If all grades are restricted to be in* $\mathbb{N}$*, then:*
   (a) *the model-checking problem is in* $(k+1)$ExpTime *if* $k \geq 1$ *is the quantifier-block rank of* φ*.*
   (b) *if* φ *is the form* $\wp\psi$*, where* $\wp$ *is a quantifier-block, and* ψ *is of quantifier-block rank* $k-1$*, then the model-checking problem is in* kExpTime*.*
   (c) *if* φ *is of the form* $\langle\!\langle x_1, \ldots, x_n \rangle\!\rangle^{\geq g} \psi$ *or* $[\![x_1, \ldots, x_n]\!]^{<g} \psi$ *then the model-checking problem w.r.t.* g *written in unary is in* ExpTime*.*

Proof. The lower bounds already hold for SL [2]. For decidability, use Lemma 3.1 to transform the arena and φ into an Apt and test its emptiness. For the upper bound in item 2, use the fact that the membership problem for Apt is in PTime in the number of states. For item 3(a), proceed as follows. The complexity of checking emptiness (resp. universality) of an Apt is in ExpTime in the number of states [30]. As discussed after Lemma 3.1, for the case that all grades are in $\mathbb{N}$, the number of states of the Apt is a tower of exponentials whose height is the quantifier-block rank of φ. This gives the $(k+1)$ExpTime bound. Finally, 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, the Apt $\mathcal{D}_\psi$, whose number of states 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 solved in kExpTime. For item 3(c), first observe that the size of the Apt constructed in Lemma 3.1 grows quadratically in g. The statement follows by recalling that the complexity of model-checking formulas of this form is exponential in the number of states of the Apt. □

Theorem 3.4. *The model-checking problem for* GradedSL[ng] *is* $(k+1)$-ExpTime *when restricted to formulas of maximum alternation number* k *and grades in* $\mathbb{N}$*.*

Proof. The lower bound already holds for SL[NG] [2]. For the upper bound, as for SL[NG], note that principal formulas are "state formulas", i.e., their truth value only depends on the state in which they are interpreted (this is because they have no free placeholders). Thus, one can replace the general algorithm in Lemma 3.1 with the following marking algorithm. Bottom up, for every principal subformula $\varphi = \wp\psi$ and state s of $\mathcal{A}$, mark s by the truth value of $\mathcal{A}, \chi, s \models \varphi$ (for some, equivalently all, assignments χ). Consider these markings as new atomic propositions. Observe that the complexity of marking a state is at most $k+1$-EXPTIME (by repeatedly applying Theorem 3.3 part 3.). Also, the cost of the whole marking algorithm is the sum of the costs of all the marking rounds, and the number of rounds is at most the size of the formula. Thus the total time is at most $k + 1$-EXPTIME. □

We conclude this section with the complexity of the model checking problem for GRADED$_{\mathbb{N}}$SL[1G]. In this case one can derive the lower bound from the one holding for the corresponding sub-logic in SL (*i.e.*, SL[1G]) and the upper bound by using the same algorithm for SL[1G] but using the (no more complex) construction for the strategic quantifier $\langle\langle x_1, \ldots, x_n \rangle\rangle^{\geq g}$ instead of $\langle\langle x \rangle\rangle$. Indeed the model checking problem for GRADED$_{\mathbb{N}}$SL[1G]is 2EXPTIME-COMPLETE. It is worth recalling that SL[1G] strictly subsumes ATL* [2]. 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 GRADEDSL[1G] are reported in the following theorem.

Theorem 3.5. *The model-checking problem for* GATL* *and* GRADED$_{\mathbb{N}}$SL[1G] *is* PTIME-COMPLETE w.r.t. *the size of the model and* 2-EXPTIME-COMPLETE *in the size of formula.*

4. Analysing Games using GRADEDSL

In this section we describe how to use the models and formulas of GRADEDSL to reason about solution concepts from game theory. In particular, we show how to use arenas to model games of finite or infinite duration, and GRADEDSL to express the uniqueness of winning strategies, Nash equilibria, subgame-perfect equilibria, and Pareto-efficient profiles. In all cases this is not more expensive than merely deciding the existence of winning strategies with LTL-goals, i.e., 2EXPTIME.

4.1. Strategic Form and Infinitely Repeated Games

The *Strategic Form* is the most familiar representation of strategic interactions in Game Theory. A game written in this way amounts to a representation

	C	D
C	$-1, -1$	$-4, 0$
D	$0, -4$	$-3, -3$

Figure 1: Prisoner's Dilemma in Strategic Form. Each row corresponds to a possible action for player 1, each column corresponds to a possible action for player 2, and each cell corresponds to one possible outcome. Payoffs of the players for an outcome are written in the corresponding cell, with the payoff of player 1 listed first.

of every player's preference for every state of the world, in the special case where states of the world depend only on the players' combined actions.

