



HAL
open science

Strategic reasoning with dependencies: hyperteam logics, realizable strategies, dependency matrices

Dylan Bellier

► **To cite this version:**

Dylan Bellier. Strategic reasoning with dependencies: hyperteam logics, realizable strategies, dependency matrices. Logic in Computer Science [cs.LO]. Université de Rennes, 2024. English. NNT: 2024URENE006 . tel-04959292

HAL Id: tel-04959292

<https://theses.hal.science/tel-04959292v1>

Submitted on 20 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT DE

L'ÉCOLE NORMALE SUPÉRIEURE DE RENNES

ÉCOLE DOCTORALE N° 601

*Mathématiques, Télécommunications, Informatique, Signal,
Systèmes, Électronique*
Spécialité : *Informatique*

Par

Dylan BELLIER

« **Strategic Reasoning with Dependencies :
Hyperteam Logics, Realizable Strategies, Dependency Matrices** »

Thèse présentée et soutenue à Rennes, le 4 Novembre 2024

Unité de recherche : IRISA

Rapporteurs avant soutenance :

Fan YANG Assistant professor at Department of Philosophy and Religious Studies, Utrecht University
Martin ZIMMERMANN Associate Professor at Aalborg University, Denmark

Composition du Jury :

Président :	Patricia Bouyer	Dr. de recherche, LMF, Univ. Paris-Saclay, CNRS, ENS Paris-Saclay
Examineurs :	Erich GRÄDEL	Professor, Mathematische Grundlagen der Informatik, RWTH Aachen
	Patricia Bouyer	Dr. de recherche, LMF, Univ. Paris-Saclay, CNRS, ENS Paris-Saclay
	Natasha Alechina	Associate Pr., Information and computing sciences, Utrecht University
	Juha Kontinen	Professor, University Helsinki
Dir. de thèse :	Sophie PINCHINAT	Professeur à l'Université de Rennes
Co-dir. de thèse :	François SCHWARZENTRUBER	Maitre de Conférences à l'ENS Rennes

ACKNOWLEDGEMENT

Je voudrais tout d'abord remercier mes superviseurs, Sophie Pinchinat et François Schwarzentruher. Pendant toute la durée de ma thèse, Sophie a été présente pour me prodiguer des conseils et pour pointer du doigt mes lacunes, ce qui m'a poussé à être exigeant envers moi-même. Je lui suis extrêmement reconnaissant de tout ce qu'elle a fait pour moi, qu'il s'agisse de m'accompagner dans mes premières reviews, de m'aider dans les différentes démarches administratives ou de s'intéresser au sujet des *hyperteams*, qui n'était pas le sien originellement. C'est un plaisir de travailler avec quelqu'un d'aussi riche intellectuellement et j'espère pouvoir continuer à collaborer avec elle aussi longtemps que possible. François n'est pas en reste et a constamment été une présence agréable et rassurante. Savoir qu'il serait toujours à l'écoute et compréhensif fut un garde-fou inestimable dans les moments de doutes qui sont inévitables quand on entreprend une thèse. Chacun de mes deux directeurs a complété l'autre afin de me fournir un environnement stimulant et source de profondeur de réflexion.

Il me faut ensuite remercier Fabio Mogavero et Massimo Benerecetti, avec qui j'ai collaboré tout au long de ma thèse. J'ai eu le privilège de passer en tout six mois à Naples en leur agréable compagnie pour travailler sur l'idée des *hyperteam*. Ma thèse ne serait pas ce qu'elle est aujourd'hui sans eux. Je tiens à mentionner que l'idée d'un ensemble d'ensembles d'assignements vient de Dario Della Monica que je remercie également.

Mes pensées vont ensuite évidemment à ma mère, que j'aime plus que tout. Merci d'avoir été là pour m'encourager lors des moments durs et pour me féliciter lors de mes réussites. J'ai de la chance d'avoir une mère aussi géniale.

Je veux aussi remercier ma chérie, Gaëlle, d'avoir rendu ma dernière année de thèse si paisible. Ta présence à mes côtés me donne toujours plus de force et de motivation pour faire ce qui doit être fait et donner le meilleur de moi-même.

Merci aussi à tous mes amis, Rémy et Solveig pour ces moments passés à la Dordogne, Quentin et Ariane pour ces sessions d'escalade et de fromage fondu, Paul pour les grandes discussions de maths, Larouci et Gaëtan pour la pierre et le caillou et bien sûr, Léon, Juliette, Pierre Jean, Manon, Charlotte, Lucie, Jean et Æddis pour tous les supers moments passés ensemble.

TABLE OF CONTENTS

Introduction	9
1 Preliminaries	23
1.1 Recall on First-Order Logic	23
1.1.1 Syntax	24
1.1.2 First-Order Structure	25
1.1.3 Semantics	25
1.1.4 Limitations of Tarski’s-like quantification	27
1.2 Recall on Team Semantics	27
1.2.1 Representing dependences with Teams	28
1.2.2 Dependence/Independence Logic	29
1.2.3 Examples	31
1.2.4 Limitations of Team Semantics	32
1.3 Conclusion	32
2 A hyperteams semantics	37
2.1 A two-level framework: the Hyperteam	37
2.2 A new semantical relation	44
2.3 Fundamentals	45
2.4 Adequacy	49
2.5 Skolemisation with hyperteams	51
2.6 Conclusion	55
2.A Proofs of Section 2.1	56
2.B Proofs of Section 2.2	57
2.C Proofs of Section 2.4	62
2.D Proofs of Section 2.5	66
3 Alternating Dependence/Independence-Friendly Logic	81
3.1 A symmetrical extension of DIF	81
3.1.1 Syntax.	82

TABLE OF CONTENTS

3.1.2	Semantics	85
3.1.3	Examples	85
3.1.4	Properties	90
3.1.5	Meta semantics	92
3.2	Comparisons to other logics	96
3.2.1	First-order Adequacy	96
3.2.2	Dependence/Independence-Friendly Logic Adequacy	96
3.2.3	Second-Order & Team Logics	100
3.3	Game-Theoretic Semantics	102
3.4	Conclusion	109
3.A	Proofs of Section 3.1	111
3.B	Proofs of Section 3.2	113
3.C	Proofs of Section 3.3	119
4	Good-For-Game Quantified Propositional Temporal Logic	125
4.1	Recalls on temporal logics	126
4.1.1	Linear-time Temporal Logic	127
4.1.2	Quantified Propositional Temporal Logic	128
4.2	Dynamic Reasoning	129
4.2.1	Realizable responses	129
4.2.2	Examples	134
4.3	Game-Theoretic Semantics	139
4.3.1	Quantification Game for Sentences	139
4.3.2	Quantification Game for Formulas	147
4.3.3	Complexity Results	150
4.4	Conclusion	151
4.A	Proofs of Section 4.2	152
4.B	Proofs of Section 4.3	153
5	Plan Logic	183
5.1	Strategy Logic	183
5.1.1	Concurrent Game Structures.	184
5.1.2	Syntax.	184
5.1.3	Semantics.	185
5.1.4	Timeline semantics.	185

5.2	Strategic Reasoning	186
5.2.1	Syntax.	186
5.2.2	Examples.	187
5.2.3	Semantics.	189
5.3	Adequacy with Strategy Logic under Timeline Semantics	193
5.3.1	Strategy Logic under Timeline Semantics and Plan Logic	193
5.3.2	Game-theoretic Semantics of SL[C/DG]	195
5.4	Decision Problems	197
5.4.1	Goal Fragments of PL.	198
5.4.2	One-goal Fragment	198
5.4.3	Boolean-Goal Fragment	199
5.4.4	Collusion-free Fragment	201
5.5	Conclusion	202
5.A	Missing Proofs of Section 5.3	204
5.A.1	Meta-PL	204
5.A.2	Skolem semantics for Meta-PL	206
5.A.3	Proof of Theorem 5.4	207
5.A.4	Realizable meta map	212
5.A.5	Proof of Theorem 5.3	215
5.B	Missing Details of Section 5.4	222
6	Dependency Matrices	225
6.1	A Multiplayer Delay Game	227
6.1.1	Definition	227
6.1.2	Strategies for Dependency Matrices	229
6.1.3	Undecidability of Resolution	236
6.2	Resolving Dependency Matrices	237
6.2.1	Definition and Properties	237
6.2.2	A Parity Game to solve EWS	239
6.2.3	Reduction from the Church Synthesis problem	242
6.2.4	Perfect-Information Matrices with Possibly Infinite Values	243
6.3	Conclusion	244
6.A	Proofs of Section 6.1	245
6.B	Proofs of Section 6.2	247

TABLE OF CONTENTS

6.C Reduction from Church matrices to Round Robin matrices	250
6.D Proof of Theorem 6.5	251
Conclusion	253
Bibliography	257

INTRODUCTION

Verification and Synthesis

The usage of formal methods. The ubiquity of complex systems naturally raises many challenges regarding their place in our modern world. Many critical systems rely on computer science to achieve their tasks, such as planes, rockets, power plants, and many more [Kni02; MK20]. Ensuring the proper behavior of these systems is one of the major issues for today's researchers. For instance, some systems have to deal with sensitive data that should not be accessible to an external observer. In that case, the behavior of the system should respect the property of *opacity* [Maz04]; that is, its behavior should be indistinguishable from the observer. There are multiple options to *verify* that a system is well-behaved. The first is to design *unit tests* to see how the system reacts to various inputs. With this approach, one has to manually design and run each individual test to check whether the system responds appropriately. This allows for fine control over the cost of verification as the number of test is extensible making unit test relevant for simple systems. However, this approach, by definition, cannot be exhaustive and might miss some problematic corner cases in more complex systems. Another option is to use *formal methods* [Bee+24] to try to *prove* or *disprove* the property in the system. This approach requires creating a *formal model* of the system, namely a mathematical abstraction that denotes the features of its behavior relevant to the property to be verified. This approach also requires expressing the property formally: it can be a mathematical statement finely tailored for the system model (for instance, opacity), or it can be expressed using a *logical language* that allows for the systematic verification of a large class of similar properties.

With the formal methods approach naturally comes the synthesis of systems *correct by design*: once a property is properly stated, one can compute a model of that property *ex nihilo* that drafts a system automatically satisfying the property. For instance, given a non-opaque system, one may want to synthesize a *controller* (i.e. a companion of the system that restricts its behavior) for this system that ensures opacity while still allowing the system to accomplish its task.

The main focuses in verification are three specific problems. The first is the *model*

checking problem [BK08; DGL16b]: it takes as input a system model and a formal property, usually expressed in logic, and outputs whether the model satisfies the property or not. The second problem is the *satisfiability problem* [BHM09]: it takes as input a formal property and outputs whether there is a model that satisfies this property or not. In this case, because only the property is given as input, the answer might depend on assumptions about the model: whether infinite models, infinitely branching models, cyclic models, or more are eligible may change the output. Restricting such classes of models is an important consideration when it comes to designing procedures to solve the satisfiability problem. Finally, the last problem is the *synthesis problem* [KVBSV13; Fin16]: it also takes as input a formal property and outputs a model that satisfies this property. Here, the model that satisfies the property must be effectively computed, as opposed to the satisfiability problem, which means that additional metrics are needed to evaluate the efficiency of algorithms that solve the synthesis problem: for example, efficient algorithms may output unnecessarily large models.

Which model to use? The choice of a model is always a trade-off between precision and the cost of verification. On one hand, a coarser model leads to better complexity results for each problem presented above. On the other hand, a richer model provides a better approximation of the system, allowing the verification procedure's outcomes to be transferred to the system more reliably. Of course, the more complex the system, the more intricate the properties we want to verify and then, the richer the model needs to be. Thus, ultimately, the choice of model depends on the type of system to represent.

In this thesis, we are particularly interested in *complex systems*, whose behavior may be non-deterministic and yield infinite computation. Such systems can be either *closed* or *open*. In both cases, what mainly matters is the ability to formalize the system behavior and its executions by providing it with an *operational semantics*. The most abstract model that is classically used in formal methods is the one of *transition system* (see for instance [BK08; DGL16a]): this model is a graph whose vertices are the states of the system and whose edges are called (state-)transitions. Both states and transitions can be labeled to represent respectively atomic facts (i.e. *propositions*) for state observations and relevant events (i.e. *actions*). Transition systems are equipped with a *semantics* for their behavior that can vary depending on the verification purpose. For instance, the behavior can be the set of all *infinite traces* — a trace is a single sequence of observations in the transition system — or by the *computation tree* — the infinite unfolding of the transition system

that correlates all traces as infinite branches.

What makes a system *closed* comes from its ability to execute on its own, independently of the context, such as a program that computes the decimal of pi. Among closed system one may find systems made of multiple interacting components. When focusing on analyzing a single component, it seems natural to abstract the remaining components as an unspecified *environment*. This yields the notion of *open* system, namely, a system whose behavior consists in interacting with the environment (the others components).

