Plan Logic

Dylan Bellier
Univ Rennes, IRISA, CNRS, France

Massimo Benerecetti
Università degli Studi di Napoli Federico II, Italy

Fabio Mogavero
Università degli Studi di Napoli Federico II, Italy

Sophie Pinchinat
Univ Rennes, IRISA, CNRS, France

Abstract

When reasoning about games, one is often interested in verifying more intricate strategic properties than the mere existence of a winning strategy for a given coalition. Several languages, among which the very expressive *Strategy Logic* (SL), have been proposed that explicitly quantify over strategies in order to express and verify such properties. However, quantifying over strategies poses serious issues: not only does this lead to a non-elementary model-checking problem, but the classic Tarskian semantics is not fully adequate, both from a conceptual and practical viewpoint, since it does not guarantee the realisability of the strategies involved.

In this paper, we follow a different approach and introduce *Plan Logic* (PL), a logic that takes *plans*, *i.e.*, sequences of actions, as first-class citizens instead of strategies. Since plans are much simpler objects than strategies, it becomes easier to enforce realisability. In this setting, we can recover strategic reasoning by means of a compositional *hyperteams semantics*, inspired by the well-known *team semantics*. We show that the *Conjunctive-Goal* and *Disjunctive-Goal fragments* of SL are captured by PL, with an effective polynomial translation. This result relies on the definition of a suitable game-theoretic semantics for the two fragments. We also investigate the model-checking problem for PL. For the full prenex fragment, the problem is shown to be fixed-parameter-tractable: it is polynomial in the size of the model, when the formula is fixed, and 2-ExpTime-complete in the size of the formula. For the *Conjunctive-Goal* and *Disjunctive-Goal fragments* of PL this result can be improved to match the optimal polynomial complexity in the size of the model, regardless of the size of the formula.

2012 ACM Subject Classification Theory of computation → Logic and verification; Theory of computation → Modal and temporal logics; Theory of computation → Complexity theory and logic

Keywords and phrases Logic for strategic reasoning, Strategy Logic, Realisable strategies, Strategies *vs.* plans, Hyperteam semantics

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2024.9

1 Introduction

When reasoning about games, one is often interested in verifying strategic properties involving the players participating in a game. The simplest such property asks whether one of the players is able to win the game, possibly under specific conditions, regardless of what the other players do. This corresponds to checking whether that player has a winning strategy, namely a set of rules, ideally in the form of a procedure or a function, stipulating how the player must choose its moves in each situation or position during the game in order to achieve the goal corresponding to its winning condition. This has led to the development of a number of logical languages specifically tailored to allow for expressing temporal properties that can predicate, more or less explicitly, over strategies. Notable examples are *Alternating Temporal Logic* (ATL*) [2, 3, 19] and *Strategy Logic* (SL) [7, 18, 8, 16, 17]. In its general form, a

44th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2024).
Editors: Siddharth Barman and Sławomir Lasota; Article No. 9; pp. 9:1–9:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

strategy can be viewed as a function that maps histories, *i.e.*, finite sequences of observables encoding what players have seen up to the current situation in the game, to moves that the player following that strategy has to perform in the current position. In SL notation, for instance, one would express the property that player a can win a game against some other player, say b, by means of the following first-order-like sentence $\exists \textsc{x}.\, \forall \textsc{y}.\, (\mathtt{a}, \textsc{x})(\mathtt{b}, \textsc{y})\psi$, where variables X and Y range over strategies, $(\mathtt{a}, \textsc{x})$ and $(\mathtt{b}, \textsc{y})$ bind each player to a specific strategy, and ψ is an LTL formula encoding a's winning condition. The sentence can be read as follows: there exists a strategy X such that, for all strategies Y, if player a follows X and b follows Y, then the objective ψ is achieved. The separation of strategy quantifications and bindings is a distinctive feature of SL and allows for comparing different strategies for multiple objectives, called goals, each corresponding to a binding of agents and variables followed by an LTL formula. This is what provides the logic with the ability to directly express complex strategic properties, such as, for instance, the existence of Nash equilibria [7, 18].

Ideally, once we know that an objective in a game is achievable, we would like to be able to synthesise the existentially quantified strategies that witness the possibility of achieving that goal. This, in turn, would provide a concrete way to obtain a solution to the game by means of logical reasoning. However, quantification over strategies may lead to situations where a formula can be satisfied only if the witness strategies are built with full knowledge of the strategies of the opponents. What this means is that, in order to synthesise such strategies, one may need to know what the other strategies prescribe in the future or even in counterfactual situations, that is along histories different from the one actually followed. Satisfaction of the sentence $\forall \textsc{y}.\, \exists \textsc{z}.\, (\mathtt{a}, \textsc{z})(\mathtt{b}, \textsc{y})\psi$, for instance, boils down to the existence of a Skolem function f such that the purely universal sentence $\forall \textsc{y}.\, (\mathtt{a}, \mathsf{f}(\textsc{y}))(\mathtt{b}, \textsc{y})\psi$ is satisfied. Function f essentially encodes the mechanism that allows one to build the required witness strategy. However, the input to f is a full-fledged strategy, namely a tree-like object that dictates a response to every history of the game. As a consequence, the response $\mathsf{f}(\sigma)(\varpi)$ of the strategy $\mathsf{f}(\sigma)$ on a given history ϖ may well depend on what the input strategy σ dictates on histories different from ϖ. Such information is, however, usually not available while playing the game, as neither future nor counterfactual situations have been encountered by the players. In many cases, the dependency of the Skolem function on these situations is actually not necessary to satisfy the formula and, in those cases, one can prove that a Skolem function exists that indeed does not. When the only Skolem functions that provide satisfaction of a formula do depend on future or counterfactual situations, we say that the corresponding witness existential strategies are unrealisable. For instance, this is the case for the multi-goal sentence $\forall \textsc{y}.\, \exists \textsc{z}.\, ((\mathtt{a}, \textsc{y})\mathtt{X}\,\mathtt{X}\, p \leftrightarrow (\mathtt{a}, \textsc{z})\mathtt{X}\, p)$, requiring that, for any strategy Y, there exists a response strategy Z which, when followed by the same agent a, ensures in the next step the same literal granted by Y two steps ahead. On structures where all positions have successors for each literal, that sentence can clearly only be satisfied by non-realisable strategies. The phenomenon where a sentence turns out to be satisfied only with unrealisable witnesses is referred to as *non-behavioural satisfaction* in the literature [16, 11]. Not only does this allow for satisfiable sentences for which no concrete and effective mechanisms can be implemented that synthesise the corresponding winning strategies for the game, it has also been shown to be the main source of complexity in strategic reasoning [16, 5], often leading to non-elementary procedures for the decision problems.

Interestingly enough, there are fragments of SL that are not intrinsically affected by the problem. More specifically, it was shown in [16] that every formula in the One-Goal fragment of SL is *behaviourally satisfiable*, meaning that if it is satisfiable then there exists a realisable Skolem function that, along each history, only needs to look at that history to choose the next move. This result was later generalised in [11, 12], where a new semantics for SL, called *timeline semantics*, was proposed and the maximal fragment SL[EG], based on the semantic notion of semi-stability, was identified that enjoys the realisability property. These results try to overcome the problem of non-behavioural satisfaction by identifying well-behaved sets of formulae, for which focusing only on the observations of the current history is enough to decide satisfaction. One may argue that, while reasoning about games, those are the only formulae we are really interested in when we want to figure out how to enforce the objectives.

Based on the above observations, we propose here a logic, called *Plan Logic* (PL), for which, by design, no such problem can arise and which can nonetheless express most of the relevant game-theoretic strategic properties. In addition, the truth of all such properties can be checked in doubly-exponential time at most, with the additional guarantee that any satisfied sentence is realisable in the sense discussed above. This is achieved by taking *plans*, namely infinite sequence of moves, as primary objects for the domain of quantification instead of strategies. Plans are much simpler objects compared to strategies, as they have an intrinsically linear nature. Strategies, by contrast, exhibit a branching nature, as they must take care of a player's behaviours along all possible histories, which, taken all together, form a tree-like structure. In a sense, strategies can be viewed as adaptive plans that may react differently depending on the context and the same strategy can also be used for different goals. Such a strategy would prescribe the same choices for two goals as long as they are indistinguishable to the players, that is as long as the histories along the corresponding plays for the two goals coincide, still allowing for different behaviours when the two goals can be distinguished. This feature is not a native one for plans though. Hence, in order to enforce the same behaviour in indistinguishable contexts, we allow for plans to be tied together by means of specific operators. Essentially, as long as two goals are indistinguishable for the players, two tied plans are required to prescribe the same actions, exactly like a single strategy would do.

The linear nature of plans, on the other hand, makes it much easier to enforce realisability. For one, dependence on counterfactual futures becomes a non-issue, since each plan dictates the moves an agent has to take along a single history and different goals would use distinct, though possibly tied, plans. In order to ensure that the choices of a plan do not depend on the future choices of other plans along the same history, we simply need to impose suitable restrictions on the possible dependencies between the quantified variables at the semantic level. Capturing such constraints requires a semantics able to meaningfully express functional dependencies among quantified variables and, at the same time, retain the determinacy of the logic, meaning that each sentence is either true or false in a given structure. To this end, we employ the *Alternating Hodges' semantics*, a semantics based on *hyperteams*, namely sets of sets of variable assignments, in a similar vein to what has been done for QPTL in [5].

Besides the design of PL and the corresponding compositional hyperteam semantics, our contribution is manifold. We provide a polynomial translation of the *Conjunctive-Goal* and *Disjunctive-Goal fragments* of SL under timeline semantics into PL, whose spirit consists in simulating strategy variables by means of several suitably-tied plan variables. The soundness of this translation (Theorem 7) deeply relies upon the introduction of a game-theoretic semantics for these fragments (Theorem 8), which, to our knowledge, has never been proposed in the literature. The result also shows that each Boolean connective, taken in isolation, exhibits a game-theoretic behaviour. In addition, we study the model-checking problem for PL, taking inspiration from the introduced game-theoretic approach. We prove that, for the *Boolean-Goal fragment* of PL, the problem is 2-EXPTIME-COMPLETE in the length of the formula and fixed-parameter-tractable in the size of the model, once the maximum number of bindings is fixed in the formula (Theorem 14). This is the first result with an elementary complexity of the entire Boolean-Goal fragment of a logic for strategic reasoning, in stark contrast with the tower-complete complexity of Boolean-Goal SL under standard semantics [6]. Incidentally, note that no model-checking procedure exists for Boolean-Goal SL under timeline semantics [11, 12]. By leveraging the similarity between the game-theoretic semantics of the Conjunctive and Disjunctive-Goal fragments of both PL and SL, we improve the model-checking complexity of those fragments to PTIME-COMPLETE in the size of the model (Theorem 16).

In light of all these results, we argue that plans not only appear to be a powerful alternative to strategies, but they may also be preferable in terms of adequacy, as most of the difficulties and annoyances that come into play when dealing with strategies do not affect plans.

2 Preliminaries

We denote by Σ^∞ (*resp.*, Σ^*, Σ^+, Σ^ω) the set of (*resp.*, *finite*, *non-empty finite*, *infinite*) *sequences* w over the alphabet Σ, with *length* $|w| \in \mathbb{N} \cup \{\infty\}$. Given $n < |w|$, the *element at* n of w is denoted by $(w)_n$, while its *prefix up to* n by $(w)_{\leq n}$. Two sequences $w, u \in \Sigma^\infty$ are *equal up to* $n \in \mathbb{N}$, in symbols $w =_{\leq n} u$, if $w = u$ or $n < \mathsf{min}\{|w|, |u|\}$ and $(w)_{\leq n} = (u)_{\leq n}$. This equivalence relation lifts to partial functions $\mathsf{f}, \mathsf{g} \colon \mathrm{Z} \rightharpoonup \mathrm{X}^\infty$ on an arbitrary domain Z as follows: $\mathsf{f} =_{\leq n} \mathsf{g}$ if $\mathsf{dom}(\mathsf{f}) = \mathsf{dom}(\mathsf{g})$ and $\mathsf{f}(z) =_{\leq n} \mathsf{g}(z)$, for all $z \in \mathsf{dom}(\mathsf{f})$.