Definition 4.1. *A* strategic form game *is a tuple* $(N, A, (\preceq_i)_{i\in N})$*, where:*

- *N is a finite set of n players, indexed by i;*
- *$A = A_1 \times \ldots \times A_n$, where A_i is a finite set of actions available to player i. Each vector $a = (a_1, \ldots, a_n) \in A$ is called an action profile;*
- *each $\preceq_i$ is a total pre-order (i.e., reflexive and transitive) on A.*

Note that a common way to give the preference relation is by using a *payoff function pay* : $A \to \mathbb{R}$, which assigns a real number to every element in A. In this case, the preference relation $\preceq_i$ is defined by having $a \preceq_i a'$ iff $pay(a) \leq pay(a')$.

A classic way to model players that repeatedly interact with each other in a game $(N, A, (\preceq_i)_{i\in N})$ is by *infinitely repeated games*, see e.g., [35]. We will illustrate by formalising an iterated prisoner's dilemma (Section 4.3).

*4.2. Quasi-Quantitative Games and Objective-*LTL *Games*

As expected, we can also specify games by arenas and a payoff function on plays. In this section we define quasi-quantitative games, a generalisation of *objective*-LTL *games* [6].

Let $\mathcal{A}$ be an arena with n agents. Let $m \in \mathbb{N}$, for each agent $\alpha_i \in \mathrm{Ag}$, a *quasi-quantitative objective* is a tuple $S_i \triangleq \langle \mathsf{f}_i, L_i^1, \ldots, L_i^m \rangle$, where $\mathsf{f}_i : \{0,1\}^m \to \mathbb{Z}$, and each L_i^j is a set of sequences of sets of atomic propositions. If π is an infinite path, then agent α_i receives payoff $\mathsf{f}_i(b_i^\pi) \in \mathbb{N}$ where the j'th bit of b_i^π is 1 if and only if $\mathsf{ap}(\pi) \in L_i^j$. We assume agents are trying to maximise their payoffs. The tuple $\mathcal{G} = \langle \mathcal{A}, S_1, \ldots, S_n \rangle$ is called a *quasi-quantitative game.* In case each $f_i : \{0,1\}^m \to \{-1,1\}$ we say that the game is *win/lose.* If $\sum_{1\leq i\leq n} \mathsf{f}_i(b_i^\pi) = 0$ for all π, then $\mathcal{G}$ is *zero-sum*, otherwise it is a *non zero-sum.*

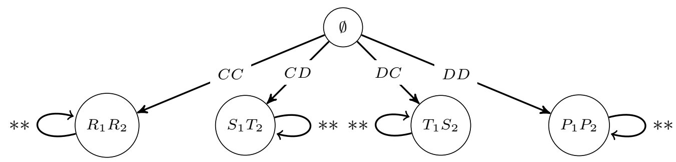

Figure 2: Arena of the Prisoner's dilemma.

In case each L_i^j is the set of models of an LTL formula φ_i^j over AP, we call $\mathcal{G}$ an *objective*-LTL *game*. We introduce the following useful shorthand. For $\alpha_i \in \mathrm{Ag}$ and $a \in \{0,1\}^m$, define η_i^a for the LTL formula $\bigwedge_{j \leq m} \psi_{a,j}$ where $\psi_{a,j} = \varphi_i^j$ if $a_j = 1$ and $\psi_{a,j} = \neg\varphi_i^j$ if $a_j = 0$.

4.3. Example: The Prisoner's Dilemma (PD)

A natural way to draw strategic form games is via an n-dimensional matrix. Figure 1 contains the matrix of the classic Prisoner's Dilemma.

The two actions are C (co-operate) and D (defect). The payoffs are the prison sentences (in years) that the prisoners get for each pair of actions that they choose.[6]

The deal is that if both confess then they each get a reduced sentence. If both is offered the choice to confess or remain silent. captured thieves are suspected of Observe that the actual numbers are not important, and that the important thing is the induced preference relation $\mathbf{T}_i > \mathbf{R}_i > \mathbf{P}_i > \mathbf{S}_i$, where $\mathbf{R}_i$ represents the *reward* 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.

We can describe the Prisoner's Dilemma with the arena in Figure 2 and, for agent α_i, the objective $S_i \triangleq \langle \mathsf{f}_i, \varphi_i^1, \varphi_i^2, \varphi_i^3, \varphi_i^4 \rangle$ 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.*, $\mathsf{f}_i(0100) = 4$ that represents the payoff in which φ_i^3 is true. In words, we have two agents α_1 and α_2. Each agent has two actions, C and D. For each possible pair of moves, the game goes in a state whose atomic propositions represent the preferences.

It is well known that in the Prisoner's Dilemma the only Nash equilibrium is for both players to defect. The reason is that each prisoner must hedge against the possibility of the other one defecting. However, it is clear that if they would have both cooperated, they would be better off. If there was a way for one prisoner to later punish a defection of the other prisoner, it may not have to

[6]The story behind the dilemma is this: Two people have been arrested for robbing a bank and placed in separate isolation cells. Each has two possible choices, remaining silent (action C) or confessing (action D). 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 are both convicted, but 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.