Models for *open* systems are many. The transition systems can be used to describe the behavior of the global system by composing its components (including the environment). The resulting transition system is an abstraction of how the components actually interact. Properties such as safety ones can be verified: one often prunes the computation tree as to fix a particular environment scenario, and wants to know if the property holds whichever pruning is considered; this is called *module checking* [KVW01]. However, it might be necessary to refine the analysis if the property fails: indeed, the considered model may under-specify for instance the parallel composition mechanism of the components, so that one is rather interested in knowing whether or not the inherent non-determinism in the model can be controlled to achieve the property; this answers the above-mentioned synthesis problem. A pioneer work is [PR89]. The methodology can naturally be deployed in a game setting, where the components and the environment become players, and where control-like issues are rephrased in game-theoretic terms, namely the existence and even the synthesis of *strategies* to ensure the property of interest. This approach is nowadays widely adopted in formal methods, with the pioneer work of [AHK02].

To go even further and tackle *multi-agent systems* [Sin94], one uses *multiplayer games*: players have their own objective and can be gathered into *coalitions* in order to state various properties. For instance, consider a fleet of drones scanning for rescue during a disaster. Multiple factors must be considered: each drone has its own input on the situation and some autonomy, thus each is represented as a player; the victims can also be modeled as other players with a different set of actions. In this situation, many properties must be verified such as “each drone has to stay connected to the fleet” (individual objective), and “the scan performed by the fleet should be exhaustive” (collective objective). We see that, from the perspective of one drone, one cannot assume that the other drones will have antagonistic behavior, but still have to account for some variability to cover unexpected contingencies.

Which logic to use? The *First-Order Logic* (FO for short) [HA28] is the logic of reference. FO *formulas* are composed of *variable quantifiers* and *predicates* and *relations* organized in *Boolean combinations*. The truth value of an FO formula is evaluated on a given FO structure with an assignment of the free variables of the input formula, i.e., a function from variables to the domain of the FO structure. Closeness between FO formulas and database queries are well studied. Even though the satisfiability problem for full FO is UNDECIDABLE, it is extensively studied to exhibit decidable fragments. On the contrary, the model checking problem for FO is PSPACE-*complete*. Meanwhile, when it comes to dynamic systems, a new range of candidates arises.

On the one hand, *Linear-time Temporal Logic* (LTL for short) [Pnu77] is suited to express trace properties of transition systems, such as *liveness* or *safety* properties, by introducing temporal operators — X for "at the next time..." and U for "...holds until...". On the other hand, *Computation Tree Logic* (CTL for short) [EC82] allows for more detailed exploration of the unfolding of the model, with the ability to quantify over branches within the formula. For instance, the formula $\forall X\psi$ states that property ψ holds in every successor of the computation tree root, and the formula $\exists\psi U\varphi$ states that there is a branch of the computation tree along which property ψ holds until φ becomes true. The two logics LTL and CTL are incomparable but are subsumed by CTL* [EH86]. All these logics are *propositional*, meaning that the first-class citizens are the *propositional variables* (alternatively called *atomic proposition*): an assignment maps each propositional variables to either true or false at each time point of the *timeline*. We can then add propositional quantifiers to LTL and get *Quantified Propositional Temporal Logic* (QPTL for short) [SVW87], which is able to express any ω -regular language. Model checking temporal logics can be costly: the LTL model checking is PSPACE-*complete* while for QPTL, the model checking is NON ELEM, which means it is k -EXPTIME-*hard* for every $k \in \mathbb{N}$. Temporal logics are suited to express properties of closed systems but when it comes to open ones, *strategic reasoning*, i.e. the ability to manipulate strategies, is mandatory.

A foundational logic for strategic reasoning is *Alternating-time Temporal Logic* (ATL for short) [AHK02], which is an extension of CTL where branch quantifiers are replaced with *strategic modalities*: \exists or \forall become $\langle\langle\Gamma\rangle\rangle$, where Γ is a coalition of players, meaning "Players of Γ have a joint strategy to satisfy..." — recall that a player strategy is a function that, given a *history* (a finite sequence of adjacent states), outputs the next decision for the player. ATL has proven useful for expressing proof obligations in system verification, as well as for expressing subroutines of verification algorithms. However, as remarked in

[CHP10], this powerful language is unable to compare strategies, because strategies are only involved in the modalities, an ability that seems necessary to express some properties of interest, such as the existence of a *Nash equilibrium* in a game [ABM19].

For this reason, Chatterjee, Henzinger, and Piterman introduced *Strategy Logic* (SL for short) [CHP10] that was later refined by Mogavero, Murano, Perelli and Vardi [MMPV14; MMPV17] as a first-order-like logic where variables are meant to range over strategies. In SL, a specific operator, called the *binding*, is responsible for assigning a strategy to each player. Once every player is given a strategy, a unique play is defined in the game model. The atoms of SL are LTL formulas that are to be evaluated on the play defined by the players' strategic assignment. Thanks to the new binding operator and because strategies become first-class citizens, SL can state the existence of a Nash equilibrium, but its model checking problem is NON ELEM, and its satisfiability problem is UNDECIDABLE. Fragments of the logic have been studied to reduce the complexity of those problems: when formulas are in prenex form — recall that a formula is in prenex form when it consists of a sequence of quantifiers, that we call the *quantifier prefix*, followed by a quantifier-free subformula — with no Boolean connectors before a binding operator, defining the so-called one-goal fragment of SL (SL[1G] for short), the model checking and satisfiability problems complexity drops to 2-EXPTIME-*complete* [MMPV14; MMPV17]. However, this fragment lacks expressiveness to specify the existence of a Nash equilibrium.

Realizable Strategies in SL. Besides the high complexity of SL, the logic faces an issue regarding quantifier alternation. When considering a quantifier string, say $\forall x \exists y$, the strategy quantified by y *depends* on the entire strategy quantified by x , including future choices and counterfactual plays. Because of this dependence, it can be impossible to automatically generate the strategies involved in the semantics of an SL formula. Thus, the truth of a formula is not a satisfying answer to the verification problems. To alleviate this problem, one might restrict the semantics of quantifier to only *realizable strategies*, i.e. strategies that are chosen independently from the future or counterfactual decisions of already quantified ones. Notice that realizability is then a dependence constraint as it does not change the domain of quantification but only the way of choosing a value for a quantified variable. Actually, the high complexity of SL and the dependency problem of non-realizable strategies seem to go hand in hand: indeed, it has been shown that SL[1G] can be artificially restricted to realizable strategies without affecting the truth values of formulas [MMPV14].

Building on this result, Gardy, Bouyer, and Markey proposed the *timeline semantics* for SL [GBM20], that enforces realizability of quantified strategies by means of restricted *Skolem-like functions*. The resulting logic has a 2-EXPTIME-*complete* model checking procedure for a vast fragment of SL, that can express Nash equilibria. Even though the result is valuable, the proposed semantics is not *compositional* [SH01] and is *ad hoc* for realizability. In this thesis, we propose a new logical framework to express strategy dependencies in strategic reasoning that is adapted to enforce realizability while providing a compositional semantics. For this purpose, we turn to logics that take dependencies into account.

Expressing Dependencies

The study of dependency in logic has come a long way. To provide better intuition on the phenomenon, we will now resort to an alternative approach for the semantics of quantifiers: the *game-theoretic semantics* of J. Hintikka and G. Sandu [HS97]. Quantification can be understood as a game between two players — the *existential player*, that we call Eloise, and the *universal player*, that we call Abelard. This game, called the *evaluation game*, or sometime the *model-checking game*, proceeds on a formula (which we assume is in prenex form for simplicity), following the order given by the quantifier prefix, by making Eloise choose a value for each existentially quantified variable and Abelard choose a value for the others. If the resulting assignment satisfies the quantifier-free subformula, Eloise wins; otherwise, Abelard wins. Even if, the evaluation game is typically defined for arbitrary input formulas, since quantification is the main focus of our discussion, we restrict ourselves to the simpler case of prenex form. Game-theoretic semantics is equivalent to the original semantics of the logic because Eloise has a winning strategy if, and only if, the formula is true, and Abelard has a winning strategy if, and only if, the formula is false.

Dependencies in FO The first appearance of dependency traces back to the definition of *Skolem functions* [Bus98a] for FO: given a closed formula in prenex form $\varphi := \forall x_1 \forall x_2 \dots \forall x_n \exists y \phi$, a Skolem function f for y is an n -ary function that computes the value of y depending on the values of the x_i 's. A Skolem function can be understood as a part of Eloise's strategy for the game-theoretic semantics of the formula φ . The Skolem function reveals the *functional dependency* of the variable y with respect

to x_i 's. Symmetrically to Skolem functions, one finds *Herbrand functions* for universally quantified variables. With the Herbrand/Skolem functions, we see that, in FO, the dependency between variables are completely defined by the order of variable occurrences in the quantifier prefix; the dependence relation $xR_{\text{DEF}}y$, meaning " x depends functionally on y ", is a linear order on the variables.

The fact that Skolem functions formalize those dependencies was the reason for Gardy's approach to define a realizable SL, i.e., to define the semantics for SL in terms of modified Skolem functions. Indeed, Gardy's timeline semantics require the Skolem function to output only realizable strategies. By definition, this approach cannot lead to compositional semantics, as the whole quantifier prefix is required to define the appropriate Skolem functions. We then turn to the different ways of expressing dependencies that have been developed.

Henkin matrices. The first attempt at providing dependencies between variables that do not form a linear order was introduced by Henkin with Henkin matrices [Hen61]. A Henkin matrix is essentially a partial order of the quantified variables. For instance, the matrix $\begin{pmatrix} \forall x_1 & \exists y_1 \\ \forall x_2 & \exists y_2 \end{pmatrix}$ expresses that y_1 depends on x_1 but not on x_2 , and, at the same time, that y_2 depends on x_2 but not on x_1 . Such a specification cannot be expressed in FO. Unfortunately, this would be insufficient for restricting quantification in SL to only realizable strategies, as such a matrix still forces monolithic dependencies between variables.

Team semantics. Later, an pioneer idea was introduced by Hodges [Hod97a; Hod97b] to define a compositional semantics for the expression of dependencies: instead of using an assignment to store the value of already quantified variables, Hodges proposed a set of assignments that he called *trumps*, though the terminology *team* later prevailed. Intuitively, a team represents all the possible scenarios for the universally quantified variables and the corresponding choices made for the existentially quantified ones. Compositional semantics for *Dependence Friendly Logic* and *Independence Friendly Logic* (DFL and IFL, respectively) [HA28; MSS11], two logics for reasoning with *imperfect information*, emerged from this idea. DFL and IFL are built as extensions of FO with the new quantifiers $\exists^{+W}y$ for DFL and $\exists^{-W}y$ for IFL, where W is a set of variables. These quantifiers convey information about the dependence ($+W$) or independence ($-W$) of the quantified variable y . These constraints are enforced by the semantic rule for those quantifiers: for

a team X to satisfy $\exists^{+W}y \varphi$ (and respectively $\exists^{-W}y \varphi$), each assignment χ of X is *extended* with a value for y such that any two *equivalent* assignments χ_1 and χ_2 in X , that is two assignments that are equal on W (respectively outside W), are extended with the same value for y . In this work, we call the function that, given an assignment, chooses the corresponding value for y , a *response function*, or, sometimes, simply *response*. The semantic rule boils down to the existence of a *uniform* response, i.e., a response that outputs the same value on two equivalent assignments. Notice that the definition scales for any equivalence relation. Constraining the response functions is the cornerstone to define a compositional semantics for SL with realizable strategies. Unfortunately, team semantics cannot be used as-is due to some inherent limitations of the approach. First, only existential quantifiers can be constrained: DFL or IFL can be dualized to allow for constraints on universal quantifiers, but then, existential quantifiers are the ones unconstrained, so only one kind of quantifier can be restricted at a time. Second, related to the first point, is that negation is not handled by team semantics. Indeed, the natural rule for negation, which states that X satisfies $\neg\varphi$ if X does not satisfy φ , is unsatisfactory because it would lack the desirable properties of De Morgan's laws (that SL has). For instance, consider the formula $\forall x \exists^{+\emptyset}y (x = y)$. As long as the domain of the model has at least two elements, this formula does not hold since y has to be assigned the same value for two distinct, yet equivalent, assignments of x . However, so does the syntactic negation $\exists x \forall^{+\emptyset}y (x \neq y)$. This is due to the imperfect information resulting from the dependencies, and thus, the evaluation game of DFL and IFL can be undetermined.

Team semantics dependence atoms. To circumvent these limitations, Jouko Väänänen [Vää07] proposed moving the dependence specification from quantifiers to the new atomic proposition, called the *independence atom* [GV13]: $=(\vec{x}, y)$, that means " y depends (only) on variables \vec{x} ." This idea led to the definition of *Dependence Logic*, which can handle negation thanks to its two dual semantics. However, this change confine the dependencies to monolithic ones which makes the logic does not help define a semantics for SL with realizable strategies.