A *concurrent game structure* (CGS, for short) *w.r.t.* an *a priori* fixed countably-infinite set of *atomic propositions* AP is a structure $\mathfrak{G} \triangleq \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{Ps}, v_I, \delta, \lambda \rangle$, where Ag is a finite non-empty set of *agents*, Ac and Ps are countable non-empty sets of *actions* and *positions*, $v_I \in \mathrm{Ps}$ is an *initial position*, $\delta \colon \mathrm{Ps} \times \mathrm{Ac}^{\mathrm{Ag}} \to \mathrm{Ps}$ is a *transition function* mapping every position $v \in \mathrm{Ps}$ and *action profile* $\vec{c} \in \mathrm{Ac}^{\mathrm{Ag}}$ to a position $\delta(v, \vec{c}) \in \mathrm{Ps}$, and, finally, $\lambda \colon \mathrm{Ps} \to 2^{\mathrm{AP}}$ is a *labelling function* mapping every position $v \in \mathrm{Ps}$ to the finite set of atomic propositions $\lambda(v) \subset_{\mathtt{fin}} \mathrm{AP}$ true at that position. The size of $\mathfrak{G}$ is the number of its positions, *i.e.*, $|\mathfrak{G}| \triangleq |\mathrm{Ps}|$. By abuse of notation, $\delta \subseteq \mathrm{Ps} \times \mathrm{Ps}$ also denotes the *transition relation* between positions such that $(v, u) \in \delta$ *iff* $\delta(v, \vec{c}) = u$, for some $\vec{c} \in \mathrm{Ac}^{\mathrm{Ag}}$. A *path* $\pi \in \mathrm{Pth} \subseteq \mathrm{Ps}^\infty \setminus \{\varepsilon\}$ is a sequence of positions compatible with the transition function and beginning with the initial position, *i.e.*, $(\pi)_0 = v_I$ and $((\pi)_i, (\pi)_{i+1}) \in \delta$, for each $0 \leq i < |\pi| - 1$. The labelling function lifts from positions to paths in the usual way: $\lambda \colon \mathrm{Pth} \to (2^{\mathrm{AP}})^+$. A *history* is a finite path $\varpi \in \mathrm{Hst} \triangleq \mathrm{Pth} \cap \mathrm{Ps}^+$, while a *play* $\pi \in \mathrm{Play} \triangleq \mathrm{Pth} \cap \mathrm{Ps}^\omega$ is an infinite one. A *strategy* is a function $\sigma \in \mathrm{Str} \triangleq \mathrm{Hst} \to \mathrm{Ac}$ mapping every history $\varpi \in \mathrm{Hst}$ to an action $\sigma(\varpi) \in \mathrm{Ac}$. A play $\pi \in \mathrm{Play}$ is *compatible* with a *strategy profile* $\vec{\sigma} \in \mathrm{Str}^{\mathrm{Ag}}$ if, for all $i \in \mathbb{N}$, it holds that $(\pi)_{i+1} = \delta((\pi)_i, \vec{c}_i)$, where $\vec{c}_i \in \mathrm{Ac}^{\mathrm{Ag}}$ is the action profile with $\vec{c}_i(a) = \vec{\sigma}(a)((\pi)_{\leq i})$, for all agents $a \in \mathrm{Ag}$. The function $\mathsf{play} \colon \mathrm{Str}^{\mathrm{Ag}} \to \mathrm{Play}$ assigns to each profile $\vec{\sigma} \in \mathrm{Str}^{\mathrm{Ag}}$ the unique play $\mathsf{play}(\vec{\sigma}) \in \mathrm{Play}$ compatible with $\vec{\sigma}$; we also say that $\vec{\sigma}$ *induces* $\mathsf{play}(\vec{\sigma})$.

3 A Logic of Plans

As opposed to existing logics for strategic reasoning, such as ATL* [2] and SL [7, 18], where the (implicit or explicit) domain of quantification is composed of strategies, quite complex objects, we introduce *Plan Logic* that relies on the much simpler notion of *plan*. Plans are infinite sequences $\rho \in \mathrm{Pln} \triangleq \mathrm{Ac}^\omega$ that describes the course of actions an agent chooses to execute in response to what the other agents already decided to do.

From a syntactic standpoint, Plan Logic bears a strong similarity with SL. In particular, PL extends LTL by allowing (i) to quantify explicitly over plans, (ii) to assign plans to agents by means of a binding mechanism similar to the one of SL that connects agents and plan variables, and (iii) to form bundles of plan variables via *tying operations* that are crucial to correlate different plans as parts of essentially the same strategy in the game model.

Syntax. Throughout this work, we implicitly assume an *a priori* fixed countably-infinite set of *variables* Vr. A *binding* $\flat \in \mathrm{Bn} \triangleq \mathrm{Vr}^{\mathrm{Ag}}$ is a function mapping every agent $a \in \mathrm{Ag}$ to a variable $\flat(a) \in \mathrm{Vr}$, commonly represented as a finite sequence of binding pairs $(a_1, x_1), \ldots, (a_k, x_k)$, where each agent occurs exactly once. By $\mathsf{vr}(\flat) \subset \mathrm{Vr}$ we denote the set of variables occurring in $\flat$ and lift the notation to sets of bindings as the union of the corresponding sets element-wise.

For simplicity, the syntax of the full logic forces formulae to be *flat*, as in the flat fragments of CTL* [9] and ATL* [13], where sentences can be combined in a Boolean way, but cannot be nested. Notice that this flatness constraint comes *w.l.o.g.*, when the model-checking problem is considered, as the latter can always be reduced to reasoning about flat formulae via a relabelling of the underlying structure (see [15, 3], for details).

▶ **Definition 1.** Plan Logic *(*PL*, for short) is the set of formulae built according to the following context-free grammar, where* $\flat \in \mathrm{Bn}$*,* $\psi \in \mathrm{LTL}$*,* $\mathrm{V} \subset_{\mathtt{fin}} \mathrm{Vr}$*, and* $x \in \mathrm{Vr}$*:*

$$\varphi \coloneqq \flat\psi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle \mathrm{V} \rangle\, \varphi \mid [\mathrm{V}]\, \varphi \mid \exists x.\, \varphi \mid \forall x.\, \varphi.$$

We shall denote by $\mathsf{free}(\varphi) \subseteq \mathsf{vr}(\varphi) \subset \mathrm{Vr}$ the sets of free variables and variables occurring in φ. Specifically, $\mathsf{free}(\flat\psi) \triangleq \mathsf{vr}(\flat)$ and $\mathsf{free}(\langle \mathrm{V} \rangle\, \varphi) = \mathsf{free}([\mathrm{V}]\, \varphi) \triangleq \mathrm{V} \cup \mathsf{free}(\varphi)$; all other cases are as usual. A *sentence* φ is a formula without free variables, *i.e.*, $\mathsf{free}(\varphi) = \emptyset$. Similarly, $\mathsf{bnd}(\varphi) \subset \mathrm{Bn}$ denotes the set of bindings occurring in φ.

The binding $\flat$ in a PL *goal* $\flat\psi$ have basically the same interpretation as in SL, namely as the mechanism that associates agents with the content of variables, plans in our case, against which LTL formulae can be evaluated, once the corresponding play is determined. Quantifiers and tying operators, on the other hand, need some explaining in game-theoretic terms. Since we are interested in realisability, we require that the plans we quantify over must be effectively computable, namely that each action chosen at some instant can only depend on the past choices of all the quantified plans. This allows us to view plans as branches of the tree representations of strategies. With this view in mind, the quantifier $\exists x$ (*resp.*, $\forall x$) can be read as "*there exists a realisable plan ...*" (*resp.*, "*for all realisable plans ...*"). Tying operators, instead, are precisely the mechanism that connects plans to strategies in the following sense. Different plan variables denote branches of the same strategy, as long as they provide the same choices for any two bindings that share the same history. The operator $\langle \mathrm{V} \rangle$ (*resp.*, $[\mathrm{V}]$) can then be read as "*the plans in* V *are part of a strategy and ...*" (*resp.*, "*if the plans associated with* V *are part of a strategy then ...*"). Essentially, the two operators filter out sets of plans that cannot be part of the same strategy, because they prescribe different actions for the same history. In a sense, these operators play the role of strategic constructs, implicitly quantifying existentially and universally over strategies via their component plans.

To better understand these intuitions, let us discuss some examples of SL formulae and their corresponding PL equivalents. The simple SL sentence $\Phi_{\mathtt{W}} = \exists \textsc{x}.\, \forall \textsc{y}.\, (a, \textsc{x})(b, \textsc{y})\psi$ states that an agent a can win a two-player game with LTL objective ψ. Specifically, it requires the existence of a strategy $\textsc{x}$ whose induced plays, each one induced by some strategy $\textsc{y}$ of the adversary b, satisfy ψ. This same property would be expressed in PL by the sentence $\varphi_{\mathtt{W}} = \exists x.\, \forall y.\, \langle x \rangle\, [y](a, x)(b, y)\psi$, which states that there exists a realisable plan followed by a that is part of some strategy, *e.g.*, the witness strategy for $\textsc{x}$ of the SL sentence, and ensures the objective, regardless of the realisable plans y that are part of possible strategies $\textsc{y}$ followed by the adversary. Note that the realisability requirement for plans is crucial here, since it means that their actions must be chosen on-the-fly only with knowledge of the past history, in order to mimic the behaviour of strategies.

For a second example, let us consider the property claiming the existence of a strategy for some objective ψ that is not strictly dominated by any other strategy. This is expressed by the SL sentence $\Phi_{\mathtt{NSD}} = \exists \textsc{x}. \forall \textsc{x}'. \exists \textsc{y}. ((a, \textsc{x}')(b, \textsc{y})\psi \rightarrow (a, \textsc{x})(b, \textsc{y})\psi)$. The formula asserts that, for some strategy $\textsc{x}$ and any other strategy $\textsc{x}'$, both for the same agent a, there is at least one strategy $\textsc{y}$ for the opponent such that following $\textsc{x}'$ instead of $\textsc{x}$ would not give a a better outcome. In PL terms, that property is captured by the sentence $\varphi_{\mathtt{NSD}} = \exists x. \forall x'. \exists y_1, y_2. \langle x \rangle [x'] \langle y_1, y_2 \rangle ((a, x')(b, y_1)\psi \rightarrow (a, x)(b, y_2)\psi)$, where we ensure that the two plans y_1 and y_2 are part of the same existentially quantified strategy $\textsc{y}$ for b.

As a final example, consider the existence of a Nash equilibrium for the two agents, a and b, whose objectives are ψ_a and ψ_b, respectively. An SL sentence for this property is $\Phi_{\mathtt{NE}} = \exists \textsc{x}. \exists \textsc{y}. \forall \textsc{z}. (((a, \textsc{z})(b, \textsc{y})\psi_a \rightarrow (a, \textsc{x})(b, \textsc{y})\psi_a) \wedge ((a, \textsc{x})(b, \textsc{z})\psi_b \rightarrow (a, \textsc{x})(b, \textsc{y})\psi_b))$, where $\textsc{x}$ and $\textsc{y}$ represent the equilibrium strategies. The sentence asserts that neither agent can improve by unilaterally deviating from the profile, *i.e.*, by deciding to follow any other strategy $\textsc{z}$ instead of $\textsc{x}$ and $\textsc{y}$. The corresponding PL sentence is

$$\varphi_{\mathtt{NE}} = \exists x_1, x_2. \exists y_1, y_2. \forall z_1, z_2. \langle x_1, x_2 \rangle \langle y_1, y_2 \rangle [z_1, z_2] \begin{pmatrix} ((a, z_1)(b, y_1)\psi_a \rightarrow (a, x_2)(b, y_2)\psi_a) \\ \wedge \\ ((a, x_1)(b, z_2)\psi_b \rightarrow (a, x_2)(b, y_2)\psi_b) \end{pmatrix},$$

where the existential strategies $\textsc{x}$ and $\textsc{y}$ are simulated via the operators $\langle x_1, x_2 \rangle$ and $\langle y_1, y_2 \rangle$ on the pairs of plans x_1, x_2 and y_1, y_2, while the universal strategy $\textsc{z}$ via $[z_1, z_2]$ on z_1, z_2.

The overall intuition underlying the correspondence between SL and PL is that, in order to express a strategic property comprising a given set of different bindings, one really only needs to be able to predicate on a small portion of the strategies involved, namely on a single plan for each binding occurring in the property. This intuition is formally substantiated in Section 4, where a formal translation of some behavioural fragments of SL is provided.