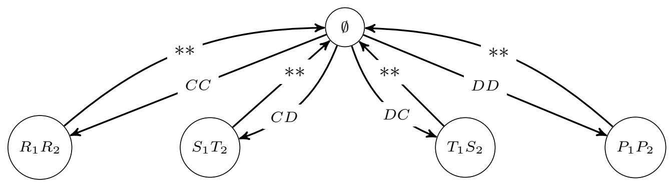

Figure 3: Arena of the Iterated Prisoner's dilemma.

hedge, and would be able to cooperate instead. Such behaviours emerge as a rational choice, for example, when one considers the infinitely repeated prisoner's dilemma, in which the prisoners repeat the basic strategic form game infinitely often.[7] Indeed, it is well known that for this iterated game (for example, with a payoff that is the mean-payoff of the prison sentences [36]), a new Nash equilibrium emerges, in which both players use the so called *Grim* strategy, in which a prisoner cooperates as long as the other prisoner cooperates, but switches to always defect the first time the other prisoner defects. Observe that the resulting infinite play of this Nash equilibrium has both players cooperating all the time. The core reason that the pair of grim strategies is a Nash equilibrium for the mean-payoff version of the iterated prisoner's dilemma is that this payoff ignores the price of being a 'sucker' on any finite prefix of the play, i.e., that the mean-payoff of a play is independent of any finite prefix of that play — other properties of the mean are not needed, and constitute unimportant noise. Indeed, the same Nash equilibria would emerge if, for example, one takes instead of the mean-payoff the maximal payoff that repeats infinitely often.

More generally, given a game in strategic form, and a preference relation $\preceq_i$ over its set of possible outcomes A, one can define a new preference relation $\preceq_i^\infty$ over A^ω by assigning a payoff to every subset of A and assigning to each play the set $inf(\pi) \subseteq A$ of outcomes in A that appear infinitely often in π. Formally, let $F_i : 2^A \to \mathbb{Z}$ be a function mapping subsets of A to integers, and define $\pi \preceq_i^\infty \pi'$ iff $F_i(inf(\pi)) \leq F_i(inf(\pi'))$.[8] For example, for the iterated prisoners' dilemma, setting $F_i(X)$ to be the number of elements y in A such that $y \prec_i x$ where x is a $\preceq_i$-maximal element of X, results in a game with the same set of Nash equilibria as in the mean payoff version.

We formalise the infinitely repeated prisoner's dilemma as an objective-LTL game. The arena is in Figure 3. The preferences, for agent α_i, are defined 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 as before.

[7] Alternatively, one can introduce the threat of a punishment for defecting by considering a probabilistic version in which it is unclear to the prisoners how many repetitions will be used. Note, however, that a fixed number of repetitions turns out to be essentially the same as playing only once [36].

[8] This is reminiscent of the Muller acceptance condition in automata theory.

4.4. Illustrating GRADEDSL*: uniqueness of solutions*

We illustrate how to express some important solution concepts in Game Theory. We start with the concept of winning strategy that is useful in zero-sum games, and then we analyse the well known solution concepts, such as Nash and subgame-perfect equilibria, that are used in non zero-sum games. We use ordinary SL quantifiers (i.e., $\langle\langle x\rangle\rangle, [[x]]$) since by Theorem 2.9, these are expressible in GRADEDSL.

4.4.1. Winning strategies

In two-player win-lose zero-sum games the main solution concept is the *winning strategy*. That is, if G is such an objective-LTL game, then a strategy for agent α_1 is *winning* if and only if for all strategies of agent α_2, the resulting induced play has payoff 1 for agent i. This can be expressed in SL as follows:

$$\phi_{WS}(x) \triangleq [[y]](\alpha_1, x)(\alpha_2, y) \bigvee_{f_1(a)=1} \eta_1^a$$

where η_1^a is the LTL formula defined in Section 4.2. Thus, the following formula expresses that there is a unique winning strategy for player 1:

$$\langle\langle x\rangle\rangle^{\geq 1}\phi_{WS}(x) \wedge \neg\langle\langle x\rangle\rangle^{\geq 2}\phi_{WS}(x) \tag{1}$$

Observe that this is a formula of GRADED$_{\mathbb{N}}$SL[NG] of alternation number 1. Thus, by Theorem 3.3 we get:

Theorem 4.1. *Deciding if a given player in a two-player zero-sum objective-*LTL *game has a unique winning strategy can be solved in* 2EXPTIME*.*