Team semantics for temporal logics. Because SL is built on top of LTL, the interaction between team semantics is to be considered. Adapting LTL for team semantics leads to the definition of TEAMLTL [KMVZ17], which is suited to express *hyperproperties* [CS10], meaning properties of subsets of traces of a transition system rather than

properties of individual traces. For instance, opacity is a hyperproperty because stating that a single trace is opaque does not make sense. Indeed, recall that opacity is achieved when, for every sensitive trace, there is another non-sensitive trace indistinguishable from the first one, which clearly requires a well-defined set of traces. Although there are no quantifiers in LTL the U (until) operator of LTL carries implicit quantification, recall that $\psi U \varphi$ the *existence* of a time point t such that ψ holds at each point up to $t - 1$, while φ holds at t . In the context of a team, this quantification have various interpretation: the question is, "Can t vary depending on the assignment (asynchronous) in the team, or should all assignments choose the same t (synchronous)?" Such considerations must be kept in mind when designing semantics for temporal logics, and thus, for strategic reasoning.

Contribution

We can see that teams are a powerful tool for dealing with dependency constraints and seem well-suited for managing realizability in a compositional SL semantics. However, their intrinsic limitations prevent a symmetric treatment of quantifiers. Therefore, we need to draw inspiration from the strengths of team semantics to design a new framework for expressing dependencies that can overcome this treatment problem.

Hyperteams. If we aim at treating quantifiers in a symmetrical manner, we need to design a framework that can express both viewpoints simultaneously. For this, we propose a two-level structure, the *hyperteam*, that is a team of teams, i.e., a set of sets of assignments. In this way, we can keep track of the imperfect information one player of the evaluation game has about the imperfect information of the other. A hyperteam can be interpreted as a game between an existential player and a universal player, for whom we keep the names of the evaluation game, namely Eloise and Abelard, respectively. Given a hyperteam, there are two dual ways of playing: either Eloise first chooses a team within the hyperteam and then Abelard selects an assignment from that team (that gives the interpretation written $\exists\forall$), or the reverse, where Abelard chooses the team and then Eloise chooses the assignment (that gives the interpretation written $\forall\exists$). Discarding any dependency constraint, a hyperteam satisfies a formula if the assignment resulting from this small game satisfies the formula, leading to the definition of two different semantics, one for each interpretation (either $\exists\forall$ or $\forall\exists$).

Among the many benefits of using a two-level structure is the ability to correlate the two interpretations $\exists\forall$ and $\forall\exists$ by *dualizing* the hyperteam: a procedure similar to translating a formula from conjunctive normal form to disjunctive normal form, that transforms a hyperteam \mathfrak{X} into its dual $\overline{\mathfrak{X}}$ such that, \mathfrak{X} satisfies a formula φ under one interpretation implies $\overline{\mathfrak{X}}$ satisfies φ under the dual interpretation. Thus, the two semantics are closely related. The semantics of quantifiers take inspiration from those for teams: to extend a hyperteam, we extend each team within the hyperteam with every possible response function. The distinction between existential and universal quantifiers lies in the interpretation of the hyperteam. Since choosing the value for the quantified variable amounts to choosing a response function for this variable, existential quantification is defined by extending a hyperteam under the $\exists\forall$ interpretation, while universal quantification requires the hyperteam to be under the $\forall\exists$ interpretation.

Incorporating dependencies into the setting becomes “straightforward”: one can constrain response functions that are used to extend a hyperteam to be uniform. As a result, one obtains two dual compositional semantics that, together, treat quantifiers in a symmetrical manner since both existential and universal quantifiers fall under this uniformity constrain. This approach allows for many kind of constraints yielding a wide range of dependencies. It is important to note that we do not introduce any specific language here; the concept of hyperteam semantics can be applied to a variety of languages, thereby defining many different logics. As we will see, this formalism is well-suited to define the *game-theoretic semantics* of these logics. This type of semantics facilitates the design of model-checking procedures for the considered logics.

Logics with hyperteam semantics. For a start, we exploit the hyperteam semantics for a simple first-order language with monolithic dependencies, generalizing the syntax of DFL and IFL by allowing for negation. The resulting logic, *Alternating Dependence/Independence-Friendly logic* (ADIF for short), allows for expressing properties *about* imperfect information. First, we show that ADIF generalizes of FO, DFL, and IFL in the sense that a closed formula in one of those logics is true if and only if it is true in ADIF. Second, we prove that ADIF is equivalent to the powerful *Second-Order Logic* [Chu56; Sha91], an extension of FO that allows for quantification over relations, by interpreting a quantified relation as a first order variable with suited dependence: it uses the idea of casting a n -ary relation into a $n + 1$ -ary function where the added dimension is used to identify each tuple of the relation. Thus, a quantified n -ary relation can trans-

lated into a quantified variable that takes the role of the added dimension, and a $+W$ dependence with $|W| = n$.

Next, since our goal is to address strategic reasoning, we define hyperteam semantics for temporal logics. Keeping in mind the objective of enforcing realizability, we constrain quantified variable to be independent from the future as it is the main issue in making strategies realizable. We choose to apply this idea to QPTL whose quantification ranges over infinite sequences of binary choices. To enforce independence from the future, we propose a new kind of equivalence between assignments: we say that assignments χ_1 and χ_2 *agree up to t* if for each variable x , we have χ_x that is equal up to t to $\chi_2(x)$. The response functions should output the same choice at time t for two assignments that agree up to time t , and yield the logic *Good-for-game QPTL* (GFG-QPTL for short). We show that GFG-QPTL and QPTL are as expressive, although the semantics of formulas differs: because of this independence from the future, a quantified variable intuitively correspond to a strategy in the sense that it becomes a function from a sequence of moves to a choice to make following this sequence. The logic GFG-QPTL is then already adapted for the simple strategic question of the existence of a winning strategy for a player but is not enough for more complex statements.

Finally, we turn to using hyperteam semantics for strategic reasoning. Building on the results of GFG-QPTL, that, as we just saw, can already quantify over strategies, we only miss the binding operator to tackle strategic reasoning. For this reason, we do not build a hyperteam semantics for SL but rather propose a new logic, that we call *Plan Logic* (PL for short), that has the same syntax as SL but for which the first-class citizen is a *plan* rather than a strategy. A plan is a simple sequence of actions, hence, it is a linear object, whereas a strategy is a branching object. Thanks to the same kinds of restrictions to the response functions as we did for GFG-QPTL, we enforce realizability *by design*. Furthermore, PL is capable of expressing the existence of a Nash equilibrium.

Game theoretic semantics for our logics. Besides the (compositional) hyperteam semantics, we provide a game-theoretic semantics for closed formulas in prenex form for each obtained logic. Even though the closeness of the formula is not a strong assumption and is mainly for concise presentation, the prenex form is mandatory, and it is unclear how to circumvent this restriction.

For ADIF, the game-theoretic semantics resembles a classic evaluation game in the sense that, following the quantification prefix, Eloise and Abelard choose in turn the

value of their respective quantified variables. However, and as opposed to a standard evaluation game, once each variable is assigned a value, instead of concluding the game by checking whether the quantifier-free subformula holds, players can *challenge* each other's dependencies by changing the value of one of their variables. The way the other player reacts will determine whether their decision respect the dependencies. The obtained game is infinite-horizon, as players can keep on challenging each other indefinitely. Still, we resort to the order of variables in the quantifier prefix and the dependency constraints to elect the winner of an infinite play.

Next, for GFG-QPTL, we simply use an iterated evaluation game: at each time point, players choose a value for each variable in the order of the quantifier prefix. Independence from the future lead to perfect information because, when a player chooses the value of a variable at time t , they can use all the values that have been chosen up to this point. For this reason, the game doesn't include the challenge mechanisms of ADIF. This game is the basis of our proof that the model-checking problem for GFG-QPTL is 2-EXPTIME-complete.

Finally, for PL, we design a game where Boolean operators are personified as *operator players* to reveal their inherent game-theoretic nature. A formula in prenex form consists of a quantifier prefix followed by a Boolean combination of *goal formulas*. For our construction, we restrict to goal formulas in *disjunctive* or *conjunctive* forms. An operator player's strategy is existentially quantified if the operation is a disjunction and universally quantified otherwise. The purpose of the operator player is to demonstrate that the choice among the various disjunct (or conjunct) to be verified, in order to prove (or disprove) the formula, is made on the fly, and thus, in a realizable manner. This result highlights the intrinsic realizability nature of PL. Thanks to this game, we have shown how to translate an SL formula into a PL formula that is equivalent for the timeline semantics of SL.

More dependence by considering delays. Now that realizability is achieved, we can explore more exotic kinds of dependency. For instance, *delays* between decisions and the observation of their effects have been studied in the context of *Gale-Stewart games* — recall that a Gale-Stewart game is played between two players who alternately choose a letter from a finite alphabet, with the winner of the play determined by the membership of the resulting infinite word in a given language. The delayed version of Gale-Stewart games is called *delay games*. To adapt strategic reasoning to account for delays, we first generalize delay games to multiple players: we introduce *dependency matrices*, a new formalism to

express delays, and even lookaheads, between multiple players. A dependency matrix specifies the delays that each player’s strategy must respect. In this sense, they are well-suited for developing a game-theoretic semantics for delayed strategic reasoning.

Organization of the manuscript

This manuscript is organized as follows. In Chapter 1, we recall the syntax and semantics of FO, IFL and DFL and all associated relevant logics. Then, in Chapter 2, we develop the hyperteam semantics for FO without dependencies and we state some properties that are inherent to the approach. Once the concept of hyperteam is properly presented, we come to the instantiation with first, the logic to reason about dependencies ADIF in Chapter 3, then, the temporal logic with realizable quantifiers GFG-QPTL in Chapter 4 and finally, the logic for realizable strategic reasoning PL in Chapter 5. At last, in Chapter 6, we present the multiplayer generalization of delay games: the dependency matrices.

The work on ADIF has been already published in the *Annals of Pure and Applied Logics* [BBDM23b] journal and the one for GFG-QPTL has been published in *ACM Transaction of Computational Logics* [BBDM23a]. The dependency matrices have been presented in the conference *Foundations of Software Technology and Theoretical Computer Science* [BPS22] in 2022. The work on PL is currently under submission.

Notations

In the document, we resort to very standard mathematical notations such as \mathbb{N} for the set natural numbers and \mathbb{Z} for the set of integers.

Given two natural numbers $k \leq n$, we denote by $\llbracket k, \dots, n \rrbracket$ the set $\{k, k + 1, \dots, n\}$.

Given a set S , we use as usual S^* (resp., S^ω) to denote the set of finite (resp. infinite) sequences over the alphabet S , and $S^\infty = S^* \cup S^\omega$. For $\pi \in S^*$ and $i \in \mathbb{N}$, we use $\pi[i]$, $\pi[: i]$, $\pi[i :]$, $\text{fst}(\pi)$, and $\text{lst}(\pi)$, to denote, respectively, the i -th element of π , the prefix of π up to index i included, the suffix of π from index i included, the first (0-th) element of π , and, finally, the last element of π .

Grammars are displayed with a $|$ symbol separating the different production rules.

PRELIMINARIES

In this chapter, we recall the syntax and semantics of First-Order Logic in a first section, and then Dependence/Independence-Friendly Logic in a second section.

1.1 Recall on First-Order Logic

In the domain of computer science and logical reasoning, *First-Order Logic* (FO for short) [HA28] serves as a fundamental framework for formalizing statements about objects and their properties. Also known as predicate logic, FO allows us to express relationships between elements, quantify over variables, and define properties and functions.

At its core, FO consists of:

Terms: These represent objects and variables in a domain. They could denote specific individuals, constants, or variables that stand for any element in the domain.

Predicates: These express relationships or properties of objects. They are statements that can be true or false depending on the elements they are applied to.

Quantifiers: Including universal (\forall) and existential (\exists) quantifiers, which allow us to make statements about all elements in a domain or to assert the existence of elements with specific properties.

Logical Connectives: Such as conjunction (\wedge), disjunction (\vee), implication (\rightarrow), and negation (\neg), which enable the construction of complex statements from simpler ones.

In FO, formulas are constructed using a combination of these elements. For instance: $\forall x.P(x)$ asserts that a property P holds for all elements x in the domain. $\exists x.Q(x)$ states that there exists an element x in the domain for which property Q holds.

The expressiveness of FO allows us to formalize a wide range of statements and reason about their truth or falsehood within a given structure.

1.1.1 Syntax

The syntax of FO is defined with respect to a *signature* $\mathcal{S} = \langle \mathcal{F}, \mathcal{P} \rangle$, which is a couple of set of symbols: \mathcal{F} is the set of *function symbols* and \mathcal{P} is the set of *predicate symbols*. The *arity* of a symbol is given by $\text{AR} : \mathcal{F} \cup \mathcal{P} \rightarrow \mathbb{N}$. For this work, we will avoid specifying the signature when it is not mandatory. We also fix an infinite set of variables $\text{Var} = \{x, y, \dots\}$.