Semantics. The semantics of PL formulae relies on the basic notion of *assignment*, a partial function $\chi \in \mathrm{Asg} \triangleq \mathrm{Vr} \rightharpoonup \mathrm{Pln}$ interpreting variables as plans. We may distinguish assignments defined exactly over some set $\mathrm{V} \subseteq \mathrm{Vr}$, *i.e.*, elements of $\mathrm{Asg}(\mathrm{V}) \triangleq \{\chi \in \mathrm{Asg} \mid \mathsf{dom}(\chi) = \mathrm{V}\}$, and those defined on a superset of V, *i.e.*, elements of $\mathrm{Asg}_{\supseteq}(\mathrm{V}) \triangleq \{\chi \in \mathrm{Asg} \mid \mathrm{V} \subseteq \mathsf{dom}(\chi)\}$. The assignment $\chi[x \mapsto \rho]$ is derived from χ by assigning plan $\rho \in \mathrm{Pln}$ to variable $x \in \mathrm{Vr}$.

To interpret a goal $\flat\psi$ *w.r.t.* an assignment $\chi \in \mathrm{Asg}_{\supseteq}(\mathsf{vr}(\flat))$, one needs to consider the play $\mathsf{play}_{\flat}(\chi)$ that is induced by the *plan profile* $\vec{\rho} \triangleq \chi \circ \flat \in \mathrm{Pln}^{\mathrm{Ag}}$ obtained as the functional composition of χ and $\flat$ and associating a plan in χ with every agent, in accordance with the binding $\flat$. Formally, $\mathsf{play}_{\flat}(\chi)$ is the unique play $\pi \in \mathrm{Play}$ such that $(\pi)_{i+1} = \delta((\pi)_i, \vec{c}_i)$, for all $i \in \mathbb{N}$, where $\vec{c}_i \in \mathrm{Ac}^{\mathrm{Ag}}$ is the action profile associating with each agent $a \in \mathrm{Ag}$ the action stipulated at time i by the plan assigned to a in the plan profile $\vec{\rho}$, *i.e.*, $\vec{c}_i(a) = (\vec{\rho}(a))_i$.

The semantics of the tying operators $\langle \mathrm{V} \rangle$ and $[\mathrm{V}]$ requires some intermediate notions. Two bindings $\flat_1, \flat_2 \in \mathrm{Bn}$ *agree up to* $n \in \mathbb{N}$ *on* an assignment $\chi \in \mathrm{Asg}$ if $\mathsf{play}_{\flat_1}(\widehat{\chi}) =_{\leq n} \mathsf{play}_{\flat_2}(\widehat{\chi})$, for some extension $\chi \subseteq \widehat{\chi} \in \mathrm{Asg}_{\supseteq}(\mathsf{vr}(\{\flat_1, \flat_2\}))$. Intuitively, $\flat_1$ and $\flat_2$ agree up to n on χ if two corresponding plan profiles induce the same history ϖ of length $n + 1$, where n evolution steps have occurred since the initial position. Note that $\flat_1$ and $\flat_2$ agree up to 0 on every assignment, since the initial position is always a common history of length 1. For an assignment $\chi \in \mathrm{Asg}$, two variables $x_1, x_2 \in \mathsf{dom}(\chi)$, and two bindings $\flat_1, \flat_2 \in \mathrm{B}$, with $x_1 \in \mathsf{vr}(\flat_1)$ and $x_2 \in \mathsf{vr}(\flat_2)$, we say that the pair (x_1, x_2) is $(\flat_1, \flat_2)$-*tied in* χ when, for every $n \in \mathbb{N}$, if $\flat_1, \flat_2$ agree up to n on χ then $\chi(x_1) =_{\leq n} \chi(x_2)$. This condition ensures the existence of a strategy σ such that the actions $(\chi(x_1))_n$ and $(\chi(x_2))_n$, at every instant of time n, coincide with the action $\sigma(\varpi)$, for some n-evolution-step history ϖ. We lift the notion to sets of variables $\mathrm{V} \subseteq \mathsf{dom}(\chi)$ and bindings $\mathrm{B} \subseteq \mathrm{Bn}$ as follows: V is B-*tied in* χ if (x_1, x_2) is $(\flat_1, \flat_2)$-tied in χ, for all $x_1, x_2 \in \mathrm{V}$ and $\flat_1, \flat_2 \in \mathrm{B}$, with $x_1 \in \mathsf{vr}(\flat_1)$ and $x_2 \in \mathsf{vr}(\flat_2)$.

A Tarskian semantics for PL, *à la* SL, would be formalised as follows.

▶ **Definition 2.** *For an implicitly given* CGS $\mathfrak{G}$*,* Tarski's semantic relation $\chi \models \varphi$ *for* PL *is inductively defined as follows, for all* PL *formulae* φ *and assignments* $\chi \in \mathrm{Asg}_{\supseteq}(\mathsf{free}(\varphi))$*.*

1. $\chi \models \flat\psi$*, if* $\lambda(\mathsf{play}_\flat(\chi)) \models_{\mathrm{LTL}} \psi$*;*
2. *the semantics of Boolean connectives is defined as usual;*
3. $\chi \models \langle \mathrm{V} \rangle \varphi$*, if* $\chi \models \varphi$ *and* V *is* $\mathsf{bnd}(\varphi)$*-tied in* χ*;*
4. $\chi \models [\mathrm{V}] \varphi$*, if* $\chi \models \varphi$ *when* V *is* $\mathsf{bnd}(\varphi)$*-tied in* χ*;*
5. $\chi \models \exists x. \phi$*, if* $\chi[x \mapsto \rho] \models \phi$*, for some plan* $\rho \in \mathrm{Pln}$*;*
6. $\chi \models \forall x. \phi$*, if* $\chi[x \mapsto \rho] \models \phi$*, for all plans* $\rho \in \mathrm{Pln}$.

The meaning of all conditions above should be self-evident. In particular, Item 3 requires, besides the satisfaction of the formula φ, that the set of variables V be tied in the assignment *w.r.t.* the entire set of bindings $\mathsf{bnd}(\varphi)$ occurring in φ, thus ensuring the existence of a strategy containing the plans associated with V. Item 4 just expresses the dual condition, witnessing the equivalence between $\neg \langle \mathrm{V} \rangle \varphi$ and $[\mathrm{V}] \neg\varphi$.

Despite its simplicity, the treatment of plan quantifiers in this semantics does not correctly capture the effective computability requirement for the plans discussed above. To see why, consider again the non-behaviourally satisfiable SL sentence given in the introduction: $\Phi_{\mathtt{NB}} \triangleq \forall \mathrm{Y}. \exists \mathrm{Z}. ((\mathsf{a}, \mathrm{Y}) \mathtt{X}\, \mathtt{X}\, p \leftrightarrow (\mathsf{a}, \mathrm{Z}) \mathtt{X}\, p)$. The corresponding PL translation, obtained similarly to the previous examples, can be, indeed, shown satisfiable under the Tarskian semantics as follows.

▶ **Example 3.** Consider the sentence $\varphi_{\mathtt{NB}} = \forall y. \exists z. [y] \langle z \rangle ((a, y) \mathtt{X}\mathtt{X}\, p \leftrightarrow (a, z) \mathtt{X}\, p)$ and the single-agent two-action two-position CGS $\mathfrak{G} = \langle \{a\}, \{0, 1\}, \{v_0, v_1\}, v_0, \delta, \lambda \rangle$, where (i) action 0 always leads to v_0 and action 1 always to v_1, regardless of the current position, *i.e.*, $\delta(v_i, \{a \mapsto j\}) = v_j$, and (ii) position v_1 is the only one labelled by p, *i.e.*, $\lambda = \{v_0 \mapsto \emptyset, v_1 \mapsto \{p\}\}$. Being a sentence, we evaluate $\varphi_{\mathtt{NB}}$ against the empty assignment $\varnothing$. By applying Items 6 and 5 of Definition 2, we obtain $\mathfrak{G}, \varnothing \models \varphi_{\mathtt{NB}}$ *iff*, for every plan ρ_y, there exists a plan ρ_z such that $\mathfrak{G}, \{y \mapsto \rho_y, z \mapsto \rho_z\} \models [y] \langle z \rangle ((a, y) \mathtt{X}\, \mathtt{X}\, p \leftrightarrow (a, z) \mathtt{X}\, p)$. Now, by Items 4 and 3, it is immediate to see that the two tying operators $[y]$ and $\langle z \rangle$ do not affect the reasoning, since a singleton set of variables is always trivially tied, no matter which assignment or set of bindings is taken into account. Thus, $\mathfrak{G}, \{y \mapsto \rho_y, z \mapsto \rho_z\} \models [y] \langle z \rangle ((a, y) \mathtt{X}\, \mathtt{X}\, p \leftrightarrow (a, z) \mathtt{X}\, p)$ *iff* $\mathfrak{G}, \{y \mapsto \rho_y, z \mapsto \rho_z\} \models (a, y) \mathtt{X}\, \mathtt{X}\, p \leftrightarrow (a, z) \mathtt{X}\, p$. At this point, one can simply choose $\rho_z \triangleq (\rho_y)_1 \cdot 0^\omega$ to satisfy the formula. Hence, following the naive interpretation of $\varphi_{\mathtt{NB}}$ via Tarski's semantics, it holds that $\mathfrak{G}$ satisfies $\varphi_{\mathtt{NB}}$ in a non-realisable way, since ρ_z requires knowledge of ρ_y one step ahead.

This example clearly shows that a precise formalisation of game-theoretic plan quantifications cannot be achieved by following a first-order Tarskian approach, due to treatment of plans as monolithic entities. To adequately model plans both as realisable objects and linear components of strategies, we are, indeed, faced with a challenge. We need to ensure that, when a plan is chosen by a quantifier, the selection of the action provided by that plan at each time instant can only depend on the choices made by the other plans so far during the play. This means that the choice must be made with knowledge of the past, but no knowledge of the future. Not only does this requirement guarantee the realisability of the plans, which is one of the main concerns of this work, but it also makes plans compatible with strategies, where the choices of actions are functionally dependent only on the histories. To overcome this challenge, we resort to a semantic framework recently proposed in [5] precisely to handle behavioural functional dependencies among quantified variables. *Alternating Hodges' semantics* is a compositional formulation of the interpretation of formulae with a distinctive game-theoretic flavour involving two players: Eloise, who wishes to prove the formula true, and Abelard, who tries to disprove it. The underlying idea is that the interpretations of the free variables of a formula φ correspond to the choices that the two players have made prior to the current stage of evaluation of φ. These possible choices are recorded in a two-level structure, called *hyperteam*, which is a set of sets of assignments or, in team-semantics terminology [20], a set of *teams*. Each level summarises the information about the choices a given player can make in its turns. To evaluate φ, then, one player chooses a team, while the opponent chooses one assignment in that team. We shall use a flag $\alpha \in \{\exists\forall, \forall\exists\}$, called *alternation flag*, to keep track of which player is assigned to which level of choice, together with two corresponding satisfaction relations, $\models^{\exists\forall}$ and $\models^{\forall\exists}$, for the evaluation. If $\alpha = \exists\forall$, Eloise chooses the team, while Abelard chooses one of the contained assignments, which must satisfy φ; if $\alpha = \forall\exists$, the dual reasoning applies. Given a flag $\alpha \in \{\exists\forall, \forall\exists\}$, we denote by $\overline{\alpha}$ the dual flag, *i.e.*, $\overline{\alpha} \in \{\exists\forall, \forall\exists\}$ with $\overline{\alpha} \neq \alpha$. For the sake of space, we refer to [5, 4] for a detailed analysis of this semantic framework, for the proofs of classic model-theoretic properties, *e.g.*, De Morgan laws, the closure under *positive normal form*, and for further discussions and explanations.

First-order quantifiers $\mathsf{Q}x$ are dealt with by means of the notion of *response function*, a refined version of Skolem function, namely a map $\mathsf{F} \in \mathrm{Rsp} \subseteq \mathrm{Asg} \to \mathrm{Pln}$ from assignments to plans such that if $\chi_1 =_{\leq n} \chi_2$ then $\mathsf{F}(\chi_1) =_{\leq n} \mathsf{F}(\chi_2)$, for all $\chi_1, \chi_2 \in \mathrm{Asg}$ and $n \in \mathbb{N}$. Intuitively, at each time instant n, the action $(\mathsf{F}(\chi))_n$ of the chosen plan $\mathsf{F}(\chi)$ only depends on the actions $(\chi(x))_t$ of the plan $\chi(x)$ at the time instants $t \leq n$, for each variable $x \in \mathsf{dom}(\chi)$. This obviously means that $(\mathsf{F}(\chi))_n$ is independent of $(\chi(x))_t$ at any future instant $t > n$. This corresponds precisely to the notion of behavioural functor in [5] and captures the realisability constraint on plans discussed earlier. For an assignment $\chi \in \mathrm{Asg}$ and a variable $x \in \mathrm{Vr}$, the F-*extension with* x of χ is the assignment $\mathsf{ext}(\chi, \mathsf{F}, x) \triangleq \chi[x \mapsto \mathsf{F}(\chi)]$.