We illustrate with an example. In [37] the authors describe a two-player game named "Cop and the Robber", played in a maze, in which the objective of the Robber is to reach an exit (and thus the objective of the Cop is to ensure the Robber never reaches the exit). The authors describe two closely related mazes in which the Robber has, respectively, exactly one and exactly two winning strategies. Both these properties can be easily expressed by GSL. For instance, the Robber has a single LTL objective $\mathsf{F}\, exit$, and the following formula of GRADED$_{\mathbb{N}}$SL expresses that the Robber has exactly one winning strategy:

$$\langle\langle x\rangle\rangle^{\geq 1}[[y]](\text{Robber}, x)(\text{Cop}, y)\, \mathsf{F}\, exit \wedge \neg\langle\langle x\rangle\rangle^{\geq 2}[[y]](\text{Robber}, x)(\text{Cop}, y)\, \mathsf{F}\, exit.$$

4.4.2. Nash Equilibria

The central solution concept in non zero-sum games 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 the case that each agent α_i has a general objective tuple $S_i \triangleq \langle \mathsf{f}_i, \varphi_i^1, \ldots, \varphi_i^m \rangle$. Recall the definition of the LTL formulas η_i^a from Section 4.2.

For $\overline{x} \triangleq (x_1 \ldots x_n)$, the following formula says that if all agents follow $\overline{x}$, then no agent i gets a better payoff by deviating and following y_i:

$$\phi_{DEV}(\overline{x}, \overline{y}) \triangleq \bigwedge_{i=1}^{n} \bigwedge_{a \in \{0,1\}^m} (\flat(\overline{x}/y_i)\eta_i^a) \to \bigvee_{\mathsf{f}_i(a') \geq \mathsf{f}_i(a)} \flat(\overline{x})\eta_i^{a'}$$

where $\flat(\overline{x}) = (\alpha_1, x_1) \ldots (\alpha_n, x_n)$, and

$$\flat(\overline{x}/y_i) = (\alpha_1, x_1) \ldots (\alpha_{i-1}, x_{i-1})(\alpha_i, y_i)(\alpha_{i+1}, x_{i+1}) \ldots (\alpha_n, x_n).$$

Then, the following SL formula says that $\overline{x}$ is a NE

$$\phi_{NE}(\overline{x}) \triangleq [\![y_1]\!] \ldots [\![y_n]\!] \phi_{DEV}(\overline{x}, \overline{y}),$$

and the following $\text{GRADED}_{\mathbb{N}}\text{SL[NG]}$ formula expresses that there is a unique NE:

$$\langle\!\langle x_1, \ldots, x_n \rangle\!\rangle^{\geq 1} [\![y_1]\!] \ldots [\![y_n]\!] \phi_{DEV}(\overline{x}, \overline{y}) \wedge \neg \langle\!\langle x_1, \ldots, x_n \rangle\!\rangle^{\geq 2} [\![y_1]\!] \ldots [\![y_n]\!] \phi_{DEV}(\overline{x}, \overline{y})$$

Thus, by Theorem 3.3 we get:

Theorem 4.2. *Deciding if an objective-*LTL *game has a unique NE can be solved in* 2EXPTIME.

Rational synthesis can be formalised as the problem of deciding if a given game has a NE such that the resulting play satisfies a given LTL formula Ψ. In our setting, we get the following result by replacing ϕ_{DEV} by $\Psi \wedge \phi_{DEV}$ in the previous formula:

Theorem 4.3. *Deciding if an objective-*LTL *game has a unique NE satisfying an* LTL *formula* Ψ *can be solved in* 2EXPTIME.

!!!! below I talk about turn-based two-player games, because it was not clear to me what the existence of NE lower-bound result was using. Please feel free to change this!!!

Our goal now is to show that the upper bound presented in Theorem 4.2 is tight. We do this by a reduction from the problem of deciding, in a two-player zero-sum turn-based LTL game $\mathtt{G}$, whether Player 1 has exactly one winning strategy, which is problem known to be 2EXPTIME-hard [][9]. We first recap

[9] It is worth noting that the hardness of deciding whether there is exactly one winning strategy does not trivially follow from the hardness of deciding whether there is some winning strategy. Indeed, a negative answer to the first problem does not imply a negative answer to the second.

the reduction from [] which is used to reduce the problem of deciding whether, in such a two-player game $\mathtt{G}$, Player 1 has a (not necessarily unique) winning strategy, to the problem of deciding whether a suitably constructed concurrent 4-player objective-LTL game $\mathtt{G}'$ has a NE. Let φ be the goal of Player 1 in $\mathtt{G}$, one constructs $\mathtt{G}'$ by first converting it into a concurrent game[10] and then adding players 3, and 4. The actions (we assume w.l.o.g, that there are at least two available actions) of these new players have no effect on the truth value of φ. However, the payoff received by these players depends on this value, as well as on the result of a 'matching pennies' game that they play amongst themselves (the 'pennies' being their first actions, and Player 3 wins if the actions are equal, and Player 4 wins if they are different), as follows: if φ is true, both players 3 and 4 get a maximal payoff; and if φ is false, then the winner of the matching-pennies game gets a medium payoff and the loser gets zero payoff. It is not hard to see that the 4-player game has a NE iff player 1 has a strategy that enforces φ. Indeed, since matching-pennies has no NE (the loser is always better off deviating), a NE for this game must be such that the resulting play satisfies ϕ and, in addition, Player 2 cannot improve his payoff by deviating and falsifying φ. It follows that $\mathtt{G}'$ has a NE iff Player 1 has a winning strategy in $\mathtt{G}$.