A *term* is either a constant (a function symbol of arity 0), which is a specific element of a domain, or a variable x , which is a placeholder for an element, or the application of a function symbol f on a sequence terms of length $\text{AR}(f)$. Intuitively, a term represents an element of a structure.

Definition 1.1. A term τ is inductively defined as follows, where x is a variable, f is a function symbol and $\vec{\tau}$ is a sequence of terms of length $\text{AR}(f)$.

$$\tau := x \mid f(\vec{\tau})$$

A *predicate* P takes a sequence of $\text{AR}(P)$ terms in input. It represents a property over the elements of the domain (if $\text{AR}(P) = 1$) or relations between them (if $\text{AR}(P) > 1$). A predicate is the smallest brick for building *FO formulas*.

Definition 1.2. A FO formula is defined inductively as follows, where x is a variable, P is a predicate symbol and $\vec{\tau}$ is a sequence of terms of length $\text{AR}(P)$.

$$\varphi := P(\vec{\tau}) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\varphi \mid \forall x\varphi$$

For convenience, we will also use the connectives \rightarrow (implies) and \leftrightarrow (is equivalent to), as syntactic sugar, defined as follows: $\varphi \rightarrow \phi \stackrel{\text{def}}{=} \neg\varphi \vee \phi$ and $\varphi \leftrightarrow \phi \stackrel{\text{def}}{=} (\varphi \rightarrow \phi) \wedge (\phi \rightarrow \varphi)$. The set of free variables of an FO formula φ (variables that appear in φ without being quantified) is denoted $\text{FREE}(\varphi)$. A *sentence* (or closed formula) is a formula with no free variables.

Example 1.1. Consider Alice wants to express her taste in fruits: she likes apple and bananas but dislikes cherries. A suited signature to formalize this sentence in FO could be $\{\langle \mathbf{apple} [0], \mathbf{banana} [0], \mathbf{cherry} [0] \rangle, \langle \mathbf{Likes} [1] \rangle\}$ where the arity of the different symbols is between brackets. The FO sentence is: $\varphi_1 = \text{Likes}(\mathbf{apple}) \wedge \text{Likes}(\mathbf{banana}) \wedge \neg\text{Likes}(\mathbf{cherry})$.

With the same signature, we can express more properties of Alice's taste in fruit: $\varphi_2 = \exists x \neg \text{Likes}(x)$ expresses that there is a fruit that Alice does not like and $\varphi_3 = \forall x \text{Likes}(x)$ expresses that Alice likes all fruits.

1.1.2 First-Order Structure

Now that we have defined the syntax of FO for a given signature $\mathcal{S} = \langle \mathcal{F}, \mathcal{P} \rangle$, we need a structure to evaluate FO formulas. Such a structure is called an *FO structure* and is composed of a *domain* \mathbb{D} , a \mathcal{F} -algebra $\mathcal{F}^{\mathcal{M}}$ to give an interpretation of the function symbols and a set of relations $\mathcal{P}^{\mathcal{M}}$, one for each predicate symbol, to give an interpretation of those predicates.

Formally, given a signature $\mathcal{S} = \langle \mathcal{F}, \mathcal{P} \rangle$, a FO structure \mathcal{M} is a tuple $\langle \mathbb{D}, \mathcal{F}^{\mathcal{M}}, \mathcal{P}^{\mathcal{M}} \rangle$, where

- \mathbb{D} is the domain of the structure
- $\mathcal{F}^{\mathcal{M}} = \{f^{\mathcal{M}} : \mathbb{D}^{\text{AR}(f)} \rightarrow \mathbb{D} \mid f \in \mathcal{F}\}$
- $\mathcal{P}^{\mathcal{M}} = \{P^{\mathcal{M}} \subseteq \mathbb{D}^{\text{AR}(P)} \mid P \in \mathcal{P}\}$.

By convention, we assimilate \mathbb{D}^0 with \top (true) and \emptyset with \perp (false) in the case of a predicate with arity 0.

Example 1.2. Continuing example 1.1, a FO structure would inform effectively about the tastes of Alice. The domain $\mathbb{D}_{\text{fruits}} = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$. The interpretation of the constants tells us which value is to be interpreted as which fruit: $\mathbf{apple}^{\mathcal{M}} = \mathbf{a}$ and $\mathbf{banana}^{\mathcal{M}} = \mathbf{b}$ and $\mathbf{cherry}^{\mathcal{M}} = \mathbf{c}$. Note that other domains or interpretations of the constants are possible but these are the ones that makes sense when it comes to talk about three different fruits. The taste of Alice is modeled in the interpretation of the predicate Likes. Lets consider two different interpretations: $\text{Likes}^{\mathcal{M}_1} = \{\mathbf{a}, \mathbf{b}\}$ and $\text{Likes}^{\mathcal{M}_2} = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$. We then have two different FO structures:

- $\mathcal{M}_1 = \langle \mathbb{D}_{\text{fruits}}, \{\mathbf{apple}^{\mathcal{M}}, \mathbf{banana}^{\mathcal{M}}, \mathbf{cherry}^{\mathcal{M}}\}, \text{Likes}^{\mathcal{M}_1} \rangle$ and
- $\mathcal{M}_2 = \langle \mathbb{D}_{\text{fruits}}, \{\mathbf{apple}^{\mathcal{M}}, \mathbf{banana}^{\mathcal{M}}, \mathbf{cherry}^{\mathcal{M}}\}, \text{Likes}^{\mathcal{M}_2} \rangle$.

In \mathcal{M}_1 , Alice likes apples and bananas and in \mathcal{M}_2 , she likes all the fruits.

1.1.3 Semantics

What is left to give a meaning to an FO formula φ is a value for each variable. This is achieved by the notion of *assignment*, which is central in the work presented in this thesis. An assignment χ is a mapping from variables to the domain: $\chi : \text{Var} \rightarrow \mathbb{D}$. An

assignment is only a partial mapping as only the variables appearing in the formula φ outside the range of a quantifier need to be assigned a value. Those variables are called the *free* variables of φ . The set of assignments is denoted by \mathbf{Asg} and the set of assignments defined over a finite subset V of variables is denoted by $\mathbf{Asg}(V)$.

Once each variable have a value, we can compute the values of terms. Given a FO structure \mathcal{M} and an assignment χ , we define $\llbracket \tau \rrbracket_{\mathcal{M}, \chi}$, the value of term τ in \mathcal{M} with χ by induction on τ :

1. $\llbracket x \rrbracket_{\mathcal{M}, \chi} = \chi(x)$,
2. $\llbracket f(\vec{\tau}) \rrbracket_{\mathcal{M}, \chi} = f^{\mathcal{M}}(\llbracket \vec{\tau} \rrbracket_{\mathcal{M}, \chi})$

where $\llbracket \vec{\tau} \rrbracket_{\mathcal{M}, \chi}$ is the tuple obtained by the application of $\llbracket \cdot \rrbracket_{\mathcal{M}, \chi}$ on each term of $\vec{\tau}$.

Definition 1.3. *Given a signature \mathcal{S} , an FO structure \mathcal{M} , an FO assignment χ and an FO formula φ , the FO semantics relation $\mathcal{M}, \chi \models_{FO} \varphi$ is defined inductively on φ as follows.*

1. $\mathcal{M}, \chi \models_{FO} P(\vec{\tau})$ if $\llbracket \vec{\tau} \rrbracket_{\mathcal{M}, \chi} \in P^{\mathcal{M}}$
2. $\mathcal{M}, \chi \models_{FO} \neg \varphi$ if $\mathcal{M}, \chi \not\models_{FO} \varphi$
3. $\mathcal{M}, \chi \models_{FO} \varphi_1 \vee \varphi_2$ if $\mathcal{M}, \chi \models_{FO} \varphi_1$ or $\mathcal{M}, \chi \models_{FO} \varphi_2$
4. $\mathcal{M}, \chi \models_{FO} \varphi_1 \wedge \varphi_2$ if $\mathcal{M}, \chi \models_{FO} \varphi_1$ and $\mathcal{M}, \chi \models_{FO} \varphi_2$
5. $\mathcal{M}, \chi \models_{FO} \exists x \varphi$ if there is a value $v \in \mathbb{D}$ such that $\mathcal{M}, \chi[x \mapsto v] \models_{FO} \varphi$
6. $\mathcal{M}, \chi \models_{FO} \forall x \varphi$ if for every value $v \in \mathbb{D}$ we have $\mathcal{M}, \chi[x \mapsto v] \models_{FO} \varphi$

Example 1.3. *We can apply the semantics to the formulas of example 1.1 with the FO structures defined in example 1.2. Since φ_1 , φ_2 and φ_3 are sentences, no assignment of variables is required. Considering the first formula $\varphi_1 = \text{Likes}(\mathbf{apple}) \wedge \text{Likes}(\mathbf{banana}) \wedge \neg \text{Likes}(\mathbf{cherry})$, it is straightforward to see that $\mathcal{M}_1, \emptyset \models_{FO} \varphi_1$ and that $\mathcal{M}_2 \not\models_{FO} \varphi_2$.*

For $\varphi_2 = \exists x \neg \text{Likes}(x)$, there is $\mathbf{c} \in \mathbb{D}$ such that the assignment χ_1 defined on x by $\chi_1(x) = \mathbf{c}$ verifies $\mathcal{M}_1, \chi_1 \models \neg \text{Likes}(x)$. Thus we can deduce that $\mathcal{M}_1, \chi_1 \models \varphi_2$.

The semantics of quantifiers can be understood as choices in a game made by two players, Eloise and Abelard, that compete to make the formula respectively true or false. With this philosophy, Eloise chooses values of existentially quantified variables and Abelard chooses the values of universally quantified ones. Such a game is central in many results presented in this thesis and will be discussed for the diverse languages introduces as *the game-theoretic semantics* of the logic. We do not formalize this viewpoint on FO here.

1.1.4 Limitations of Tarski's-like quantification

FO expressiveness is constrained when it comes to capturing intricate dependencies due to the Tarskian semantics of quantifiers[Tar36]. The foundational principles within Tarski's approach impose limitations on its ability to articulate nuanced relationships or subtle dependencies within logical statements in the sense that it imposes a linear order on the dependencies between the quantified variables. Indeed, if one wants to make some existentially quantified variable x independent from some universally quantified variable y , the only possibility is to place the quantifier for x before the one for y in the syntax of the formula.

Example 1.4. *For instance, consider the following formulas:*

$$\begin{aligned}\varphi_1 &\stackrel{\text{def}}{=} \forall y \exists x P(x, y) \\ \varphi_2 &\stackrel{\text{def}}{=} \exists x \forall y P(x, y)\end{aligned}$$

In φ_1 , the choice of the value of y comes before the choice of the value of x , then, x depends on y . On the other hand, in φ_2 , it is the choice of the value of x that comes first, then x is independent of y .

It is not possible in FO to quantify existentially over x_1 and x_2 and universally over y_1 and y_2 such that x_1 depends only on y_1 and x_2 depends only on y_2 when the four variables are involved in common predicates.

Remark 1.1. *We assume that formulas are in prenex form here, i.e. that all quantifiers are in front of the formula. The formula $\varphi \stackrel{\text{def}}{=} \forall y (\exists x P(x, y)) \vee (\exists z P(z, y))$ showcase mutual independence between x and z . However, we cannot express a property on both x and z at the same time. Ultimately, φ is equivalent to a formula in prenex form and then, to a linear ordering of the dependencies between variables.*

To enforce independence, we have to enrich our syntax and adapt our semantics. In other words, we need a new logic.

1.2 Recall on Team Semantics

An approach to model dependencies in logic is to postpone the choice of a specific value when quantifying and then, to consider all possible values for the quantified variable. With this approach, the assignment is replaced with a set of assignments, named a *trump*

or a *team*, to keep track of all the possible values of variables. Intuitively, a team represents the imperfect information on the actual value of variables from the viewpoint of one player of the evaluation game (recall that Eloise aim at verifying the formula and Abelard aims at falsifying it). We see in this section how is formalized this intuition. For the whole subsection, we fix a signature \mathcal{S} and a \mathcal{S} -structure \mathcal{M} .

1.2.1 Representing dependences with Teams

For this subsection, we assume the viewpoint of Eloise when discussing the intuitions on the operators on teams. The dual explanation can be applied when considering the viewpoint of Abelard.