Similar to [14, 20], a *team* $\mathrm{X} \in \mathrm{Tm} \triangleq \{\mathrm{X} \subseteq \mathrm{Asg}(\mathrm{V}) \mid \mathrm{V} \subseteq \mathrm{Vr}\}$ is a set of assignments on the same domain. Teams defined over some prescribed $\mathrm{V} \subseteq \mathrm{Vr}$ and those defined at least over V are grouped in $\mathrm{Tm}(\mathrm{V}) \triangleq \{\mathrm{X} \in \mathrm{Tm} \mid \mathrm{X} \subseteq \mathrm{Asg}(\mathrm{V})\}$ and $\mathrm{Tm}_{\supseteq}(\mathrm{V}) \triangleq \left\{\mathrm{X} \in \mathrm{Tm} \,\middle|\, \mathrm{X} \subseteq \mathrm{Asg}_{\supseteq}(\mathrm{V})\right\}$. The set of variables on which all the assignments inside a team $\mathrm{X} \in \mathrm{Tm}$ are defined is denoted by $\mathsf{vr}(\mathrm{X})$. The team $\{\varnothing\}$ only containing the empty assignment $\varnothing$ is called the *trivial team*. A set of variables $\mathrm{V} \subseteq \mathsf{vr}(\mathrm{X})$ is B-*tied in* X, for a set of bindings $\mathrm{B} \subseteq \mathrm{Bn}$, if V is B-tied in every assignment $\chi \in \mathrm{X}$. For a response function $\mathsf{F} \in \mathrm{Rsp}$ and a variable $x \in \mathrm{Vr}$, the notion of F-*extension with* x lifts from assignments to teams as follows: $\mathsf{ext}(\mathrm{X}, \mathsf{F}, x) \triangleq \{\mathsf{ext}(\chi, \mathsf{F}, x) \mid \chi \in \mathrm{X}\}$.

As defined in [5, 4], a *hyperteam* $\mathfrak{X} \in \mathrm{HTm} \triangleq \{\mathfrak{X} \subseteq \mathrm{Tm}(\mathrm{V}) \mid \mathrm{V} \subseteq \mathrm{Vr}\}$ is a set of teams. Hyperteams defined over some given $\mathrm{V} \subseteq \mathrm{Vr}$ and those defined at least over V are grouped in $\mathrm{HTm}(\mathrm{V}) \triangleq \{\mathfrak{X} \in \mathrm{HTm} \mid \mathfrak{X} \subseteq \mathrm{Tm}(\mathrm{V})\}$ and $\mathrm{HTm}_{\supseteq}(\mathrm{V}) \triangleq \{\mathfrak{X} \in \mathrm{HTm} \mid \mathfrak{X} \subseteq \mathrm{Tm}_{\supseteq}(\mathrm{V})\}$. The set of variables shared by all teams inside a hyperteam $\mathfrak{X} \in \mathrm{HTm}$ is denoted by $\mathsf{vr}(\mathfrak{X})$. The hyperteam $\{\{\varnothing\}\}$ comprised only of the trivial team is called the *trivial hyperteam*.

The semantics of PL is based on four operations on hyperteams, that take care of the various logical operators. The *partitioning* $\mathsf{par}(\mathfrak{X}) \triangleq \{(\mathfrak{X}_1, \mathfrak{X}_2) \in 2^{\mathfrak{X}} \times 2^{\mathfrak{X}} \mid \mathfrak{X}_1 \uplus \mathfrak{X}_2 = \mathfrak{X}\}$ handles the Boolean connectives, by reducing the evaluation of the entire formula *w.r.t.* $\mathfrak{X}$ to the evaluation of its Boolean components *w.r.t.* disjoint parts $\mathfrak{X}_1$ and $\mathfrak{X}_2$ of $\mathfrak{X}$. The *filtering* $\mathsf{flt}(\mathfrak{X}, \mathrm{V}, \mathrm{B}) \triangleq \{\mathrm{X} \in \mathfrak{X} \mid \mathrm{V} \text{ is B-tied in } \mathrm{X}\}$ *w.r.t.* the sets of variables $\mathrm{V} \subseteq \mathsf{vr}(\mathrm{X})$ and bindings $\mathrm{B} \subseteq \mathrm{Bn}$ deals with the tying operators, by filtering out of $\mathfrak{X}$ all teams X in which V is not B-tied. The *extension* $\mathsf{ext}(\mathfrak{X}, x) \triangleq \{\mathsf{ext}(\mathrm{X}, \mathsf{F}, x) \mid \mathrm{X} \in \mathfrak{X} \text{ and } \mathsf{F} \in \mathrm{Rsp}\}$ *w.r.t.* the variable $x \in \mathrm{Vr}$ takes care of the first-order quantifiers, by F-extending with x every team X in $\mathfrak{X}$, for all possible response functions F. Finally, the *dualisation* $\overline{\mathfrak{X}}$ swaps the role of the two players in a hyperteam, allowing for connecting the two satisfaction relations and for a symmetric treatment of all PL constructs. The swap is accomplished via the notion of choice function $\Gamma\colon \mathfrak{X} \to \mathrm{Asg}$ over a hyperteam $\mathfrak{X}$, which picks a single assignment from each team: $\mathrm{Chc}(\mathfrak{X}) \triangleq \{\Gamma\colon \mathfrak{X} \to \mathrm{Asg} \mid \Gamma(\mathrm{X}) \in \mathrm{X}, \text{for each } \mathrm{X} \in \mathfrak{X}\}$. Then, the dualisation builds a new hyperteam, whose teams are obtained by gathering all the assignments chosen by one of the choice functions: $\overline{\mathfrak{X}} \triangleq \{\mathsf{img}(\Gamma) \mid \Gamma \in \mathrm{Chc}(\mathfrak{X})\}$. Observe that the trivial hyperteam is self-dual, *i.e.*, $\overline{\{\{\varnothing\}\}} = \{\{\varnothing\}\}$. This approach bears strong similarity with the transformations between DNF and CNF formulae, where a hyperteam can be viewed as a disjunction of conjunctive clauses over assignments, if $\alpha = \forall\exists$, and as a conjunction of disjunctive clauses, if $\alpha = \exists\forall$.

The compositional semantics of PL based on hyperteams can then be defined as follows.

▶ **Definition 4.** *For an implicitly given* CGS $\mathfrak{G}$, Hodges' alternating semantic relation $\mathfrak{X} \models^{\alpha} \varphi$ *for* PL *is inductively defined as follows, for all* PL *formulae* φ*, alternation flags* $\alpha \in \{\exists\forall, \forall\exists\}$*, and hyperteams* $\mathfrak{X} \in \mathrm{HTm}_{\supseteq}(\mathsf{free}(\varphi))$*:*

1. $\mathfrak{X} \models^{\exists\forall} \flat\psi$*, if there exists* $\mathrm{X} \in \mathfrak{X}$ *such that* $\lambda(\mathsf{play}_{\flat}(\chi)) \models_{\mathrm{LTL}} \psi$*, for all* $\chi \in \mathrm{X}$*;*
2. $\mathfrak{X} \models^{\alpha} \neg\varphi$*, if* $\mathfrak{X} \not\models^{\overline{\alpha}} \varphi$*;*
3. $\mathfrak{X} \models^{\exists\forall} \varphi_1 \wedge \varphi_2$*, if* $\mathfrak{X}_1 \models^{\exists\forall} \varphi_1$ *or* $\mathfrak{X}_2 \models^{\exists\forall} \varphi_2$*, for all* $(\mathfrak{X}_1, \mathfrak{X}_2) \in \mathsf{par}(\mathfrak{X})$*;*
4. $\mathfrak{X} \models^{\forall\exists} \varphi_1 \vee \varphi_2$*, if* $\mathfrak{X}_1 \models^{\forall\exists} \varphi_1$ *and* $\mathfrak{X}_2 \models^{\forall\exists} \varphi_2$*, for some* $(\mathfrak{X}_1, \mathfrak{X}_2) \in \mathsf{par}(\mathfrak{X})$*;*
5. $\mathfrak{X} \models^{\exists\forall} \langle \mathrm{V} \rangle\, \varphi$*, if* $\mathsf{flt}(\mathfrak{X}, \mathrm{V}, \mathsf{bnd}(\varphi)) \models^{\exists\forall} \varphi$*;*
6. $\mathfrak{X} \models^{\forall\exists} [\mathrm{V}]\, \varphi$*, if* $\mathsf{flt}(\mathfrak{X}, \mathrm{V}, \mathsf{bnd}(\varphi)) \models^{\forall\exists} \varphi$*;*
7. $\mathfrak{X} \models^{\exists\forall} \exists x.\, \varphi$*, if* $\mathsf{ext}(\mathfrak{X}, x) \models^{\exists\forall} \varphi$*;*
8. $\mathfrak{X} \models^{\forall\exists} \forall x.\, \varphi$*, if* $\mathsf{ext}(\mathfrak{X}, x) \models^{\forall\exists} \varphi$*;*
9. $\mathfrak{X} \models^{\alpha} \varphi$*, if* $\overline{\mathfrak{X}} \models^{\overline{\alpha}} \varphi$*, for all other cases.*

The base case (Item 1) for the goals $\flat\psi$ formalises the intuition for satisfaction relative to the flag $\exists\forall$: there exists a team X in $\mathfrak{X}$, all assignments χ of which induce a play $\mathsf{play}_{\flat}(\chi)$ with a labelling that satisfy the LTL formula ψ. One could equivalently define the semantics for the dual flag $\forall\exists$: for all $\mathrm{X} \in \mathfrak{X}$, it holds that $\lambda(\mathsf{play}_{\flat}(\chi)) \models_{\mathrm{LTL}} \psi$, for some $\chi \in \mathrm{X}$. The choice here is immaterial, thanks to the dualisation rule of Item 9. Negation, in accordance with the game-theoretic interpretation, is dealt with in Item 2 by swapping the players associated with the two internal levels of the hyperteam. The semantics of the Boolean connectives (Items 3 and 4), tying operators (Items 5 and 6), and first-order quantifiers (Items 7 and 8) relies on the first three hyperteam operations discussed above. Finally, the semantics for all the remaining cases reduce, thanks to Item 9, to one of the cases presented, after dualising both the hyperteam and the alternation flag. It is immediate to observe that, for a fixed CGS $\mathfrak{G}$, the truth value of a PL sentence φ, when evaluated *w.r.t.* the trivial hyperteam, does not depend on the specific flag, *i.e.*, $\{\{\varnothing\}\} \models^{\exists\forall} \varphi$ *iff* $\{\{\varnothing\}\} \models^{\forall\exists} \varphi$, due to the self duality of $\{\{\varnothing\}\}$. We shall thus write $\mathfrak{G} \models_{\mathrm{PL}} \varphi$ to assert both $\{\{\varnothing\}\} \models^{\exists\forall} \varphi$ and $\{\{\varnothing\}\} \models^{\forall\exists} \varphi$.

The following result, whose proof is a trivial adaptation of the corresponding one in [5], shows that, when no quantifiers are present, the hyperteam semantics bears a natural correspondence with the Tarskian one.

▶ **Theorem 5.** *For all* PL *quantifier-free formulae* φ *and hyperteams* $\mathfrak{X} \in \mathrm{HTm}_{\supseteq}(\mathsf{free}(\varphi))$*:*

1. $\mathfrak{X} \models^{\exists\forall} \varphi$ iff *there exists* $\mathrm{X} \in \mathfrak{X}$ *such that* $\chi \models \varphi$*, for all* $\chi \in \mathrm{X}$*;*
2. $\mathfrak{X} \models^{\forall\exists} \varphi$ iff*, for all* $\mathrm{X} \in \mathfrak{X}$*, it holds that* $\chi \models \varphi$*, for some* $\chi \in \mathrm{X}$*.*

We can now show that, under the hyperteam semantics, the non-behavioural property reported in the introduction is, as expected, no more satisfiable.