Observe that the NE in $\mathtt{G}'$ are exactly the strategy profiles $(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ in which σ_1 is a winning Player 1 strategy in $\mathtt{G}$, and $\sigma_2, \sigma_3, \sigma_4$ are unconstrained. The proof of the following theorem modifies the reduction presented above, to obtain a reduction from the problem of uniqueness of winning strategies in two-player games to the problem of uniqueness of NE, by simply changing the payoff functions of players $2, 3$ and 4 so as to incentivise them to always stick to a single strategy.

Theorem 4.4. *Deciding if an objective-*LTL *game has a unique NE is* 2EXPTIME-*hard.*

Proof 4.1. *Given a two-player zero-sum turn-based* LTL *game* $\mathtt{G}$*, where Player* 1 *wins iff the formula* φ *holds, we construct a* 4*-player game* $\mathtt{G}''$*, by taking* $\mathtt{G}'$ *described above, but using different payoffs, as follows. Assume w.l.o.g, that there is more than one action, that one of the actions is called* a*, and let* σ_a *be the strategy in which the chosen action is always* a*. The payoffs of the players are:*

- *Player* 1 *gets a payoff of* 1 *if* φ *holds, and* 0 *otherwise.*

[10] Observe that any turn-based game can be viewed as a concurrent game in which the transition function has the property that it ignores the actions of the agent(s) except the one whose turn it is.

- *Player* 2 *gets a payoff of* 2 *if* φ *does not hold; a payoff of* 1 *if* φ *holds and he uses* σ_a*; and a payoff of* 0 *otherwise.*

- *Player* 3 *gets a payoff of* 2 *if* φ *holds and he uses* σ_a*; a payoff of* 1 *if* φ *does not hold but his first action was equal to that of Player* 4*; and a payoff of* 0 *otherwise.*

- *Player* 4 *gets a payoff of* 2 *if* φ *holds and he uses* σ_a*; a payoff of* 1 *if* φ *does not hold but his first action was different than that of Player* 3*; and a payoff of* 0 *otherwise.*

First, observe that the the different cases in the definitions of the payoffs can be easily captured by LTL *formulae. For example, the Player-*3 *condition '*φ *holds and he uses* σ_a*' is captured by the formula* $\varphi \wedge \mathsf{G}\, p_a^3$*, where* p_a^3 *is an atomic proposition that is true iff the last action performed by Player* 3 *was* a*. It follows that* $\mathtt{G}''$ *can be expressed as an objective-*LTL *game. Also observe that (*$\dagger$*) if* σ *is a winning strategy for Player* 1 *in* $\mathtt{G}$*, then the strategy profile* $(\sigma, \sigma_a, \sigma_a, \sigma_a)$ *is a NE in* $\mathtt{G}''$*. The theorem follows by showing that Player* 1 *has a unique winning strategy in* $\mathtt{G}$ *iff there is a unique NE in* $\mathtt{G}''$ *(and the fact that deciding the uniqueness of a winning strategy for* $\mathtt{G}$ *is* 2ExpTime*-hard).*

For the first direction, assume that Player 1 *has a unique winning strategy, and recall that* $(\sigma, \sigma_a, \sigma_a, \sigma_a)$ *is a NE. Note that it is the only NE since a strategy profile in which Player* 1 *does not follow* σ *is one in which (by the uniqueness assumption) he plays a non-winning strategy, and Player* 2 *can benefit by deviating; and that a profile in which Player* 1 *follows* σ *results in a play in which* φ *holds, and thus the other players get maximum payoff iff they follow* σ_a*. For the other direction, assume that there is a single NE. By* $\dagger$*, it follows that Player* 1 *has at most one winning strategy. Assume by contradiction that he has no winning strategy. It follows that the play resulting from the NE does not satisfy* φ *(otherwise Player* 2 *would deviate to achieve a play not satisfying* φ*). But this is a contradiction since the loser in the matching-pennies game between players* 3 *and* 4 *would be better off by deviating.*