Formally, a team X over a set of variables V is a set of assignments $X \subseteq \text{Asg}(V)$. The set of teams over V is denoted by $\text{Teams}(V)$ and we define $\text{Teams} \stackrel{\text{def}}{=} \bigcup_{V \subseteq \text{Var}} \text{Teams}(V)$ and, given a set of variables W , we define $\text{Teams}_{\supseteq}(W) \stackrel{\text{def}}{=} \bigcup_{W \subseteq V} \text{Teams}(V)$. As a team replace the assignment in the semantic rules, we provide new operations to use on the team. The first operation we introduce is the partitioning of a team. Given a team $X \in \text{Teams}$, we define $\text{PART}(X) = \{(X_1, X_2) \in \text{Teams}^2 \mid X_1 \cup X_2 = X\}$. The *partitionning* operator is used for the semantic of the disjunctive operator. Intuitively, when Eloise has to satisfy $\varphi_1 \vee \varphi_2$, she can choose to split the team X between worlds in which she knows how to satisfy φ_1 and those in which she knows how to satisfy φ_2 . This way, she can ensure that either φ_1 or φ_2 is true for each possible world.

For universal quantifiers, we have to enlarge the team with every possible value for the quantified variable. We introduce the *cylindrification* of a team X on a variable x , denoted by $\text{CYL}(X, x)$ and defined as follows.

$$\text{CYL}(X, x) = \{\chi[x \mapsto v] \mid \chi \in X, v \in \mathbb{D}\}$$

This definition is deterministic in the sense that there is no specific value that has to be chosen for x .

For existential quantifiers, we extend each assignment of the team with a value. The function that associates a value to each assignment is called the *Response function*. Formally, a response function is a function $F : \text{Asg} \rightarrow \mathbb{D}$. The set of response function is denoted by Rsp . The *extension* of a team X over a variable x with the response F is

denoted by EXT and defined as follows.

$$\text{EXT}(X, F, x) = \{\chi[x \mapsto F(\chi)] \mid \chi \in X\}$$

Remark that the extension does not increase the size of the team.

Of course, we want to allow some kind of *dependence* or *independence constraints* between variables. Such a constraint is given in the form of a equivalence relation \sim between assignments.

Definition 1.4. *Given a dependence \sim , a response function F is uniform with respect to a dependence \sim if the following holds.*

$$\text{For every } \chi_1, \chi_2 \in \text{Asg}, \text{ if } \chi_1 \sim \chi_2, \text{ then } F(\chi_1) = F(\chi_2)$$

Various equivalence relations can be considered this way. For this work, we focus on a special kind of dependence relation which is the dependence on (or independence from) a set of variable. Given a set of variables W , we denote by \sim_{+W} (*resp.* \sim_{-W}) the equivalence relation defined as follows, where $\chi_1, \chi_2 \in \text{Asg}(V)$ with V a set of variables.

$$\begin{aligned} \chi_1 \sim_{+W} \chi_2 & \text{ iff } \chi_1 \upharpoonright W = \chi_2 \upharpoonright W \\ \chi_1 \sim_{-W} \chi_2 & \text{ iff } \chi_1 \upharpoonright (V \setminus W) = \chi_2 \upharpoonright (V \setminus W) \end{aligned}$$

Remark that these relations are defined only on assignments with the same domain. Given a set of variable W , we denote by $\text{Rsp}_{\pm W}$ the set of $\sim_{\pm W}$ -uniform responses.

1.2.2 Dependence/Independence Logic

We now have the material to define a logic using teams as models. The language in itself is an extension of FO by allowing to specify dependence (or independence) constraints on quantifiers. However, we have to assume that formulas are in negation normal form.

$$\varphi := \text{P}(\vec{\tau}) \mid \neg \text{P}(\tau) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists^{\pm W} x \varphi \mid \forall^{\pm W} x \varphi$$

We use \exists and \forall as syntactic sugar for $\exists^{-\emptyset}$ and $\forall^{-\emptyset}$ respectively.

We call *existential dependence-friendly logic* (DFL_{\exists} for short), *resp.* *universal dependence-friendly logic* (DFL_{\forall} for short), the fragment with only \exists^{+W} and \forall (*resp.* \forall^{+W} and \exists) quantifiers and we call *dependence-friendly logic* (DFL for short), the union

of DFL_{\exists} and DFL_{\forall} . Similarly, we call *existential independence-friendly logic* (IFL_{\exists} for short), *resp. universal independence-friendly logic* (IFL_{\forall} for short), the fragment with only \exists^{-W} and \forall (*resp.* \forall^{-W} and \exists) quantifiers and we call *independence-friendly logic* (IFL for short), the union of IFL_{\exists} and IFL_{\forall} . The whole logic, *dependence/independence-friendly logic* (DIF for short) is the union of DFL and IFL and we call DIF_{\exists} the fragment $\text{DFL}_{\exists} \cup \text{IFL}_{\exists}$ and DIF_{\forall} the fragment $\text{DFL}_{\forall} \cup \text{IFL}_{\forall}$.

Definition 1.5 (DIF Semantics). *The Hodges' semantic relation $\mathcal{M}, X \models_{\text{DIF}}^{\beta} \varphi$ for $\text{DIF}_{\bar{\beta}}$ is inductively defined as follows, for all $\text{DIF}_{\bar{\beta}}$ formulas φ and teams $X \subseteq \text{Asg}_{\subseteq}(\text{FREE}(\varphi))$, with $\beta \in \{\exists, \forall\}$ and $\beta \neq \bar{\beta}$:*

1. (a) $\mathcal{M}, X \models_{\text{DIF}}^{\forall} P(\vec{x})$ if, for all $\chi \in X$, it holds that $\chi(\vec{x}) \in P^{\mathcal{M}}$;
- (b) $\mathcal{M}, X \models_{\text{DIF}}^{\forall} \neg P(\vec{x})$ if, for all $\chi \in X$, it holds that $\chi(\vec{x}) \notin P^{\mathcal{M}}$;
- (c) $\mathcal{M}, X \models_{\text{DIF}}^{\forall} \varphi_1 \wedge \varphi_2$ if $\mathcal{M}, X \models_{\text{DIF}}^{\forall} \varphi_1$ and $\mathcal{M}, X \models_{\text{DIF}}^{\forall} \varphi_2$;
- (d) $\mathcal{M}, X \models_{\text{DIF}}^{\forall} \varphi_1 \vee \varphi_2$ if $\mathcal{M}, X_1 \models_{\text{DIF}}^{\forall} \varphi_1$ and $\mathcal{M}, X_2 \models_{\text{DIF}}^{\forall} \varphi_2$, for some bipartition $(X_1, X_2) \in \text{PART}(X)$;
- (e) $\mathcal{M}, X \models_{\text{DIF}}^{\forall} \exists^{\pm W} x \varphi$ if $\mathcal{M}, \text{EXT}(X, F, x) \models_{\text{DIF}}^{\forall} \varphi$, for some $F \in \text{Rsp}_{\pm W}$;
- (f) $\mathcal{M}, X \models_{\text{DIF}}^{\forall} \forall x \varphi$ if $\mathcal{M}, \text{CYL}(X, x) \models_{\text{DIF}}^{\forall} \varphi$;
2. (a) $\mathcal{M}, X \models_{\text{DIF}}^{\exists} P(\vec{x})$ if there exists $\chi \in X$ such that $\chi(\vec{x}) \in P^{\mathcal{M}}$;
- (b) $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \neg P(\vec{x})$ if there exists $\chi \in X$ such that $\chi(\vec{x}) \notin P^{\mathcal{M}}$;
- (c) $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \varphi_1 \wedge \varphi_2$ if $\mathcal{M}, X_1 \models_{\text{DIF}}^{\exists} \varphi_1$ or $\mathcal{M}, X_2 \models_{\text{DIF}}^{\exists} \varphi_2$, for all bipartitions $(X_1, X_2) \in \text{PART}X$;
- (d) $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \varphi_1 \vee \varphi_2$ if $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \varphi_1$ or $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \varphi_2$;
- (e) $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \exists x \varphi$ if $\mathcal{M}, \text{CYL}(X, x) \models_{\text{DIF}}^{\exists} \varphi$;
- (f) $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \forall^{\pm W} x \varphi$ if $\mathcal{M}, \text{EXT}(X, F, x) \models_{\text{DIF}}^{\exists} \varphi$, for all $F \in \text{Rsp}_{\pm W}$.

Remark that the conjunction is not the dual of the disjunction. Some references chose to name the \vee operator *splitjunction* when given such a semantic rule to differentiate it from the usual disjunction. Also remark that, as expected, the existential and universal quantifications are not dual of each other. These remarks are at the core of the necessity to consider formulas in negation normal form.

1.2.3 Examples

Example 1.5. Let us fix a structure $\mathcal{M} = \langle \mathbb{D}, (\emptyset, =^{\mathcal{M}}) \rangle$ with $\mathbb{D} = \{0, 1\}$ and $=^{\mathcal{M}}$ being the equality. Consider the following DFL_{\exists} sentence.

$$\varphi \stackrel{\text{def}}{=} \forall x_1 \exists y_1 \forall x_2 \exists^{+y_1} y_2 (x_1 = y_2 \vee x_2 = y_1)$$

Let us show that this sentence is true by unraveling the semantics rules. We first start with the empty team \emptyset . We apply first the $CYL(X, x)$ operator.

$$\mathcal{M}, \emptyset \models_{DIF}^{\forall} \varphi \text{ iff } \mathcal{M}, X_1 \models_{DIF}^{\forall} \exists y_1 \forall x_2 \exists^{+y_1} y_2 (x_1 = y_2 \vee x_2 = y_1)$$

With $X_1 \stackrel{\text{def}}{=} CYL(\emptyset, x_1) = \{x_1 : 0, x_1 : 1\}$.

We choose the response function F defined by $F(x_1 : 0) = 0$ and $F(x_1 : 1) = 1$ and then apply the EXT operator on X_1, F and y_1 .

$$\mathcal{M}, X_1 \models_{DIF}^{\forall} \exists y_1 \forall x_2 \exists^{+y_1} y_2 (x_1 = y_2 \vee x_2 = y_1) \text{ if } \mathcal{M}, X_2 \models_{DIF}^{\forall} \forall x_2 \exists^{+y_1} y_2 (x_1 = y_2 \vee x_2 = y_1)$$

$$\text{With } X_2 \stackrel{\text{def}}{=} EXT(X_1, F, y_1) = \left\{ \begin{array}{cc} x_1 : 0 & x_1 : 1 \\ y_1 : 0 & y_1 : 1 \end{array} \right\}.$$

We then apply the CYL operator again.

$$\mathcal{M}, X_2 \models_{DIF}^{\forall} \forall x_2 \exists^{+y_1} y_2 (x_1 = y_2 \vee x_2 = y_1) \text{ iff } \mathcal{M}, X_3 \models_{DIF}^{\forall} \exists^{+y_1} y_2 (x_1 = y_2 \vee x_2 = y_1)$$

$$\text{With } X_3 \stackrel{\text{def}}{=} CYL(X_2, x_2) = \left\{ \begin{array}{cccc} x_1 : 0 & x_1 : 0 & x_1 : 1 & x_1 : 1 \\ y_1 : 0 & , & y_1 : 0 & , & y_1 : 1 & , & y_1 : 1 \\ x_2 : 0 & x_2 : 1 & x_2 : 0 & x_2 : 1 \end{array} \right\}.$$

We now have to choose a response function for y_2 depending only on y_1 . We define G by $G(y_1 : 0) = 0$ and $G(y_1 : 1) = 1$.

$$\mathcal{M}, X_3 \models_{DIF}^{\forall} \exists^{+y_1} y_2 (x_1 = y_2 \vee x_2 = y_1) \text{ if } \mathcal{M}, X_4 \models_{DIF}^{\forall} (x_1 = y_2 \vee x_2 = y_1)$$

$$\text{With } X_4 \stackrel{\text{def}}{=} CYL(X_3, x_2) = \left\{ \begin{array}{cccc} x_1 : 0 & x_1 : 0 & x_1 : 1 & x_1 : 1 \\ y_1 : 0 & y_1 : 0 & y_1 : 1 & y_1 : 1 \\ x_2 : 0 & , & x_2 : 1 & , & x_2 : 0 & , & x_2 : 1 \\ y_1 : 0 & y_1 : 0 & y_1 : 1 & y_1 : 1 \end{array} \right\}.$$

We can check that $x_1 = y_2$ is verified for every assignment in X_4 , proving that \mathcal{M}

satisfies φ .