▶ **Example 6.** Consider again the sentence $\varphi_{\mathtt{NB}} = \forall y.\, \exists z.\, [y]\, \langle z \rangle\, ((a, y)\mathtt{X}\,\mathtt{X}\, p \leftrightarrow (a, z)\mathtt{X}\, p)$ and the CGS $\mathfrak{G}$ of Example 3. We want to show that $\mathfrak{G} \not\models_{\mathrm{PL}} \varphi_{\mathtt{NB}}$, meaning that $\varphi_{\mathtt{NB}}$ is not behaviourally satisfiable on $\mathfrak{G}$, *i.e.*, there is no realisable plan for z ensuring a match of the truth values of p at time instants 1 and 2. Since $\mathsf{free}(\varphi_{\mathtt{NB}}) = \emptyset$, we evaluate $\varphi_{\mathtt{NB}}$ against the trivial hyperteam $\{\{\varnothing\}\}$, which, as observed before, implies that the alternation flag is of no consequence. *W.l.o.g.*, we choose $\alpha = \forall\exists$, thus focusing on proving $\{\{\varnothing\}\} \not\models^{\forall\exists} \varphi_{\mathtt{NB}}$.

The rule for the universal quantifier $\forall y$ (Item 8) requires to compute the extension $\mathfrak{X} \triangleq \mathsf{ext}(\{\{\varnothing\}\}, y) = \{\{y\!:\!000^\omega\}, \{y\!:\!010^\omega\}, \{y\!:\!100^\omega\}, \{y\!:\!110^\omega\}, \ldots\}$ of $\{\{\varnothing\}\}$, containing a singleton team for each one of the uncountably many plans to assign to y. This results in

$$\{\{\varnothing\}\} \models^{\forall\exists} \forall y.\, \exists z.\, [y]\, \langle z \rangle\, ((a, y)\mathtt{X}\,\mathtt{X}\, p \leftrightarrow (a, z)\mathtt{X}\, p) \;\; \textit{iff} \;\; \mathfrak{X} \models^{\forall\exists} \exists z.\, [y]\, \langle z \rangle\, ((a, y)\mathtt{X}\,\mathtt{X}\, p \leftrightarrow (a, z)\mathtt{X}\, p).$$

To apply the rule for the existential quantifier $\exists z$ (Item 7), we first need to dualise the hyperteam and switch to the $\exists\forall$ flag (Item 9). Since every team of $\mathfrak{X}$ is a singleton set, there is only one possible choice function for it, thus, the result is

$$\mathfrak{X} \models^{\forall\exists} \exists z.\, [y]\, \langle z \rangle\, ((a, y)\mathtt{X}\,\mathtt{X}\, p \leftrightarrow (a, z)\mathtt{X}\, p) \;\; \textit{iff} \;\; \overline{\mathfrak{X}} \models^{\exists\forall} \exists z.\, [y]\, \langle z \rangle\, ((a, y)\mathtt{X}\,\mathtt{X}\, p \leftrightarrow (a, z)\mathtt{X}\, p),$$

where $\overline{\mathfrak{X}} = \{\{y\!:\!000^\omega, y\!:\!010^\omega, y\!:\!100^\omega, y\!:\!110^\omega, \ldots\}\}$ is the singleton hyperteam composed of the unique team containing all plans for y. The quantifier $\exists z$ and the alternation flag $\exists\forall$ are coherent, so we can proceed extending the hyperteam to obtain $\mathfrak{X}' \triangleq \mathsf{ext}\big(\overline{\mathfrak{X}}, z\big)$. The result is

$$\overline{\mathfrak{X}} \models^{\exists\forall} \exists z.\, [y]\, \langle z \rangle\, ((a, y)\mathtt{X}\,\mathtt{X}\, p \leftrightarrow (a, z)\mathtt{X}\, p) \;\; \textit{iff} \;\; \mathfrak{X}' \models^{\exists\forall} [y]\, \langle z \rangle\, ((a, y)\mathtt{X}\,\mathtt{X}\, p \leftrightarrow (a, z)\mathtt{X}\, p),$$

where $\left\{\left\{\begin{matrix} y\!:\!000^\omega \\ z\!:\!00^\omega \end{matrix}, \begin{matrix} y\!:\!010^\omega \\ z\!:\!00^\omega \end{matrix}, \begin{matrix} y\!:\!100^\omega \\ z\!:\!00^\omega \end{matrix}, \begin{matrix} y\!:\!110^\omega \\ z\!:\!00^\omega \end{matrix}, \ldots\right\}, \left\{\begin{matrix} y\!:\!000^\omega \\ z\!:\!10^\omega \end{matrix}, \begin{matrix} y\!:\!010^\omega \\ z\!:\!10^\omega \end{matrix}, \begin{matrix} y\!:\!100^\omega \\ z\!:\!00^\omega \end{matrix}, \begin{matrix} y\!:\!110^\omega \\ z\!:\!00^\omega \end{matrix}, \ldots\right\}, \ldots\right\}$ is the hyperteam $\mathfrak{X}'$ containing one team $\mathsf{ext}(\mathrm{X}, \mathsf{F}, z)$ for every response function $\mathsf{F} \in \mathrm{Rsp}$, where $\mathrm{X} = \{y : 000^\omega, y : 010^\omega, y : 100^\omega, y : 110^\omega, \ldots\}$ is the unique team in $\overline{\mathfrak{X}}$. For instance, the first team in $\mathfrak{X}'$ is obtained by applying the constant function $\mathsf{F}(\chi) = 0^\omega$, while, for the second one, we use the time-0-flip function $\mathsf{F}(\chi) = (1 - (\chi(y))_0) \cdot 0^\omega$. In general, by the behavioural restriction, the action $(\mathsf{F}(\chi))_0$ may only depend on the action $(\chi(y))_0$. Hence, every team X' of $\mathfrak{X}'$ contains at least one assignment χ such that $(\chi(y))_1 \neq (\chi(z))_0$, which implies that $(\mathsf{play}_{(a,y)}(\chi))_2 \neq (\mathsf{play}_{(a,z)}(\chi))_1$. Therefore, for all $\mathrm{X}' \in \mathfrak{X}'$, it holds that $\chi \not\models [y]\, \langle z \rangle\, ((a, y)\mathtt{X}\,\mathtt{X}\, p \leftrightarrow (a, z)\mathtt{X}\, p)$, for some $\chi \in \mathrm{X}'$. As a consequence of Item 1 of Theorem 5, it holds that $\mathfrak{X}' \not\models^{\exists\forall} [y]\, \langle z \rangle\, ((a, y)\mathtt{X}\,\mathtt{X}\, p \leftrightarrow (a, z)\mathtt{X}\, p)$, which, in turn, means that $\{\{\varnothing\}\} \not\models^{\forall\exists} \varphi_{\mathtt{NB}}$ and, so, $\mathfrak{G} \not\models_{\mathrm{PL}} \varphi_{\mathtt{NB}}$, as expected.

4 Adequacy with Strategy Logic under Timeline Semantics

While the PL semantics – thanks to the tying operators – ensures that the strategies involved are also realisable, we have shown for instance that the SL formula $\forall y.\, \exists z.\, ((a, y)\mathtt{X}\,\mathtt{X}\, p \leftrightarrow (a, z)\mathtt{X}\, p)$ involves strategies that are not. As an immediate consequence, the two logics are not directly comparable. Still, as shown in [16], for the *one-goal fragment* of SL, written SL[1G] here, a formula is satisfiable *iff* it is satisfiable when quantifying only over realisable strategies [16]. Moreover, prenex formulae of SL can be given the so-called *timeline semantics* [11, 12] that enforces realisability of the strategies. This semantics relies on the important notion of *maps* – which are objects very close to Skolem functions.

We relate SL with timeline semantics and PL, by showing that the SL *conjunctive goal* and the *disjunctive goal* fragments, respectively denoted by SL[CG] and SL[DG] (their union is written SL[CG/DG]) can be translated into PL. Due to lack of space, we do not recall here the original timeline semantics of SL, instead we introduce a game-theoretic version (whose correctness is established in Theorem 8), that we use to prove the soundness of this translation. It is worth noting that the SL[CG] fragment encompasses the ATL* extension studied in [10].

4.1 Strategy Logic under Timeline Semantics and Plan Logic

Syntax. The timeline semantics of SL is given for the prenex fragment of the language, in which each formula starts with a *quantifier prefix*, namely a finite sequence $\wp$ of existential $\exists \mathrm{x}$ and universal $\forall \mathrm{x}$ quantifiers, where each variable occurs at most once. The set of variables occurring in a quantifier prefix $\wp$ is $\mathsf{vr}(\wp)$, and we let $\mathsf{vr}_{\exists}(\wp)$ (*resp.* $\mathsf{vr}_{\forall}(\wp)$) be the set of existentially (*resp.* universally) quantified variables. In the rest of this section, we implicitly consider SL under the timeline semantics, and thus every SL formula is in prenex form.

Formulae of the fragments SL[CG] and SL[DG] are, respectively, of the form $\wp \bigwedge_{\flat \in B} \flat \psi_{\flat}$ and $\wp \bigvee_{\flat \in B} \flat \psi_{\flat}$, where $\wp$ is a quantifier prefix, B is a set of bindings, and each $\psi_{\flat}$ is an LTL formula. Observe that the one-goal fragment SL[1G] of SL is contained in the intersection of SL[CG] and SL[DG], which amounts to requiring B to be a singleton set. We may use notation SL[BG] to refer to the SL fragment allowing for arbitrary Boolean combinations of goals.

Translation from SL to PL. The translation for the full SL[BG] fragment involves three steps. First we encode each strategy variable with as many plan variables as there are goals in the formula. These plan variables inherit the same quantifier as the original SL variable in the resulting quantifier prefix. Second, to account for the fact that the corresponding plans must be part of the same strategy, we tie such plan variables together by means of a *tying prefix* of suitable tying operators. Third, we replace the strategy variables occurring in the goals of the matrix, *i.e.* the quantifier free subformula following the prefix, with the corresponding plan variable for that goal. More in detail, let $\Phi = \wp\phi$ be a SL[BG] formula. Each quantifier $\mathsf{Q}\mathrm{x}$ in $\wp$ is transformed into a sequence of quantifiers of the form $\mathsf{Q}x_{\flat}$, one for every $\flat \in \mathsf{bnd}(\Phi)$ with $\mathrm{x} \in \mathsf{vr}(\flat)$. Formally, the quantifier prefix of the translation is $\wp_{\mathsf{SL2PL}}(\Phi) \triangleq ((\mathsf{Q}_{\mathrm{x}} x_{\flat})_{\flat \in B_{\mathrm{x}}})_{\mathrm{x} \in \mathsf{vr}(\wp)}$ with $B_{\mathrm{x}} = \{\flat \in \mathsf{bnd}(\varphi) \mid \mathrm{x} \in \mathsf{vr}(\flat)\}$ and $\mathsf{Q}_{\mathrm{x}} = \exists$ if $\mathrm{x} \in \mathsf{vr}_{\exists}(\wp)$ and $\mathsf{Q}_{\mathrm{x}} = \forall$ if $\mathrm{x} \in \mathsf{vr}_{\forall}(\wp)$.

We now keep track of the fact that the various obtained variables $x_{\flat}$ stem from a single variable x by tying them in a coherent manner via a tying prefix $\tau_{\mathsf{SL2PL}}(\Phi)$: when variable x was existentially (*resp.* universally) quantified, the tying of the $x_{\flat}$'s is existential (*resp.* universal) as follows. Writing $\mathrm{V}_{\mathrm{x}} \triangleq \{x_{\flat} \mid \flat \in B_{\mathrm{x}}\}$ for the set of plan variables associated with the strategy variable x, we let $\tau_{\mathsf{SL2PL}}(\Phi) \triangleq (\langle \mathrm{V}_{\mathrm{x}} \rangle)_{\mathrm{x} \in \mathsf{vr}_{\exists}(\wp)}([\mathrm{V}_{\mathrm{x}}])_{\mathrm{x} \in \mathsf{vr}_{\forall}(\wp)}$. Finally, in each original goal subformula $\flat\psi$, we replace every occurrence of variable x with the new variable $x_{\flat}$. The complete translation of the matrix ϕ (a Boolean combination of goals) is denoted by $\phi_{\mathsf{SL2PL}}(\Phi)$. Gathering all the translation components we have defined, we obtain $\mathsf{SL2PL}(\Phi) \triangleq \wp_{\mathsf{SL2PL}}(\Phi)\, \tau_{\mathsf{SL2PL}}(\Phi)\, \phi_{\mathsf{SL2PL}}(\Phi)$, whose size is polynomial in that of Φ. In the next subsection we show that this translation is sound for both the conjunctive and disjunctive goal fragments of SL. We refer the reader to the examples in Section 3 for instances of this translation.