4.4.3. Pareto efficiency

A strategy profile is said to be *Pareto efficient (PE)* if there is no other strategy profile that makes some agent better off without making another agent worse off. The formula $\phi_{PE}(\overline{x}) \triangleq [\![x'_1]\!] \dots [\![x'_n]\!]\psi(\overline{x}, \overline{x'})$ expresses that $\overline{x}$ is PE where $\psi(\overline{x}, \overline{x'})$ is

$$\bigwedge_{i \leq n} \bigwedge_{(a,a') \in X_i} \left((\flat(\overline{x})\eta_i^a \wedge \flat(\overline{x'})\eta_i^{a'}) \to \bigvee_{j \neq i} \bigvee_{(c,c') \in Y_i} (\flat(\overline{x})\eta_j^c \wedge \flat(\overline{x'})\eta_j^{c'}) \right),$$

where $(a, a') \in X_i$ iff $\mathsf{f}_i(a') > \mathsf{f}_i(a)$, where $(c, c') \in Y_i$ iff $\mathsf{f}_j(c') < \mathsf{f}_j(c)$, $\flat(\overline{x}) \triangleq (\alpha_1, x_1) \dots (\alpha_n, x_n)$, and $\flat(\overline{x'}) \triangleq (\alpha_1, x'_1) \dots (\alpha_n, x'_n)$. Using graded modalities, we can thus express that there is a unique PE using the following $\text{Graded}_{\mathbb{N}}\text{SL[ng]}$ formula of alternation number 1:

$$\langle\!\langle x_1, \dots, x_n \rangle\!\rangle^{\geq 1} \phi_{PE}(\overline{x}) \wedge \neg \langle\!\langle x_1, \dots, x_n \rangle\!\rangle^{\geq 2} \phi_{PE}(\overline{x}).$$

Thus, by using Theorem 3.3 we get:

Theorem 4.5. *Deciding if an objective-*LTL *game has a unique Pareto efficient profile can be solved in* 2ExpTime.

4.4.4. Subgame-perfect equilibria

Finally, we end with a discussion of the problem of deciding if a game has a unique subgame-perfect equilibrium, and give an upper bound.

It has been argued (in [38, 6]) that NE may be implausible when used for sequential games (of which infinitely repeating games are central examples), and that a more robust notion is subgame-perfect equilibrium [39].

Informally, a strategy profile is a subgame-perfect equilibrium if it is a NE in every reachable subgame. Here is the mathematical definition instantiated for quasi-quantitative games (following the definition in [35] for extensive-form games). Given a history $h \in \text{Hst}(s_I)$ ending in state s, say $h = us$, and a strategy $\sigma \in \text{Str}(s_I)$, the *h-translation* of σ is the strategy $\sigma_{|h} \in \text{Str}(s)$ that maps $h' \in \text{Hst}(s)$ to $\sigma(u \cdot h')$. Given a quasi-quantitative game $\mathcal{G} = \langle \mathcal{A}, S_1, \dots, S_n \rangle$, the profile $\sigma_1, \dots, \sigma_n$ is a *subgame-perfect equilibrium (SPE)* iff for all histories $h \in \text{Hst}(s_I)$, the profile $\sigma_{1|h}, \dots, \sigma_{n|h}$ is a NE in $\mathcal{G} = \langle \mathcal{A}_{|h}, S_{1|h}, \dots, S_{n|h} \rangle$ where $\mathcal{A}_{|h}$ is the same arena as $\mathcal{A}$ but with s as the initial state, and if $S = \langle \mathsf{f}_i, L_i^1, \dots, L_i^m \rangle$ then $S_{|h} = \langle \mathsf{f}_i, H_i^1, \dots, H_i^m \rangle$ where $\pi \in H_i^j$ iff $u \cdot \pi \in L_i^j$. The point is that the payoff in $\mathcal{G}_{|h}$ applies to the whole path (i.e., starting from s_I), even though the strategies only apply after h.

Using the notation in the previous paragraph, suppose each L_i^j is prefix-independent, i.e., $\pi \in L_i^j$ iff $\pi_{\geq n} \in L_i^j$ for all $n \geq 1$ (here $\pi_{\geq n}$ is the suffix of π starting at position n). In this case, $H_i^j = L_i^j$. Observe that the assumption that the objectives are prefix-independent is not too restrictive. Indeed, as discussed in Section 2.9, in many infinitely repeated games the outcome ignores all finite prefixes of the play.

Thus, suppose $\mathcal{G}$ is an objective-LTL game in which the set of models of each φ_i^j is prefix-independent. The following formula of SL expresses that $\overline{x}$ is an SPE:[11]

$$\phi_S(\overline{x}) \triangleq [\![z_1]\!] \ldots [\![z_n]\!][\![y_1]\!] \ldots [\![y_n]\!](\alpha_1, z_1) \ldots (\alpha_n, z_n)\, \mathsf{G}\, \phi_{DEV}(\overline{x}, \overline{y}).$$

Indeed, since $[\![\cdot]\!]$ commutes with G, the formula $\phi_S(\overline{x})$ is equivalent to

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

which is true in $\mathcal{A}, \chi, s_I$ iff for all histories h starting in s_I and ending, say, in state s, we have that $\mathcal{A}, \chi', s \models \phi_{NE}$ where the strategy $\chi'(x)$ is the h-translation of the strategy $\chi(x)$, i.e., the profile $\chi'(x_1), \ldots, \chi'(x_n)$ is a NE in $\mathcal{G}_{|h}$.