Example 1.6. We can use the dependence constraints to compare the size of sets in a FO structure. Consider the following signature: $\mathcal{S} = \langle \emptyset, \{P [1], Q [1], = [2]\} \rangle$ where we assume that the $=$ relation is to be interpreted as the equality. We can define a formula that states "the set denoted by P is bigger than the set denoted by Q " as follows.

$$\varphi_{inj} \stackrel{\text{def}}{=} \forall x_1 \forall x_2 \exists^{+x_1} y_1 \exists^{+x_2} y_2 (P(x_1) \rightarrow Q(y_1)) \wedge (P(x_2) \rightarrow Q(y_2)) \wedge (x_1 = x_2 \leftrightarrow y_1 = y_2)$$

Intuitively, φ_{inj} defines an injection from P to Q . The injection is computed by each y_i , on the respective inputs x_i . The dependence constraint ensures that each y_i is the result of a single function f_i and the subformula $P(x_i) \rightarrow Q(y_i)$ ensures that the function realized by y_i goes from P to Q . The last subformula $x_1 = x_2 \leftrightarrow y_1 = y_2$ ensures two things: first, that $f_1 = f_2$ thanks to $x_1 = x_2 \rightarrow y_1 = y_2$, and second, that f_i is an injection thanks to $x_1 = x_2 \leftarrow y_1 = y_2$.

1.2.4 Limitations of Team Semantics

The team semantics approach at expressing dependencies has two main downsides: it does not allow for negations and it prevents from using dependencies on both kinds of quantifiers in a meaningful manner.

The lack of negation underlines the presence of undetermined formulas. An undetermined formula φ is such that both φ and the positive form of $\neg\varphi$ are false. It is a consequence of the intrinsic imperfect information nature of the semantics. Semantics have been developed to try to add the negation and they lead to the consideration of the new boolean operators that are the duals of the negation and the splitjunction. Unfortunately, even by doing so, the obtained logic does not enjoy determinacy.

The inability to use dependencies on both kinds of quantifiers is the main limitation for our purpose. A semantics using two teams, one for the information of Eloise and the other for Abelard, have been proposed. The resulting logic is still undetermined cannot be adapted for the strategic reasoning.

1.3 Conclusion

Independence-Friendly Logic was originally introduced by [HS89], and later extensively studied, *e.g.*, in [MSS11], as an extension of First-Order Logic with informational

independence as first-class notion, and with applications in semantics of *natural language* in mind. Unlike in FO, where quantified variables always functionally depend on all the previously quantified ones, one can force in IFL the values of certain quantified variables to be chosen independently of the values of some specific variables quantified before in the formula. This originally has been syntactically represented by means of the so called *slashed operator* notation, where, for instance, $(\exists x/W)\varphi$ (written $\exists^{-W}x\varphi$ in this work) is intended to mean that variable x must be chosen independently (i.e., without knowledge) of the values of the variables contained in the set W . The logic has a nice game-theoretic semantics [HS97], given in terms of games of imperfect information, where a sentence is true if the verifier player, usually called Eloise, has a *strategy* to win the semantic game. If the falsifier player, Abelard, has a *winning strategy*, then the sentence is declared false. Since games with imperfect information are considered here, neither situation may occur, as the specific game may be *undetermined*. In this case, the corresponding sentence is neither true nor false, therefore establishing a failure of the *law of excluded middle*. [Hod97b] later developed a compositional semantics for IFL, by defining satisfaction *w.r.t* a set of assignment, called *trump* (*a.k.a. teams*, in later iterations of the idea and in this work), instead of a single assignment as in classic Tarskian semantics [Tar36; Tar44] of FO.

Dependence Logic [Vää07] (DL) takes a slightly different approach to the problem, by separating quantifiers from dependence specification. This is achieved by adding to FO the so called *dependence atoms* of the form $=(x, y)$, with the intended meaning that the value of variable y is completely determined by, hence functionally dependent on, the value of variables in the vector \vec{x} . The separation of dependence constraints and quantifiers can express very naturally dependencies on both quantified and non quantified variables, and allows for a quite flexible approach to reasoning about dependence and independence. DL has also been extended with other types of atoms like, *e.g.*, *independence atoms* [GV13] and *inclusion/exclusion atoms* [Gal12]. The logic is expressively equivalent to both IFL and the existential fragment of *Second Order Logic* (SO) [Chu56; Sha91]. As such, DL still allows for undetermined sentences and is not closed under classical negation.

To recover closure under negation and, consequently, the law of excluded middle, [Vää07] introduced *Team Logic* (TL), an extension of DL with the so called *contradictory negation* \sim , an idea already investigated by [Hin96] in the context of IFL, where it was allowed only in front of a sentence. TL is substantially more expressive than DL, reaching the full descriptive power of SO, covering, thus, the entire polynomial hierarchy [Sto76]. However, in order to recover the nice properties of FO, such as the duality

between Boolean connectives and quantifiers, TL requires two different versions of the propositional connectives, \neg and \sim for negation, \wedge and \oplus for conjunction, \vee and \otimes for disjunction, as well as an additional pseudo quantifier $!x$ called *shriek*. This approach also bears significant consequences. In particular, TL lacks any meaningful direct game-theoretic interpretation, as also pointed out by [Vää07], which DL still retains, mainly thanks to its equivalence with existential SO.

There is a well-known connection between logics to reason with or about informational independence and the extension of first-order logic with the *partially ordered* (a.k.a. *branching* or *Henkin*) *quantifiers*, originally proposed by [Hen61] to overcome the linear dependence intrinsic in classic quantifier prefixes (see also [KM95] for a comprehensive survey on the topic). For instance, the sentence $\left(\begin{smallmatrix} \forall x_1 \exists y_1 \\ \forall x_2 \exists y_2 \end{smallmatrix}\right)\varphi$ states that for all x_1 and x_2 , there exists a value for y_1 , that only depends on x_1 , and a value for y_2 , that only depends on x_2 , such that φ is true. Sentences like this can easily be expressed in IFL by means of suitable independence constraints. For the sentence in the example, $\forall x_1 \forall x_2 \exists^{+x_1} y_1 \exists^{+x_2} y_2 \varphi$ is an equivalent IFL sentence. Similarly to IFL, the prenex fragment of the logic with Henkin quantifiers, where a Henkin quantifier prefix is followed by a quantifier-free FO formula [Wal70], is known to be expressively equivalent to Σ_1^1 , the existential fragment of SO, while the full (non-prenex) logic was proved to be able to express Δ_2^1 -properties by [End70].

As observed by [BG86], logics with Henkin quantifiers exhibit an asymmetric nature from a game-theoretic viewpoint, in that they typically consider only whether the existential player, Eloise, has a winning strategy that proves a formula true. This is, instead, solved in IFL, at the cost of indeterminacy of the logic, by introducing two satisfaction relations, one for truth and one for falsity, and by defining them in terms of uniform strategies for the players [MSS11]. More specifically, a strategy for a player, either Eloise or Abelard, is said to be uniform if for every variable x , which is controlled by that player and is required to be independent of a set of variables W , the strategy always chooses the same value in all the states of the game that differ only for the values of the variables in W . To win the game and prove the sentence true, Eloise is required to have a uniform strategy that wins every play induced by her strategy. These compatible plays need not be compatible with any uniform strategy of the adversary, meaning that when evaluating truth of a sentence, no restrictions to the universal quantifiers controlled by Abelard actually apply. A similar situation happens when evaluating falsity of a sentence. In this case, Abelard, needs to have a uniform strategy that wins all the compatible plays. Here,

the constraints on the existential variables are ignored. The imperfect information nature of these games manifests itself in the uniformity requirements that leads to indeterminacy of the logic. This, in turn, implies that some sentences are neither true nor false. For instance, $\forall x \exists (y/\{x\}) x = y$ is undetermined as Eloise cannot copy the value of x when choosing for y and Abelard cannot guess the future value of y when choosing for x .

The situation described above is also reflected in Hodges' separate use of trumps and co-trumps in the compositional semantics he proposed for IFL. His idea of using sets of assignments allows for mimicking the uniformity constraints on the strategies in a compositional way. Essentially, a trump records all the states, represented here as assignments, the game could be in, depending on the possible choices made by Abelard and the corresponding responses by Eloise. These assignments correspond, intuitively, to the (partial) plays compatible with the strategy followed by Eloise when evaluating the formula. A trump can, then, encode the uncertainty that Eloise has about the actual current state of the play, in that assignments that only differ for the variables in W are indistinguishable to Eloise when she has to choose the value of a variable x that is independent of the variables in W . This allows Eloise to make her choice in each such state in a uniform way and adhere to the constraints on her variables when trying to prove the truth of the formula. Analogously, a co-trump encodes the states induced by the possible choices of Eloise and allows Abelard to behave uniformly when he wants to falsify the formula.

A HYPERTEAMS SEMANTICS

In this chapter, we propose a generalization of the team semantics approach that allows us to incorporate negation into the framework in a natural way and obtain a fully determined logic. Similarly to teams, the idea is that the interpretations of the free variables correspond to the choices that the two players could make up to the current stage of the game, i.e., the stage where the formula φ has to be evaluated. These possible choices are organized in a two-level structure, i.e., a set of sets of assignments, each level summarizing the information about the choices a player may have made in previous turns. We call such a structure a *hyperteam*.

The chapter is organized as follows. In Section 2.1, we present hyperteams and the various operations that we can apply to them: we show how to *dualize*, *extend* and *partition* hyperteams; and also, we define a relation to compare hyperteams. Then, in Section 2.2, we present the hyperteam semantics rules for FO. Following, in Section 2.3, we state the main results that come with hyperteam semantics. Thereafter, in Section 2.4, we compare the hyperteam semantics with the classic Tarskian one and show that the hyperteam semantics follows the intuitive explanation below when there are no dependence constraints. Finally, in Section 2.5, we give an alternative semantics that is a form of Skolemization/Herbrandization of the semantics, that we call the *Meta* semantics.

2.1 A two-level framework: the Hyperteam

In order to evaluate the formula φ on a hyperteam, a player — Eloise or Abelard — chooses a team, while its opponent chooses one assignment in that set where φ must hold. 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. If $\alpha = \exists\forall$, Eloise chooses the teams, while Abelard chooses one of those assignments; if $\alpha = \forall\exists$, the dual reasoning applies. In a sense, the level associated with a given player, say Eloise, encodes the uncertainty that the opponent Abelard has about her actual choices up to that stage.

Given a flag $\alpha \in \{\exists\forall, \forall\exists\}$, we denote by $\bar{\alpha}$ the dual flag, i.e., $\bar{\alpha} \in \{\exists\forall, \forall\exists\}$ with $\bar{\alpha} \neq \alpha$.

The idea described above is, then, captured by the notion of *hyperteam (of assignments)*, namely a set of teams defined over some arbitrary set $V \subseteq \text{Var}$:

$$\text{HypTeams} \stackrel{\text{def}}{=} \{\mathfrak{X} \subseteq \text{Teams}(V) \mid V \subseteq \text{Var}\}.$$

By $\text{HypTeams}(V) \stackrel{\text{def}}{=} \{\mathfrak{X} \in \text{HypTeams} \mid \mathfrak{X} \subseteq \text{Teams}(V)\}$ we denote the set of hyperteams over V , while $\text{HypTeams}_{\supseteq}(V) \stackrel{\text{def}}{=} \{\mathfrak{X} \in \text{HypTeams} \mid \mathfrak{X} \subseteq \text{Teams}(W) \text{ with } V \subseteq W \subset \text{Var}\}$ contains the hyperteams defined on supersets of V . All the assignments inside a team $X \in \text{Teams}$ or hyperteam $\mathfrak{X} \in \text{HypTeams}$ are defined on the same variables, whose sets are indicated by $\text{var}(X)$ and $\text{var}(\mathfrak{X})$, respectively. We shall call the empty set of teams \emptyset the *empty hyperteam*, every set containing the empty team, for instance $\{\emptyset\}$, a *null hyperteam*, and the set $\{\{\emptyset\}\}$ containing a single team comprised only of the empty assignment the *trivial hyperteam*. Essentially, the trivial hyperteam encodes the situation in which none of the players has made any choice yet and, hence, contains the minimal “consistent” state of a game. In this sense, then, null and empty hyperteams do not convey any meaningful information about the possible state of a game and are included here mainly for technical reasons, as they allow for a cleaner formal definition of the semantics. For this reason, we shall refer to every hyperteam which is neither the empty hyperteam nor a null hyperteam with the term *proper hyperteam*.

For any pair of hyperteams $\mathfrak{X}_1, \mathfrak{X}_2 \in \text{HypTeams}$, we write $\mathfrak{X}_1 \sqsubseteq \mathfrak{X}_2$ to state that, for all teams $X_1 \in \mathfrak{X}_1$, there exists a team $X_2 \in \mathfrak{X}_2$ such that $X_2 \subseteq X_1$ (observe that the inclusion of the teams is the reversed of the square inclusion of the hyperteams). We introduce $\mathfrak{X}_1 \equiv \mathfrak{X}_2$ to denote the fact that both $\mathfrak{X}_1 \sqsubseteq \mathfrak{X}_2$ and $\mathfrak{X}_2 \sqsubseteq \mathfrak{X}_1$ hold true. Obviously, $\mathfrak{X}_1 \subseteq \mathfrak{X}_2$ implies $\mathfrak{X}_1 \sqsubseteq \mathfrak{X}_2$, which, in turn, implies $\text{var}(\mathfrak{X}_1) = \text{var}(\mathfrak{X}_2)$. It is clear that the relation \sqsubseteq is both reflexive and transitive, hence it is a preorder; as an immediate consequence, \equiv is an equivalence relation. In particular, we shall show (see Corollary 2.1 later in this section) that \equiv captures the intuitive notion of equivalence between hyperteams, in the sense that two equivalent hyperteams *w.r.t* \equiv do satisfy the same FO formulas. Figure 2.1 provides a graphical representation of the preorder relation \sqsubseteq .