▶ **Theorem 7.** $\mathfrak{G} \models_{\mathrm{SL}} \Phi$ iff $\mathfrak{G} \models \mathsf{SL2PL}(\Phi)$, *for every* SL[CG/DG] *sentence* Φ *and* CGS $\mathfrak{G}$.

The proof of Theorem 7 is reported in Section 4.2. A crucial step in the proof is the introduction of a game-theoretic semantics for SL, which reduces the evaluation of an SL[CG/DG] sentence Φ in a given CGS $\mathfrak{G}$ to the evaluation of a corresponding PL formula $\mathsf{GTS}_{\mathrm{SL}}(\Phi)$ in a modified CGS $\mathsf{GTS}_{\mathrm{SL}}(\mathfrak{G}, \Phi)$. This construction turns out to be a game-theoretic semantics for the conjunctive and disjunctive goal fragments of SL.

4.2 Game-theoretic Semantics of SL[CG/DG]

The game-theoretic semantics of SL[CG/DG] employs an additional *operator agent*, who plays the role of the single Boolean operator involved in the quantifier-free matrix of the sentence (either $\wedge$ or $\vee$) and can choose the specific goal formula to be falsified/verified. Essentially, the key idea behind the proposed semantics is that, as long as two bindings follow the same play, the operator agent can postpone the decision of which of the corresponding goal formula to falsify/verify.

Given a CGS $\mathfrak{G} = \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{Ps}, v_I, \delta, \lambda \rangle$ and an SL[CG/DG] sentence $\Phi = \wp\phi$, we build the new CGS $\mathsf{GTS}_{\mathrm{SL}}(\mathfrak{G}, \Phi)$ and the new formula $\mathsf{GTS}_{\mathrm{SL}}(\Phi)$ as follows, where $B_\Phi = \mathsf{bnd}(\Phi)$.

▶ **Construction 1.** *In* CGS $\mathsf{GTS}_{\mathrm{SL}}(\mathfrak{G}, \Phi)$*, a position* $\widehat{v} = (v, B)$ *stems from a position* v *in* $\mathfrak{G}$ *equipped with a set* B *of bindings, precisely those that agree so far along the history that led to* $\widehat{v}$*. We set* $\widehat{\mathrm{Ps}} \triangleq \{\widehat{v}_\exists, \widehat{v}_\forall, \widehat{v}_\circledast\} \cup \mathrm{Ps} \times 2^{B_\Phi}$*, where three special sink positions* $\widehat{v}_\exists, \widehat{v}_\forall$ *and* $\widehat{v}_\circledast$ *are explained later. The initial position of* $\mathsf{GTS}_{\mathrm{SL}}(\mathfrak{G}, \Phi)$ *is* $\widehat{v_I} = (v_I, B_\Phi)$*, since at the beginning all the bindings agree on the empty history. The set of agents in* $\mathsf{GTS}_{\mathrm{SL}}(\mathfrak{G}, \Phi)$ *gathers the* variable agents*, one for each variable quantified in* $\wp$*, and the extra operator agent, written* $x_\circledast$*,* i.e. $\widehat{\mathrm{Ag}} \triangleq \mathsf{vr}(\Phi) \cup \{x_\circledast\}$*. The actions of* $\mathsf{GTS}_{\mathrm{SL}}(\mathfrak{G}, \Phi)$ *include all the actions of the original* CGS $\mathfrak{G}$ *and a new* binding action *for each binding occurring in the original formula* Φ*,* i.e. $\widehat{\mathrm{Ac}} \triangleq \mathrm{Ac} \cup B_\Phi$*. The variable agents are only allowed to choose an action from the original* CGS *, while binding actions are reserved to agent* $x_\circledast$*, who can only choose a binding belonging to the decoration of the current position. To force each agent to always pick the right type of action, we use the three sink positions* $\widehat{v}_\exists$*,* $\widehat{v}_\forall$ *and* $\widehat{v}_\circledast$*. Specifically, position* $\widehat{v}_\exists$ *(*resp. $\widehat{v}_\forall$*) is reached every time the agent for a universally (*resp. *existentially) quantified variable mischooses a binding action instead of a proper one. Conversely,* $\widehat{v}_\circledast$ *is reached any time agent* $x_\circledast$ *either mischooses a proper action or takes a binding action outside of the decoration of the current position. Formally, we say that an action profile* $\vec{c} \in (\mathrm{Ac} \cup \mathsf{bnd}(\varphi))^{\mathsf{vr}(\wp) \cup \{x_\circledast\}}$ *is* Q-ill-typed*, for* $\mathsf{Q} \in \{\forall, \exists\}$*, if the leftmost variable* x *in the quantifier prefix* $\wp$ *such that* $\vec{c}(x) \notin \mathrm{Ac}$ *is* Q*-quantified, and that* $\vec{c}$ *is* $\circledast$-ill-typed *in position* $\widehat{v} = (v, B)$ *if* $\vec{c}(x) \notin B$*. An action profile is* well-typed *in position* $\widehat{v}$ *if it is neither* Q*-ill-typed nor* $\circledast$*-ill-typed in position* $\widehat{v}$*. The notion of bindings that agree with the choice of* $x_\circledast$ *is formalized as follows. We say that two bindings* $\flat_1, \flat_2 \in \mathrm{Bn}$ *(whose variables are in* $\mathsf{vr}(\wp)$*) are* indistinguishable at position $v \in \mathrm{Ps}$ w.r.t. action assignment $\vec{c} \in \mathrm{Ac}^{\mathsf{vr}(\wp)}$ *of variable agents, in symbols* $\flat_1 \equiv_v^{\vec{c}} \flat_2$*, whenever* $\delta(v, \vec{c} \circ \flat_1) = \delta(v, \vec{c} \circ \flat_2)$*,* i.e.*, the same position is reached by playing either* $\vec{c} \circ \flat_1$ *or* $\vec{c} \circ \flat_2$*. A move at position* $\widehat{v} = (v, B)$ *with well-typed action profile* $\vec{c}$ *in* $\widehat{v}$ *leads to position* $\widehat{u} = (u, C)$ *where* $u = \delta(v, \vec{c} \circ \flat)$ *for the choice* $\flat = \vec{c}(x_\circledast)$ *of agent* $x_\circledast$*, and* $C \subseteq B$ *retains only the bindings that are indistinguishable from* $\flat$ *(at* v w.r.t. $\vec{c}$*). Formally,*

$$\widehat{\delta}(\widehat{v}, \vec{c}) \triangleq \begin{cases} \widehat{v}_{\mathsf{Q}} & \text{if } \widehat{v} = \widehat{v}_{\mathsf{Q}}, \text{ or } \widehat{v} \neq \widehat{v}_\circledast \text{ and } \vec{c} \text{ is } \overline{\mathsf{Q}}\text{-ill-typed, with } \mathsf{Q} \in \{\exists, \forall\}; \\ \widehat{v}_\circledast & \text{if } \widehat{v} = \widehat{v}_\circledast \text{ or } \vec{c} \text{ is } \circledast\text{-ill-typed in } \widehat{v}; \\ (\delta(v, \vec{c} \circ \flat), \{\flat' \in B \mid \flat' \equiv_v^{\vec{c}} \flat\}) & \text{with } (v, B) = \widehat{v} \text{ and } \flat = \vec{c}(x_\circledast), \text{ otherwise.} \end{cases}$$

Finally, the label of $\widehat{v} = (v, B)$ *inherits from the label of* v *in* $\mathfrak{G}$ *with the extra propositions* $q_\flat$*, one for each binding* $\flat \in B$*. Formally,* $\widehat{\lambda}(\widehat{v}_\exists) \triangleq \{p_\exists\}$*,* $\widehat{\lambda}(\widehat{v}_\forall) \triangleq \{p_\forall\}$*,* $\widehat{\lambda}(\widehat{v}_\circledast) \triangleq \emptyset$*, and* $\widehat{\lambda}((v, B)) \triangleq \lambda(v) \cup \{q_\flat \in \mathrm{AP} \mid \flat \in B\}$*.*

We now turn to the definition of $\mathsf{GTS}_{\mathrm{SL}}(\Phi)$ that is to be evaluated on $\mathsf{GTS}_{\mathrm{SL}}(\mathfrak{G}, \Phi)$. Since in the construction above variables turned into agents, the involved bindings all collapse to the single identity binding $\flat_{\mathrm{id}}$, *i.e.* $\flat_{\mathrm{id}}(x) = x$ for every $x \in \mathsf{vr}(\wp) \cup \{x_{\circledast}\}$. As a consequence, formula $\mathsf{GTS}_{\mathrm{SL}}(\Phi)$ contains only one goal of the form $\flat_{\mathrm{id}}\psi$, where the definition of ψ depends on whether Φ belongs to SL[CG] or to SL[DG]. Here we illustrate the case $\Phi = \wp \bigwedge_{\flat \in \mathrm{B}} \flat \psi_{\flat} \in$ SL[CG], for which we set:

$$\mathsf{GTS}_{\mathrm{SL}}(\wp \bigwedge_{\flat \in \mathrm{B}} \flat \psi_{\flat}) \triangleq \wp \forall x_{\circledast} \flat_{\mathrm{id}}((\mathtt{F}\, p_{\exists}) \vee ((\mathtt{G}\, \neg p_{\forall}) \wedge \bigwedge_{\flat \in \mathrm{B}} ((\mathtt{G}\, q_{\flat}) \rightarrow \psi_{\flat}))).$$

Intuitively, formula $(\mathtt{F}\, p_{\exists}) \vee ((\mathtt{G}\, \neg p_{\forall}) \wedge \bigwedge_{\flat \in \mathrm{B}}((\mathtt{G}\, q_{\flat}) \rightarrow \psi_{\flat}))$ gives the win to the existential agents as soon as a universal variable agent makes an ill-typed decision (this is the disjunct $\mathtt{F}\, p_{\exists}$). Otherwise, for the existential variable agents to win, they should never make an ill-typed decision (see the $\mathtt{G}\, \neg p_{\forall}$ subformula) and should guarantee each $\flat$-objective $\psi_{\flat}$ if the obtained play coincides with the original play, namely the one induced by $\flat$ in the original arena; note that in case operator agent chooses an ill-typed action, no such original play exits.

A dual approach holds for the disjunctive case, that results in setting:

$$\mathsf{GTS}_{\mathrm{SL}}(\wp \bigvee_{\flat \in \mathrm{B}} \flat \psi_{\flat}) \triangleq \wp \exists x_{\circledast} \flat_{\mathrm{id}}((\mathtt{G}\, \neg p_{\forall}) \wedge ((\mathtt{F}\, p_{\exists}) \vee \bigvee_{\flat \in \mathrm{B}} ((\mathtt{G}\, q_{\flat}) \wedge \psi_{\flat}))).$$

The following theorem states that the above constructions provide a proper game-theoretic semantics for SL[CG/DG].

▶ **Theorem 8.** $\mathfrak{G} \models_{\mathrm{SL}} \Phi$ iff $\mathsf{GTS}_{\mathrm{SL}}(\mathfrak{G}, \Phi) \models \mathsf{GTS}_{\mathrm{SL}}(\Phi)$*, for all* SL[CG/DG] *sentences* Φ*.*

We sketch the proof road-map of Theorem 8 that consists in showing both (a) that the truth of an SL[CG] formula entails the truth of its $\mathsf{GTS}_{\mathrm{SL}}$ translation, and (b) that the truth of an SL[DG] formula entails the truth of its $\mathsf{GTS}_{\mathrm{SL}}$ translation. Observe that the if direction of Theorem 8 follows from (the contrapositions of) items (a) and (b), the determinacy of SL, and the duality of the $\mathsf{GTS}_{\mathrm{SL}}$ constructions for SL[CG] and SL[DG]. Recall that one quantifies over strategies in SL and over plans in PL, the target setting of the game-theoretic semantics. According to the hyperteam semantics of PL, quantifications of plan variables is dealt with by means of responses to variable assignments. What one needs to do, then, is design a correspondence between the strategies of the SL sentence and those responses.