Using graded modalities, we can thus express there is a unique SPE (assuming each φ_i^j is prefix-independent) as the following $\text{GRADED}_{\mathbb{N}}\text{SL[NG]}$ formula of alternation number 1:

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

Thus, by using Theorem 3.3 we get:

Theorem 4.6. *Deciding if an objective-*LTL *game with prefix-independent objectives φ_i^j has a unique SPE can be solved in* 2EXPTIME.

5. 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 [1, 2, 4, 40, 41, 7, 5]. 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 [42].

One of the most important questions about NE in computational game theory is *"does the game admit more than one NE?"* [9, 10] — the unique NE problem. This problem is deeply investigated in game theory and is shown to be very challenging [11, 12, 13, 8, 14, 43, 44, 45]. Prior to this work, no logic-based technique, as far as we know, solved this problem.[12] In this paper

[11] Previous formalisations of SPE overlook the need for a condition like prefix-independence [38, 2, 6].

[12] 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.

we introduced GradedSL to address and solve the unique NE problem. We have demonstrated that GradedSL 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 illustrated that one can express the uniqueness of other solution concepts, including winning strategies, subgame-perfect equilibria, and Pareto-efficient profiles, all 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.

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 [46] with the graded quantifiers we introduced for GradedSL. Since strategic reasoning under imperfect information has an undecidable model-checking problem [47], 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" [48].

Finally, another direction is to implement GradedSL and its model-checking procedure in a formal verification tool. A reasonable approach would be, for example, to extend the tool SLK-MCMAS [46].

Acknowledgments

We thank Michael Wooldridge for pointing out the importance of uniqueness of Nash Equilibria in game theory. 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] F. Mogavero, A. Murano, M. Vardi, Reasoning About Strategies., in: FSTTCS'10, LIPIcs 8, Leibniz-Zentrum fuer Informatik, 2010, pp. 133–144. doi:10.4230/LIPIcs.FSTTCS.2010.133.

[2] F. Mogavero, A. Murano, G. Perelli, M. Vardi, Reasoning About Strategies: On the Model-Checking Problem., TOCL 15 (4) (2014) 34:1–42. doi:10.1145/2631917.

[3] R. Alur, T. Henzinger, O. Kupferman, Alternating-Time Temporal Logic., JACM 49 (5) (2002) 672–713. doi:10.1145/585265.585270.

[4] K. Chatterjee, T. Henzinger, N. Piterman, Strategy Logic., IC 208 (6) (2010) 677–693. doi:10.1016/j.ic.2009.07.004.

[5] J. Gutierrez, P. Harrenstein, M. Wooldridge, Reasoning about equilibria in game-like concurrent systems, in: KR 2014, AAAI, 2014.

[6] O. Kupferman, G. Perelli, M. Y. Vardi, Synthesis with rational environments, in: EUMAS 2014, 2014, pp. 219–235. doi:10.1007/978-3-319-17130-2_15.

[7] F. Belardinelli, A logic of knowledge and strategies with imperfect information, in: LAMAS 15, 2015.

[8] E. Altman, H. Kameda, Y. Hosokawa, Nash equilibria in load balancing in distributed computer systems, IGTR 4 (2) (2002) 91–100. doi:10.1142/S0219198902000574.

[9] G. Papavassilopoulos, J. B. Cruz, On the uniqueness of nash strategies for a class of analytic differential games, Journal of Optimization Theory and Applications 27 (2) (1979) 309–314. doi:10.1007/BF00933234.

[10] R. Cornes, R. Hartley, T. Sandler, An elementary proof via contraction, Journal of Public Economic Theory 1 (4) (1999) 499–509. doi:10.1111/1097-3923.00023.

[11] J. B. D. Simchi-Levi, X. Chen, The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management, Science and Business Media, Springer, 2013.

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

[13] L. Pavel, Game Theory for Control of Optical Networks, Science and Business Media, Springer, 2012. doi:10.1007/978-0-8176-8322-1.

[14] A. Orda, R. Rom, N. Shimkin, Competitive routing in multiuser communication networks, IEEE/ACM Trans. Netw. 1 (5) (1993) 510–521. doi:10.1109/90.251910.

[15] K. Fine, In So Many Possible Worlds., NDJFL 13 (1972) 516–520.

[16] E. Grädel, M. Otto, E. Rosen, Two-Variable Logic with Counting is Decidable., in: LICS'97, IEEE Computer Society, 1997, pp. 306–317.

[17] B. Hollunder, F. Baader, Qualifying Number Restrictions in Concept Languages., in: KR'91, Kaufmann, 1991, pp. 335–346.

[18] O. Kupferman, U. Sattler, M. Vardi, The Complexity of the Graded muCalculus., in: CADE'02, LNCS 2392, Springer, 2002, pp. 423–437. doi:10.1007/3-540-45620-1_34.