Example 2.1. *In Figure 2.1, the hyperteam \mathfrak{X}_1 is \sqsubseteq -included in the hyperteam \mathfrak{X}_2 , since, for each team X in \mathfrak{X}_1 , there is a team in \mathfrak{X}_2 that is set-included in X . For instance, the team X_1 of \mathfrak{X}_1 contains the assignments $\chi_1, \chi_2, \chi_3, \chi_4$, and χ_5 , so, it includes the team*

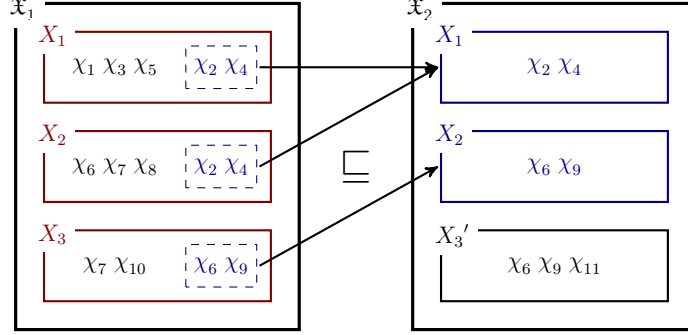


Figure 2.1 – Two hyperteams with $\mathfrak{X}_1 \subseteq \mathfrak{X}_2$, but $\mathfrak{X}_2 \not\subseteq \mathfrak{X}_1$.

X_1 of \mathfrak{X}_2 composed of χ_2 and χ_4 . Note that not all teams in \mathfrak{X}_2 are included in a team in \mathfrak{X}_1 and different teams of \mathfrak{X}_1 can choose the same team of \mathfrak{X}_2 to include.

Given a hyperteam $\mathfrak{X} \in \text{HypTeams}$ and a set of variables $W \subseteq \text{Var}$, we define $\mathfrak{X}_{|W} \stackrel{\text{def}}{=} \{X_{|W} \mid X \in \mathfrak{X}\}$ and $X_{|W} \stackrel{\text{def}}{=} \{\chi_{|W} \mid \chi \in X\}$ recall that $\chi_{|W}$ is the restriction of the assignment χ to the domain $\text{DOM}(\chi) \cap W$. We can, then, compare hyperteams relative to W by writing $\mathfrak{X}_1 =_W \mathfrak{X}_2$ for $\mathfrak{X}_{1|W} = \mathfrak{X}_{2|W}$, meaning that the two hyperteams are indistinguishable when only variables in W are considered. Similarly, $\mathfrak{X}_1 \equiv_W \mathfrak{X}_2$ stands for $\mathfrak{X}_{1|W} \equiv \mathfrak{X}_{2|W}$ and means that they are equivalent on W , while $\mathfrak{X}_1 \sqsubseteq_W \mathfrak{X}_2$ abbreviates $\mathfrak{X}_{1|W} \sqsubseteq \mathfrak{X}_{2|W}$ and relativises the ordering to a dependence constraint. Obviously, $\mathfrak{X}_1 =_W \mathfrak{X}_2$, $\mathfrak{X}_1 \equiv_W \mathfrak{X}_2$, and $\mathfrak{X}_1 \sqsubseteq_W \mathfrak{X}_2$ imply $\mathfrak{X}_1 =_{W'} \mathfrak{X}_2$, $\mathfrak{X}_1 \equiv_{W'} \mathfrak{X}_2$, and $\mathfrak{X}_1 \sqsubseteq_{W'} \mathfrak{X}_2$, respectively, for all $W' \subseteq W$.

Example 2.2. In Figure 2.1, \mathfrak{X}_2 is not \sqsubseteq -included in \mathfrak{X}_1 , as none of the teams of \mathfrak{X}_2 includes a team of \mathfrak{X}_1 . Now, assume the existence of a set of variables W that makes $\{\chi_1, \chi_3, \chi_4, \chi_5, \chi_6, \chi_7, \chi_{10}\}_{|W}$ collapse to $\{\chi_1\}_{|W}$. Then, we have:

$$\begin{array}{c|c}
 \mathfrak{X}_{2|W} & \mathfrak{X}_{1|W} \\
 \hline
 X_{21|W} = \{\chi_{1|W}, \chi_{2|W}\} & X_{11|W} = \{\chi_{1|W}, \chi_{2|W}\} \\
 X_{22|W} = \{\chi_{1|W}, \chi_{9|W}\} & X_{12|W} = \{\chi_{1|W}, \chi_{2|W}, \chi_{8|W}\} \\
 X_{23|W} = \{\chi_{1|W}, \chi_{9|W}, \chi_{11|W}\} & X_{13|W} = \{\chi_{1|W}, \chi_{9|W}\}
 \end{array}$$

Now, team $X_{11|W}$ is included in $X_{21|W}$ and team $X_{13|W}$ is included in both $X_{22|W}$ and $X_{23|W}$. Therefore, $\mathfrak{X}_2 \sqsubseteq_W \mathfrak{X}_1$ and, so, $\mathfrak{X}_1 \equiv_W \mathfrak{X}_2$, since $\mathfrak{X}_1 \sqsubseteq \mathfrak{X}_2$.

The alternating semantics is given by means of a satisfaction relation between a hyperteam \mathfrak{X} and a formula φ , w.r.t a given interpretation of the players in \mathfrak{X} , that is w.r.t

an alternation flag $\alpha \in \{\exists\forall, \forall\exists\}$. As a consequence, we shall introduce two satisfaction relations, $\models^{\exists\forall}$ and $\models^{\forall\exists}$, one for each interpretation of players in the hyperteam. The intuition is that, when the alternation flag α is $\exists\forall$, then a team is chosen existentially by Eloise and all its assignments, chosen universally by Abelard, must satisfy φ . Conversely, when α is $\forall\exists$, then all teams, chosen universally by Abelard, must contain at least one assignments, chosen existentially by Eloise, that satisfies φ .

The definition of the semantics relies on three basic operations on hyperteam: the *dualisation* swaps the role of the two players in a hyperteam, allowing for connecting the two satisfaction relations and a symmetric treatment of quantifiers later on; the *extension* directly handles quantifications; finally, the *partition* deals with disjunction and conjunction.

Let us consider the *dualisation operator* first. Given a hyperteam \mathfrak{X} , the dual hyperteam $\overline{\mathfrak{X}}$ exchanges the role of the two players *w.r.t* \mathfrak{X} . This means that, if Eloise is the player choosing the team in \mathfrak{X} and Abelard the one choosing the assignment in the team, it will be Abelard who chooses the team in $\overline{\mathfrak{X}}$ and Eloise the one who chooses the assignment. To ensure that the semantics of the underlying game is not altered when exchanging the order of choice for the two players, we need to reshuffle the assignments in \mathfrak{X} so as to simulate the original dependencies between the choices. To this end, for a hyperteam \mathfrak{X} , we introduce the set

$$\text{Chc}(\mathfrak{X}) \stackrel{\text{def}}{=} \{\Gamma : \rightarrow \text{Asg} \mid \forall X \in \mathfrak{X}, \Gamma(X) \in X\}$$

of *choice functions*, whose definition implicitly assumes the axiom of choice, whenever the structure domain \mathbb{D} is uncountable. Set $\text{Chc}(\mathfrak{X})$ contains all the functions Γ that, for every team X in \mathfrak{X} , pick a specific assignment $\Gamma(X)$ in that set. Each such function simulates a possible choice of the second player of \mathfrak{X} depending on the choice of (the team chosen by) the first player. The dual hyperteam $\overline{\mathfrak{X}}$, then, collects the images of the choice functions in $\text{Chc}(\mathfrak{X})$. We, thus, obtain a hyperteam in which the choice order of the two players is inverted:

$$\overline{\mathfrak{X}} \stackrel{\text{def}}{=} \{\text{img}(\Gamma) \mid \Gamma \in \text{Chc}(\mathfrak{X})\}.$$

It is immediate to check that the only hyperteam equivalent to the empty or null ones are themselves and they are also dual of one another. Therefore, the class of proper hyperteam is closed under dualisation. In addition, the trivial hyperteam is self-dual.

Proposition 2.1. *For every hyperteam \mathfrak{X} , we have the following.*

1. $\mathfrak{X} \equiv \emptyset$ iff $\mathfrak{X} = \emptyset$ iff $\overline{\mathfrak{X}} = \{\emptyset\}$;
2. $\mathfrak{X} \equiv \{\emptyset\}$ iff $\emptyset \in \mathfrak{X}$ iff $\overline{\mathfrak{X}} = \emptyset$. Moreover,
3. $\overline{\{\{\emptyset\}\}} = \{\{\emptyset\}\}$. Finally,
4. \mathfrak{X} is proper iff $\overline{\mathfrak{X}}$ is proper as well.

Example 2.3. Consider the following two dual hypertteams

$$\mathfrak{X} = \left\{ \begin{array}{l} X = \{\chi_{11}, \chi_{12}\}, \\ X = \{\chi_{21}, \chi_{22}\}, \\ X = \{\chi_3\} \end{array} \right\} \quad \text{and} \quad \overline{\mathfrak{X}} = \left\{ \begin{array}{l} \text{img}(\Gamma_1) = \{\chi_{11}, \chi_{21}, \chi_3\}, \\ \text{img}(\Gamma_2) = \{\chi_{11}, \chi_{22}, \chi_3\}, \\ \text{img}(\Gamma_3) = \{\chi_{12}, \chi_{21}, \chi_3\}, \\ \text{img}(\Gamma_4) = \{\chi_{12}, \chi_{22}, \chi_3\} \end{array} \right\},$$

where the teams of \mathfrak{X} are $X = \{\chi_{11}, \chi_{12}\}$, $X = \{\chi_{21}, \chi_{22}\}$, and $X = \{\chi_3\}$. Every team in $\overline{\mathfrak{X}}$ is obtained as the image of one of the four choice functions $\Gamma_i \in \text{Chc}(\mathfrak{X})$, each choosing exactly one assignment from X , one from X , and the unique one from X . Intuitively, in \mathfrak{X} the strategy of the first player, say Eloise, can only choose the colour of the final assignments (either red for X , blue for X , or green for X), while the one for Abelard decides which assignment of each colour will be picked. After dualisation, the two players exchange the order in which they choose. Therefore, Abelard, starting first in $\overline{\mathfrak{X}}$, will select one of the four choice functions, which picks an assignment for each colour. Eloise, choosing second, by using her strategy that selects the colour will give the final assignment. In other words, the original strategies of the players encoded in the hypertteam, as well as their dependencies, are preserved, regardless of the swap of their role in the dual hypertteam. The example also shows that, as we shall prove shortly (see Theorem 2.2 later in this section), if we dualise a hypertteam \mathfrak{X} and, at the same time, swap the original interpretation $\alpha \in \{\exists\forall, \forall\exists\}$ of the player to $\overline{\alpha}$, we obtain that the pair $(\overline{\mathfrak{X}}, \overline{\alpha})$ gives an equivalent representation of the information contained in the original pair (\mathfrak{X}, α) .

Dualisation enjoys an *involution property* similar to the classic Boolean negation: by applying the dualisation twice, we obtain a hypertteam equivalent to the original one. This confirms that the operation preserves the entire information encoded in the hypertteams.

Lemma 2.1 (Dualisation I). *For all hypertteams $\mathfrak{X} \in \text{HypTeams}$ and set of variables W , it holds that $\mathfrak{X} \equiv_W \overline{\overline{\mathfrak{X}}}$. In addition, $\mathfrak{X} \subseteq \overline{\overline{\mathfrak{X}}}$, if \mathfrak{X} is proper.*

The proof of this lemma, together with those of all the non-trivial results, can be found in appendix of this section.

Observe the clear analogy between the structure of hyperteams with alternation flag $\exists\forall$ (*resp.*, $\forall\exists$) and the structure of DNF (*resp.*, CNF) Boolean formulas, where the dualisation swaps between two equivalent forms. The following lemma formally states that this operation swaps the role of the two players, while still preserving the original dependencies among their choices.