▶ **Theorem 9.** $\mathfrak{G} \models \mathsf{SL_2PL}(\Phi)$ iff $\mathsf{GTS}_{\mathrm{SL}}(\mathfrak{G}, \Phi) \models \mathsf{GTS}_{\mathrm{SL}}(\Phi)$*, for all* SL[CG/DG] *sentences* Φ*.*

Similarly to the preceding proof approach, we show that (a) the truth of $\mathsf{SL_2PL}(\Phi)$, where $\Phi \in$ SL[CG], entails the truth of its $\mathsf{GTS}_{\mathrm{SL}}$ translation, and (b) the truth of $\mathsf{SL_2PL}(\Phi)$, where $\Phi \in$ SL[DG], entails the truth of its $\mathsf{GTS}_{\mathrm{SL}}$ translation. Notice that both formulae are in PL, but that formula $\mathsf{SL_2PL}(\Phi)$ is based on duplicates $x_{\flat}$'s of the original variables x in Φ, while in $\mathsf{GTS}_{\mathrm{SL}}(\Phi)$ the original variables of Φ are kept as is, with an extra operator agent variable. Reconstructing a response for x from those of the $x_{\flat}$'s is made possible thanks to the tying operators introduced in the formula $\mathsf{SL_2PL}(\Phi)$.

Theorems 8 and 9 entail Theorem 7.

5 Model Checking of Plan Logic

We finally consider the model-checking problem of four fragments of PL, namely PL[BG], PL[CG], PL[DG], and PL[1G], similar to the ones exhibited for SL. Recall that the model-checking problem of PL (and its fragments) is a decision problem that asks whether an input CGS is a model of an input PL sentence. In [6], it has been shown that, due to the non-behaviouralness, *i.e.*, unrealisability, of the Tarskian semantics of the Boolean-Goal fragment of SL (SL[BG]) [16], its model-checking problem is tower complete in the alternation of quantifiers. We prove instead that, despite its high expressive power, PL[BG] enjoys a problem with a 2ExpTime-complete *formula complexity*, which is not harder than the one for the much simpler logic ATL*. This result is obtained by reducing the evaluation of a PL[BG] sentence φ in a given CGS $\mathfrak{G}$ to the evaluation of a PL[1G] sentence $\widehat{\varphi}$ in a modified structure $\widehat{\mathfrak{G}}$. Also, by tuning the reduction for PL[CG] and PL[DG], we obtain a model-checking procedure with an optimal PTime-complete *model complexity*.

Goal Fragments of PL. The *Boolean-Goal fragment* of PL (PL[BG]) comprises all positive Boolean combinations of formulae (in prenex form) $\wp\,\tau\,\phi$, where $\wp$ is a quantifier prefix, τ a tying prefix, and ϕ a positive Boolean combination of goals $\flat\psi$. The *Conjunctive-Goal fragment* of PL (PL[CG]) (*resp.*, *Disjunctive-Goal fragment* of PL (PL[DG])) further restricts PL[BG] by requiring ϕ to be a conjunction (resp, disjunction) of goals. Finally, in the *One-Goal fragment* of PL (PL[1G]), ϕ is assumed to be a single goal $\flat\psi$.

The encoding $\varphi_{\mathtt{NE}}$ of the existence of a Nash equilibrium discussed in Section 3 is an example of PL[BG] formula, as well as the sentence $\varphi_{\mathtt{NB}}$ of Example 3. The sentence $\varphi_{\mathtt{W}}$ stating the existence of a winning strategy in a two player game clearly belongs to PL[1G], while the existence of a non strictly-dominated strategy can be expressed in PL[DG], as witnessed by the encoding $\varphi_{\mathtt{NSD}}$. By turning $\neg\varphi_{\mathtt{NSD}}$ into positive normal form, we obtain the following PL[CG] sentence:

$$\forall x.\, \exists x'.\, \forall y_1, y_2.\, [x]\, \langle x'\rangle\, [y_1, y_2]\, ((a, x')(b, y_1)\neg\psi \wedge (a, x)(b, y_2)\neg\psi)\,.$$

In [1], it has been shown that Nash equilibria can actually be expressed in SL[CG]. Thus, the corresponding translations into PL would result in sentences of the PL[CG] fragment. Indeed, the conversion function $\mathsf{SL_2PL}\colon \mathrm{SL} \to \mathrm{PL}$, when applied to an SL[CG/DG] sentence, necessarily returns a PL[CG/DG] one. Finally, $\mathsf{GTS}_{\mathrm{SL}}\colon \mathrm{SL} \to \mathrm{PL}$ always produces a PL[1G] sentence.

The One-Goal Fragment. A simple inspection of the syntactic translation $\mathsf{SL_2PL}\colon \mathrm{SL} \to \mathrm{PL}$ of the previous section shows that its application to an SL[1G] sentence results in a PL[1G] one with the same quantifier prefix, the same goal, and an eliminable prefix of tying operators on a single variable. Actually, a more general elimination property can be proven for arbitrary tying operators in a PL[1G] formula $\wp\tau\flat\psi$: (a) if τ contains $\langle \mathrm{V}\rangle$, with two variables $x, y \in \mathrm{V}$, where y is universally quantified after x in $\wp$, then the subformula originating in $\langle \mathrm{V}\rangle$ is equivalent to $\bot$; (b) dually, if τ contains $[\mathrm{V}]$, with two variables $x, y \in \mathrm{V}$, where y is existentially quantified after x in $\wp$, then the subformula originating in $[\mathrm{V}]$ is equivalent to $\top$; (c) in all other cases, the tying operator can be eliminated, by replacing all the variables in V with the first one of V quantified in $\wp$. *E.g.*, assuming $\wp = \forall x \exists y \forall z$ and $\flat = (a, x)(b, y)(c, y)$, we have that (i) $\wp\,[x, y]\langle y, z\rangle\,\flat\psi \equiv \top$, (ii) $\wp\,\langle y, z\rangle[x, y]\,\flat\psi \equiv \bot$, and (iii) $\wp\,\langle x, y\rangle[y, z]\,\flat\psi \equiv \forall x.\,(a, x)(b, x)(c, x)\psi$. Thus, the following tying-elimination property holds true.

▶ **Proposition 10.** *Every sentence* $\wp\tau\flat\psi$ *in* PL[1G] *has an equivalent sentence of the form* $\wp'\flat'\psi$.

By combining this proposition with Theorem 7, we obtain that the One-Goal fragments of SL and PL semantically coincide.

▶ **Theorem 11.** *For every* SL[1G] *sentence* Φ*, there is a* PL[1G] *sentence* φ *and,* vice versa, *for every* PL[1G] *sentence* φ*, there is an* SL[1G] *sentence* Φ *such that:* $|\Phi| = \Theta(|\varphi|)$ *and* $\mathfrak{G} \models_{\mathrm{SL}} \Phi$ iff $\mathfrak{G} \models_{\mathrm{PL}} \varphi$*.*

Due to the known $2^{2^{\mathcal{O}(|\varphi|)}} \cdot |\mathfrak{G}|^{\mathcal{O}(1)}$ complexity of the model-checking problem of SL[1G] [16, Theorem 5.14], we can immediately derive the following theorem.

▶ **Theorem 12.** PL[1G] *model-checking problem is* 2-ExpTime-complete($|\varphi|$) *in the length of the specification* φ *and* PTime-complete($|\mathfrak{G}|$) *in the size of the model* $\mathfrak{G}$*.*

The Boolean-Goal Fragment. The encoding underlying Theorem 8 of the game-theoretic semantics for SL[CG/DG] into PL[1G] allowed us to prove the equivalence between these logics and the corresponding fragments of PL. We shall leverage the same idea here to solve the model-checking problem of PL[BG]. Given a CGS $\mathfrak{G}$ and a sentence $\varphi = \wp\tau\phi$, with binding set $\mathrm{B}_\varphi \triangleq \mathsf{bnd}(\varphi)$, we build a new CGS $\mathsf{GTS}_{\mathrm{BG}}(\mathfrak{G}, \varphi)$, whose plays are bundles of plays from $\mathfrak{G}$, one per binding in ϕ. This is done, intuitively, by composing in parallel as many copies of $\mathfrak{G}$ as there are bindings in ϕ, resulting in positions that correspond to vectors $\widehat{v} \in \mathrm{Ps}^{\mathrm{B}_\varphi}$ of original positions of $\mathfrak{G}$. Agents of the new game coincide with the variables quantified in $\wp$, while actions carries over unchanged. A move from a position $\widehat{v}$ to a position $\widehat{u}$ is then a vector of parallel moves in $\mathfrak{G}$, one per original position contained in $\widehat{v}$, while forbidding incoherent concurrent choices *w.r.t.* the tying operators occurring in τ. Formally, an action assignment $\vec{c} \in \mathrm{Ac}^{\mathsf{vr}(\wp)}$ is V*-incoherent at* $\widehat{v}$ *w.r.t.* φ, where $\mathrm{V} \subset \mathrm{Vr}$, if there are two variables $x_1, x_2 \in \mathrm{V}$ and two bindings $\flat_1, \flat_2 \in \mathrm{B}_\varphi$, with $x_1 \in \mathsf{vr}(\flat_1)$ and $x_2 \in \mathsf{vr}(\flat_2)$, such that $\widehat{v}(\flat_1) = \widehat{v}(\flat_2)$, but $\vec{c}(x_1) \neq \vec{c}(x_2)$. In other words, a concurrent move $\vec{c}$ is V-incoherent at $\widehat{v}$ *w.r.t.* φ, if there are variables in V whose different associated actions in $\vec{c}$ should have been equal, being part of bindings that are indistinguishable at $\widehat{v}$. We say that $\vec{c}$ is $\exists$*-incoherent* (*resp.*, $\forall$*-incoherent*) *at* $\widehat{v}$ *w.r.t.* φ if the leftmost set of variables V in τ, such that $\vec{c}$ is V-incoherent at $\widehat{v}$ *w.r.t.* φ, occurs in a tying operator of type $\langle V \rangle$ (*resp.*, $[V]$). Intuitively, $\vec{c}$ is $\exists/\forall$-incoherent at $\widehat{v}$ *w.r.t.* φ if the first violated tying constraint specified in τ is existential/universal. If $\vec{c}$ is neither $\exists$-incoherent nor $\forall$-incoherent at $\widehat{v}$ *w.r.t.* φ, we say that $\vec{c}$ is *coherent* at $\widehat{v}$ *w.r.t.* φ.

▶ **Construction 2.** *Given a* CGS $\mathfrak{G} = \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{Ps}, v_I, \delta, \lambda \rangle$ *and a* PL[BG] *sentence* φ*, with binding set* $\mathrm{B}_\varphi \triangleq \mathsf{bnd}(\varphi)$*, let* $\mathsf{GTS}_{\mathrm{BG}}(\mathfrak{G}, \varphi) \triangleq \langle \widehat{\mathrm{Ag}}, \widehat{\mathrm{Ac}}, \widehat{\mathrm{Ps}}, \widehat{v_I}, \widehat{\delta}, \widehat{\lambda} \rangle$ *be the* CGS *obtained as follows: (a) agents are the variables quantified in* φ, i.e., $\widehat{\mathrm{Ag}} \triangleq \mathsf{vr}(\varphi)$*; (b)* $\widehat{\mathrm{Ac}} \triangleq \mathrm{Ac}$*; (c) positions are* B_φ*-indexed vectors of original positions from* $\mathfrak{G}$*, plus two distinguished sink positions* $\widehat{v}_\exists$ *and* $\widehat{v}_\forall$, i.e., $\widehat{\mathrm{Ps}} \triangleq \{\widehat{v}_\exists, \widehat{v}_\forall\} \cup \mathrm{Ps}^{\mathrm{B}_\varphi}$*; (d) the initial position is the* v_I*-constant vector,* i.e., $\widehat{v_I} \triangleq \{\flat \in \mathrm{B}_\varphi \mapsto v_I\}$*; (e) every position, but the distinguished ones, are labelled with a set of fresh atomic propositions, one per binding and original labelling,* i.e., $\widehat{\lambda}(\widehat{v}_\exists) \triangleq \{p_\exists\}$*,* $\widehat{\lambda}(\widehat{v}_\forall) \triangleq \{p_\forall\}$*, and* $\widehat{\lambda}(\widehat{v}) \triangleq \{p_\flat \in \mathrm{AP} \mid \flat \in \mathrm{B}_\varphi, p \in \lambda(\widehat{v}(\flat))\}$*; (f) the transition function* $\widehat{\delta}$ *maps every position* $\widehat{v} \in \widehat{\mathrm{Ps}} \setminus \{\widehat{v}_\exists, \widehat{v}_\forall\}$ *and action profile* $\vec{c} \in \mathrm{Ac}^{\mathsf{vr}(\varphi)}$ *coherent at* $\widehat{v}$ w.r.t. φ *to position* $\widehat{u} \in \widehat{\mathrm{Ps}} \setminus \{\widehat{v}_\exists, \widehat{v}_\forall\}$*, where, for each binding* $\flat \in \mathrm{B}_\varphi$*, the position* $\widehat{u}(\flat)$ *is the successor of* $\widehat{v}(\flat)$ *in* $\mathfrak{G}$ *following the action profile* $\vec{c} \circ \flat$*, which associates with each agent* $a \in \mathrm{Ag}$ *the action stipulated by* $\vec{c}$ *for the variable* $\flat(a)$*; formally,*

$$\widehat{\delta}(\widehat{v}, \vec{c}) \triangleq \begin{cases} \widehat{v}_{\mathtt{Q}}, & \text{if } \widehat{v} = \widehat{v}_{\mathtt{Q}} \text{ or } \vec{c} \text{ is } \overline{\mathtt{Q}}\text{-incoherent at } \widehat{v} \text{ w.r.t. } \varphi, \text{ with } \mathtt{Q} \in \{\exists, \forall\}; \\ \widehat{u}, & \text{otherwise, where } \widehat{u}(\flat) \triangleq \delta(\widehat{v}(\flat), \vec{c} \circ \flat), \text{ for all } \flat \in \mathrm{B}_\varphi. \end{cases}$$