[19] A. Bianco, F. Mogavero, A. Murano, Graded Computation Tree Logic., TOCL 13 (3) (2012) 25:1–53. doi:10.1145/2287718.2287725.

[20] S. Rubin, B. Aminof, A. Murano, On CTL* with graded path modalities, in: LPAR 20, 2015.

[21] B. Aminof, A. Legay, A. Murano, O. Serre, μ-calculus pushdown module checking with imperfect state information, in: IFIP-TCS'08, Vol. 273 of IFIP, Springer, 2008, pp. 333–348.

[22] A. Ferrante, A. Murano, M. Parente, Enriched Mu-Calculi Module Checking., LMCS 4 (3) (2008) 1–21. doi:10.2168/LMCS-4(3:1)2008.

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

[24] V. Malvone, F. Mogavero, A. Murano, L. Sorrentino, On the counting of strategies, in: TIME 2015, 2015, pp. 170–179. doi:10.1109/TIME.2015.19.

[25] F. Mogavero, A. Murano, G. Perelli, M. Vardi, What Makes ATL* Decidable? A Decidable Fragment of Strategy Logic., in: CONCUR'12, LNCS 7454, Springer, 2012, pp. 193–208. doi:10.1007/978-3-642-32940-1_15.

[26] J. Gutierrez, P. Harrenstein, M. Wooldridge, Expresiveness and complexity results for strategic reasoning, in: LIPIcs, Vol. 42, 2015.

[27] F. Mogavero, A. Murano, M. Vardi, Relentful Strategic Reasoning in Alternating-Time Temporal Logic., in: LPAR'10, LNAI 6355, Springer, 2010, pp. 371–387.

[28] K. Leyton-Brown, Y. Shoham, Essentials of Game Theory: A Concise, Multidisciplinary Introduction (Synthesis Lectures on Artificial Intelligence and Machine Learning), M&C, 2008. doi:10.2200/S00108ED1V01Y200802AIM003.

[29] F. Mogavero, Logics in Computer Science., Ph.D. thesis, Universitá degli Studi di Napoli Federico II, Napoli, Italy (January 2011).

[30] O. Kupferman, M. Vardi, P. Wolper, An Automata Theoretic Approach to Branching-Time Model Checking., JACM 47 (2) (2000) 312–360. doi:10.1145/333979.333987.

[31] W. Thomas, Automata on infinite objects, Handbook of theoretical computer science, Volume B (1990) 133–191.

[32] E. A. Emerson, C. S. Jutla, Tree automata, mu-calculus and determinacy (extended abstract), in: ASFCS 1991, 1991, pp. 368–377. doi:10.1109/SFCS.1991.185392.

[33] V. Bárány, L. Kaiser, A. M. Rabinovich, Expressing cardinality quantifiers in monadic second-order logic over trees, Fundam. Inform. 100 (1-4) (2010) 1–17. doi:10.3233/FI-2010-260.

[34] D. E. Muller, 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) (1995) 69–107. doi:10.1016/0304-3975(94)00214-4.

[35] M. Osborne, A. Rubinstein, A course in game theory, MIT press, 1994.

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

[37] V. Malvone, A. Murano, L. Sorrentino, Games with additional winning strategies, in: CILC, 2015, CEUR, 2015, pp. 175–180.

[38] M. Ummels, Rational behaviour and strategy construction in infinite multiplayer games, in: FSTTCS, 2006, pp. 212–223. doi:10.1007/11944836_21.

[39] R. Selten, Spieltheoretische behandlung eines oligopolmodells mit nachfragetragheit., Zeitschrift fur die gesamte Staatswissenschaft 121 (1965) 301–324.

[40] T. Brihaye, A. D. C. Lopes, F. Laroussinie, N. Markey, ATL with strategy contexts and bounded memory, in: LFCS 2009, 2009, pp. 92–106. doi:10.1007/978-3-540-92687-0_7.

[41] A. D. C. Lopes, F. Laroussinie, N. Markey, ATL with strategy contexts: Expressiveness and model checking, in: FSTTCS 2010, 2010, pp. 120–132. doi:10.4230/LIPIcs.FSTTCS.2010.120.

[42] R. Alur, S. La Torre, P. Madhusudan, Playing games with boxes and diamonds, in: CONCUR 2003, Vol. 2761 of LNCS 2761, Springer, 2003, pp. 127–141.

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

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

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

[46] P. Čermák, A. Lomuscio, F. Mogavero, A. Murano, MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications., in: CAV'14, LNCS 8559, Springer, 2014, pp. 524–531. doi:10.1007/978-3-319-08867-9_34.

[47] C. Dima, F. Tiplea, Model-checking ATL under Imperfect Information and Perfect Recall Semantics is Undecidable., Tech. rep., arXiv (2011).

[48] W. van der Hoek, J. Meyer, Graded modalities in epistemic logic, in: LFCS'92, Vol. 620, Springer, 1992, pp. 503–514.