The PL[1G] encoding of the game-theoretic semantics for the PL[BG] sentence $\varphi = \wp\tau\phi$ is relatively easy to formalise at this point: besides verifying the coherence constraints dictated by the tying prefix τ, we only need to check that the bundles of plays induced by plans in the CGS $\mathsf{GTS}_{\mathrm{BG}}(\mathfrak{G}, \varphi)$ satisfy the matrix ϕ. Checking the constraints amounts to requiring avoidance of the two distinguished sink positions $\widehat{v}_{\exists}$ and $\widehat{v}_{\forall}$. The verification of the matrix is obtained by transforming ϕ into the LTL formula $\widehat{\phi}$, where each goal $\flat\psi$ is replaced by the LTL formula $\widehat{\psi}$, in turn obtained by replacing in ψ every atomic proposition p with $p_\flat$, *i.e.*, $\widehat{\phi} \triangleq \phi\left[\flat\psi/\widehat{\psi}\right]$, with $\widehat{\psi} \triangleq \psi\,[p/p_\flat]$. Altogether, we get the following:

$$\mathsf{GTS}_{\mathrm{BG}}(\wp\tau\phi) \triangleq \wp\,\flat_{\mathrm{id}}\left((\mathtt{F}\,p_\exists) \vee \left((\mathtt{G}\,\neg p_\forall) \wedge \widehat{\phi}\right)\right).$$

Since the original PL[BG] sentence φ and its PL[1G] translation $\mathsf{GTS}_{\mathrm{BG}}(\varphi)$ share the same quantifier prefix $\wp$, thanks to Theorem 5, we can prove the correctness of the encoding, on the basis that $\chi \models \tau\phi$ *iff* $\chi \models \flat_{\mathrm{id}}((\mathtt{F}\,p_\exists) \vee ((\mathtt{G}\,\neg p_\forall) \rightarrow \widehat{\phi}))$, for all $\chi \in \mathrm{Asg}_{\supseteq}(\mathsf{vr}(\wp))$, which can be done by structural induction on $\tau\phi$ (using the simple semantic rules of Definition 2).

▶ **Theorem 13.** $\mathfrak{G} \models \varphi$ iff $\mathsf{GTS}_{\mathrm{BG}}(\mathfrak{G}, \varphi) \models \mathsf{GTS}_{\mathrm{BG}}(\varphi)$*, for every* PL[BG] *sentence* φ*.*

Once we observe that $|\mathsf{GTS}_{\mathrm{BG}}(\mathfrak{G}, \varphi)| = 2 + |\mathfrak{G}|^{|\mathsf{bnd}(\varphi)|}$ and $|\mathsf{GTS}_{\mathrm{BG}}(\varphi)| = \mathrm{O}(|\varphi|)$, thanks to Theorem 12, we can derive the following result, where FPT means *fixed-parameter tractable*.

▶ **Theorem 14.** *The model-checking problem for* PL[BG] *is* 2-ExpTime-complete($|\varphi|$) *in the length of the specification* φ *and* $\mathrm{FPT}_{|\varphi|}(|\mathfrak{G}|)$ *in the size of the model* $\mathfrak{G}$*, with the length of the specification* φ *as parameter, once the maximum number of bindings is fixed.*

The Conjunctive & Disjunctive Goal Fragments. The simpler conjunctive/disjunctive nature of goal combinations in PL[CG/DG] allows us to considerably improve on the model complexity of the model-checking problem of PL[BG], by removing redundant information from the position space, which is necessary only to handle arbitrary Boolean combinations. This is done by suitably merging ideas from Constructions 1 and 2: from the former we inherit the structure topology, while of the latter we use the criteria for determining the compliance of the choices *w.r.t.* the tying operators (compliance issues are irrelevant in Construction 1, since strategies are considered). We end-up with an *ad hoc* game-theoretic semantics for PL[CG/DG], whose resulting CGS encoding $\mathsf{GTS}_{\mathrm{CDG}}(\mathfrak{G}, \varphi)$ is virtually identical to Construction 1, where the notion of well-typed action assignment is generalised to take into account the coherence constraints introduced for Construction 2. The sentence encoding $\mathsf{GTS}_{\mathrm{CDG}}(\varphi)$ is also identical to the one used for SL[CG/DG] in association with Construction 1.

The following theorem can be obtained as a slight adaptation of the proof of Theorem 9.

▶ **Theorem 15.** $\mathfrak{G} \models \varphi$ iff $\mathsf{GTS}_{\mathrm{CDG}}(\mathfrak{G}, \varphi) \models \mathsf{GTS}_{\mathrm{CDG}}(\varphi)$*, for every* PL[CG/DG] *sentence* φ*.*

Once we observe that $|\mathsf{GTS}_{\mathrm{CDG}}(\mathfrak{G}, \varphi)| = 2 + 2^{|\mathsf{bnd}(\varphi)|} \cdot |\mathfrak{G}|$ and $|\mathsf{GTS}_{\mathrm{CDG}}(\varphi)| = \mathrm{O}(|\varphi|)$, we can derive the following result, again thanks to Theorem 12.

▶ **Theorem 16.** *The model-checking problem for* PL[CG/DG] *is* 2-ExpTime-complete($|\varphi|$) *in the length of the specification* φ *and* PTime-complete($|\mathfrak{G}|$) *in the size of the model* $\mathfrak{G}$*.*

6 Conclusion

We have introduced *Plan Logic* as a language for strategic reasoning, alternative to Strategy Logic, based on the notion of *plans* instead of strategies. We show that this conceptual shift is quite beneficial, as the intrinsic linear nature of plans allows for a semantics that guarantees realisability of the satisfiable sentences via *behavioural functional constraints*. To this end, we propose *hyperteams* as a novel semantic framework for strategic reasoning, which enjoys several important model-theoretic properties, *e.g.*, *compositionality* and *determinacy*. Observe that, for instance, the only semantics for SL that tackle the problem is the timeline semantics proposed in [11, 12], which, however, exhibits neither compositionality nor determinacy. The authors of [12] show, indeed, that such a semantics is not adequate already when applied to SL[BG], as it is undetermined on sentences of that fragment. It is worth noting that the hyperteam semantics, unlike the timeline one based on an *ad hoc* Skolem semantics, is a principled approach that has been applied to model very general functional dependencies among variables in other logics, such as QPTL [5] and FOL [4].

We showed that, thanks to the behavioural nature of the semantics, the model-checking problem of PL[BG] is still 2-EXPTIME-COMPLETE, in stark contrast with the non-elementarity of the same problem for SL[BG] [6]. This further highlights the importance of enforcing behavioural constraints. In addition, we study the *conjunctive and disjunctive goal fragments* of PL in direct comparison with the respective fragments of SL. We show their expressive equivalence, by introducing a novel game-theoretic semantics that allows for a direct comparison between the two logics. Thanks to the connection between the game-theoretic semantics of the PL[CG] and SL[CG], on the one hand, and of PL[DG] and SL[DG], on the other, we can improve the model-checking complexity of those fragments to PTIME-COMPLETE in the size of the model (Theorem 16). Note that these fragments strictly include ATL*, a prominent logic in strategic reasoning, which, in turn, is subsumed by the one-goal fragment of both SL and PL. These fragments are quite interesting, as they enable several forms of complex strategic reasoning, such as strategy domination and various forms of equilibria (*e.g.*, Nash equilibria).

References

1. E. Acar, M. Benerecetti, and F. Mogavero. Satisfiability in Strategy Logic Can Be Easier than Model Checking. In *AAAI'19*, pages 2638–2645. AAAI Press, 2019. doi:10.1609/AAAI.V33I01.33012638.
2. R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-Time Temporal Logic. In *FOCS'97*, pages 100–109. IEEECS, 1997.
3. R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-Time Temporal Logic. *JACM*, 49(5):672–713, 2002. doi:10.1145/585265.585270.
4. D. Bellier, M. Benerecetti, D. Della Monica, and F. Mogavero. Alternating (In)Dependence-Friendly Logic. *APAL*, 174(10):103315:1–58, 2023.
5. D. Bellier, M. Benerecetti, D. Della Monica, and F. Mogavero. Good-for-Game QPTL: An Alternating Hodges Semantics. *TOCL*, 24(1):4:1–57, 2023. doi:10.1145/3565365.
6. P. Bouyer, P. Gardy, and N. Markey. Weighted Strategy Logic with Boolean Goals Over One-Counter Games. In *FSTTCS'15*, LIPIcs 45, pages 69–83. Leibniz-Zentrum fuer Informatik, 2015.
7. K. Chatterjee, T.A. Henzinger, and N. Piterman. Strategy Logic. In *CONCUR'07*, LNCS 4703, pages 59–73. Springer, 2007. doi:10.1007/978-3-540-74407-8_5.
8. K. Chatterjee, T.A. Henzinger, and N. Piterman. Strategy Logic. *IC*, 208(6):677–693, 2010. doi:10.1016/J.IC.2009.07.004.
9. D. Dams. Flat Fragments of CTL and CTL*: Separating the Expressive and Distinguishing Powers. *LJIGPL*, 7(1):55–78, 1999. doi:10.1093/JIGPAL/7.1.55.
10. S. Enqvist and V. Goranko. The Temporal Logic of Coalitional Goal Assignments in Concurrent Multiplayer Games. *TOCL*, 23(4):21:1–58, 2022. doi:10.1145/3517128.
11. P. Gardy, P. Bouyer, and N. Markey. Dependences in Strategy Logic. In *STACS'18*, LIPIcs 96, pages 34:1–15. Leibniz-Zentrum fuer Informatik, 2018.
12. P. Gardy, P. Bouyer, and N. Markey. Dependences in Strategy Logic. *TCS*, 64(3):467–507, 2020. doi:10.1007/S00224-019-09926-Y.

13 V. Goranko and S. Vester. Optimal Decision Procedures for Satisfiability in Fragments of Alternating-time Temporal Logics. In *AIML'14*, pages 234–253. College Publications, 2014. URL: http://www.aiml.net/volumes/volume10/Goranko-Vester.pdf.

14 W. Hodges. Compositional Semantics for a Language of Imperfect Information. *LJIGPL*, 5(4):539–563, 1997. doi:10.1093/JIGPAL/5.4.539.

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

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

17 F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. Reasoning About Strategies: On the Satisfiability Problem. *LMCS*, 13(1:9):1–37, 2017.

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

19 S. Schewe. ATL* Satisfiability is 2ExpTime-Complete. In *ICALP'08*, LNCS 5126, pages 373–385. Springer, 2008. doi:10.1007/978-3-540-70583-3_31.

20 J.A. Väänänen. *Dependence Logic: A New Approach to Independence Friendly Logic*, volume 70 of *London Mathematical Society Student Texts*. CUP, 2007.