Lemma 2.2 (Dualisation II). *The following equivalences hold true, for all hyperteams $\mathfrak{X} \in \text{HypTeams}$ and properties $\Psi \subseteq \text{Asg}$.*

1. *Statements 1a and 1b are equivalent:*

(a) *there exists a team $X \in \mathfrak{X}$ (*resp.*, $X \in \overline{\mathfrak{X}}$) such that $X \subseteq \Psi$;*

(b) *for all teams $X' \in \overline{\mathfrak{X}}$ (*resp.*, $X' \in \mathfrak{X}$), it holds that $X' \cap \Psi \neq \emptyset$.*

2. *Statements 2a and 2b are equivalent:*

(a) *there exists a team $X \in \mathfrak{X}$ such that $X \cap \Psi \neq \emptyset$;*

(b) *there exists a team $X' \in \overline{\mathfrak{X}}$ such that $X' \cap \Psi \neq \emptyset$.*

3. *Statements 3a and 3b are equivalent:*

(a) *for all teams $X \in \mathfrak{X}$, it holds that $X \subseteq \Psi$;*

(b) *for all teams $X' \in \overline{\mathfrak{X}}$, it holds that $X' \subseteq \Psi$.*

Item 1 provides the semantic meaning of the operation, stating that if there exists a team in \mathfrak{X} all of whose assignments satisfy some property Ψ , then each team in $\overline{\mathfrak{X}}$ has an assignment satisfying the property, and *vice versa*. This directly connects the two interpretations of hyperteams, $\forall\exists$ and $\exists\forall$. Item 2 establishes that no assignment is lost from the original teams in \mathfrak{X} , while Item 3 asserts that no new assignments are added to $\overline{\mathfrak{X}}$. It could be proved that any two operators that satisfies the three conditions in the lemma will produce equivalent hyperteams, in the sense of \equiv , when applied to the same hyperteam.

Quantifications are taken care of by the *extension operator*. The extension of a hyperteam $\mathfrak{X} \in \text{HypTeams}$ with x is simply the set of extensions with x of all its teams by all possible response:

$$\text{EXT}(\mathfrak{X}, x) \stackrel{\text{def}}{=} \{\text{EXT}(X, F, x) \mid X \in \mathfrak{X}, F \in \text{Rsp}\}.$$

The extension operation essentially embeds into \mathfrak{X} all possible (W -uniform) strategies for choosing the value of x , each one encoded by a function F in Rsp .

Example 2.4. Let $\mathfrak{X} = \{X_1 = \{\chi_1, \chi_2\}, X_2 = \{\chi_1, \chi_3\}\}$ be a hyperteam. To extend \mathfrak{X} with variable x over the structure domain $\mathbb{D} = \{0, 1\}$, one needs to extend each team in \mathfrak{X} with the eight possible response functions: F_{ijk} that maps χ_1 to i and χ_2 to j and χ_3 to k for $i, j, k \in \{0, 1\}$.

$$\text{EXT}(\mathfrak{X}, x) = \left\{ \begin{array}{l} \text{EXT}(X_1, F_{000}, x) = \{\chi_1[x \mapsto 0], \chi_2[x \mapsto 0]\}, \\ \text{EXT}(X_1, F_{000}, x) = \{\chi_1[x \mapsto 0], \chi_3[x \mapsto 0]\}, \\ \text{EXT}(X_1, F_{001}, x) = \{\chi_1[x \mapsto 0], \chi_2[x \mapsto 0]\}, \\ \text{EXT}(X_1, F_{001}, x) = \{\chi_1[x \mapsto 0], \chi_3[x \mapsto 1]\}, \\ \vdots \\ \text{EXT}(X_1, F_{ijk}, x) = \{\chi_1[x \mapsto i], \chi_2[x \mapsto j]\}, \\ \text{EXT}(X_1, F_{ijk}, x) = \{\chi_1[x \mapsto i], \chi_3[x \mapsto k]\}, \\ \vdots \\ \text{EXT}(X_1, F_{111}, x) = \{\chi_1[x \mapsto 1], \chi_2[x \mapsto 1]\}, \\ \text{EXT}(X_1, F_{111}, x) = \{\chi_1[x \mapsto 1], \chi_3[x \mapsto 1]\} \end{array} \right\}$$

Notice that multiple teams collapse, for instance $\text{EXT}(X_1, F_{000}, x) = \text{EXT}(X_1, F_{001}, x)$.

Conjunctions and disjunctions are dealt with by means of the *partition operator*. We provide here the intuition for disjunction, the dual reasoning applies to conjunction. Assume that the two players of \mathfrak{X} , defined over the variables $\{x, y\}$, are interpreted according to the alternation flag $\forall\exists$: Abelard chooses the team and Eloise chooses the assignment in the team. In our setting, then, in order to satisfy, for instance, $(x = 0) \vee (x = 1)$, Eloise has to show that, for each team $X \in \mathfrak{X}$ chosen by Abelard, she has a way to select one of the disjuncts $x = 0$ or $x = 1$, so that the given team has an assignment satisfying the disjunct. To capture Eloise's choice of disjunct based on the team given by Abelard, we define, for a hyperteam \mathfrak{X} , the set $\text{PART}(\mathfrak{X})$ that collects all the possible bipartitions of \mathfrak{X} .

$$\text{PART}(\mathfrak{X}) \stackrel{\text{def}}{=} \{(\mathfrak{X}_1, \mathfrak{X}_2) \in 2^{\mathfrak{X}} \times 2^{\mathfrak{X}} \mid \mathfrak{X}_1 \cap \mathfrak{X}_2 = \emptyset \wedge \mathfrak{X}_1 \cup \mathfrak{X}_2 = \mathfrak{X}\},$$

Intuitively, the hyperteam \mathfrak{X}_1 will be used to satisfy $x = 0$, while \mathfrak{X}_2 will be used for $x = 1$. Basically, $\text{PART}(\mathfrak{X})$ contains all the possible strategies by means of which Eloise can try to satisfy the two disjuncts. Then, we say that Eloise satisfies the disjunction if there is a pair $(\mathfrak{X}'_1, \mathfrak{X}'_2)$ (hence, a hyperteam-partition strategy) in that set such that \mathfrak{X}'_1 satisfies the left disjunct and \mathfrak{X}'_2 satisfies the right one.

2.2 A new semantical relation

We assume a signature \mathcal{S} without function symbols, so that we do not have to bother with terms that are not tuples of variables. The compositional hyperteam semantics of an FO formula can be, then, defined as follows.

Definition 2.1 (Hyperteam Semantics). *The hyperteam semantic relation $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$ for FO is inductively defined as follows, for all FO formulas φ , hyperteam $\mathfrak{X} \in \text{HypTeams}_{\subseteq}(\text{FREE}(\varphi))$, tuples of variables \vec{x} , FO predicates P and alternation flags $\alpha \in \{\exists\forall, \forall\exists\}$:*

1. (a) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \perp$ if $\emptyset \in \mathfrak{X}$;
 (b) $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \perp$ if $\mathfrak{X} = \emptyset$;
2. (a) $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \top$ if $\emptyset \notin \mathfrak{X}$;
 (b) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \top$ if $\mathfrak{X} \neq \emptyset$;
3. (a) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} P(\vec{x})$ if there exists a team $X \in \mathfrak{X}$ such that, for all assignments $\chi \in X$, it holds that $\chi(\vec{x}) \in P^{\mathcal{M}}$;
 (b) $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} P(\vec{x})$ if, for all teams $X \in \mathfrak{X}$, there exists an assignment $\chi \in X$ such that $\chi(\vec{x}) \in P^{\mathcal{M}}$;
4. $\mathcal{M}, \mathfrak{X} \models^\alpha \neg\phi$ if $\mathcal{M}, \mathfrak{X} \not\models^{\bar{\alpha}} \phi$;
5. (a) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \phi_1 \wedge \phi_2$ if, for all bipartitions $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$, it holds that $\mathcal{M}, \mathfrak{X}_1 \models^{\exists\forall} \phi_1$ or $\mathcal{M}, \mathfrak{X}_2 \models^{\exists\forall} \phi_2$;
 (b) $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \phi_1 \wedge \phi_2$ if $\mathcal{M}, \bar{\mathfrak{X}} \models^{\exists\forall} \phi_1 \wedge \phi_2$;
6. (a) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \phi_1 \vee \phi_2$ if $\mathcal{M}, \bar{\mathfrak{X}} \models^{\forall\exists} \phi_1 \vee \phi_2$;
 (b) $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \phi_1 \vee \phi_2$ if there exists a bipartition $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that $\mathcal{M}, \mathfrak{X}_1 \models^{\forall\exists} \phi_1$ and $\mathcal{M}, \mathfrak{X}_2 \models^{\forall\exists} \phi_2$;
7. (a) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \exists x\phi$ if $\mathcal{M}, \text{EXT}(\mathfrak{X}, x) \models^{\exists\forall} \phi$;
 (b) $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \exists x\phi$ if $\mathcal{M}, \bar{\mathfrak{X}} \models^{\exists\forall} \exists x\phi$;
8. (a) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \forall x\phi$ if $\mathcal{M}, \bar{\mathfrak{X}} \models^{\forall\exists} \forall x\phi$;
 (b) $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \forall x\phi$ if $\mathcal{M}, \text{EXT}(\mathfrak{X}, x) \models^{\forall\exists} \phi$.

Items 1 and 2 take care of the Boolean constants, requiring, for instance, \top to be satisfied by all hyperteam, except for the empty one, under the $\exists\forall$ interpretation, and the null one, under $\forall\exists$. A dual reasoning applies to \perp . The other base case for atomic

formulas, Item 3, is trivial and follows the interpretation of the alternation flag. Negation, in accordance with the classic game-theoretic interpretation, is dealt with by Item 4 by exchanging the interpretation of the players of the hyperteam. The semantics of the remaining Boolean connectives (Items 5 and 6) and quantifiers (Items 7 and 8) is a direct application of the partition and extension operators previously defined. Observe that swapping between $\models^{\exists\forall}$ and $\models^{\forall\exists}$ (Items 5b, 6a, 7b and 8a) is done according to Lemma 2.2 and represents the fundamental point where our approach departs from team semantics.

Remark 2.1. *An alternative option for the semantics of Boolean connectives is to use coverings instead of partitions, i.e., pairs of hyperteam $(\mathfrak{X}_1, \mathfrak{X}_2)$ such that $\mathfrak{X}_1 \cup \mathfrak{X}_2 = \mathfrak{X}$. However, from a covering $(\mathfrak{X}_1, \mathfrak{X}_2)$, one can extract the partition $(\mathfrak{X}_1, \mathfrak{X}_2 \setminus \mathfrak{X}_1)$, where $\mathfrak{X}_2 \setminus \mathfrak{X}_1 \sqsubseteq \mathfrak{X}_2$. Then, an application of Theorem 2.1 below would allow to immediately conclude on the equivalence of the two semantics.*

For every FO formula φ and alternation flag $\alpha \in \{\exists\forall, \forall\exists\}$, we say that φ is α -satisfiable in \mathcal{M} , in symbols $\mathcal{M} \models^\alpha \varphi$, if there exists a proper hyperteam $\mathfrak{X} \in \text{HypTeams}_{\geq}(\text{FREE}(\varphi))$ such that $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$. As already mentioned before, here we are not considering the empty and null hyperteam as potential hyperteam, since these do not convey meaningful information. We simply say that φ is α -satisfiable iff it is α -satisfiable in some structure \mathcal{M} . Also, φ α -implies (resp., is α -equivalent to) an FO formula ϕ in \mathcal{M} , in symbols $\varphi \Rightarrow_{\mathcal{M}}^\alpha \phi$ (resp., $\varphi \equiv_{\mathcal{M}}^\alpha \phi$), whenever $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$ implies $\mathcal{M}, \mathfrak{X} \models^\alpha \phi$ (resp., $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$ iff $\mathcal{M}, \mathfrak{X} \models^\alpha \phi$), for all $\mathfrak{X} \in \text{HypTeams}_{\geq}(\text{FREE}(\varphi) \cup \text{FREE}(\phi))$. If the implication (resp., equivalence) holds for all structures \mathcal{M} , we just state that φ α -implies (resp., is α -equivalent to) ϕ , in symbols $\varphi \Rightarrow^\alpha \phi$ (resp., $\varphi \equiv^\alpha \phi$). Finally, we say that φ is *satisfiable* if it is both $\exists\forall$ - and $\forall\exists$ -satisfiable, and φ *implies* (resp., is *equivalent* to) ϕ , in symbols $\varphi \Rightarrow \phi$ (resp., $\varphi \equiv \phi$), if both $\varphi \Rightarrow^{\exists\forall} \phi$ and $\varphi \Rightarrow^{\forall\exists} \phi$ (resp., $\varphi \equiv^{\exists\forall} \phi$ and $\varphi \equiv^{\forall\exists} \phi$) hold true. These notions of satisfiable formulas and of implication are justified by Theorem 2.2 that makes $\exists\forall$ - and $\forall\exists$ -satisfiable collapse to simply satisfiable and $\exists\forall$ - and $\forall\exists$ -implication to just implication.

2.3 Fundamentals

We show that the hyperteam semantics enjoys several classic properties, such as *Boolean laws* and the canonical representation for formulas in *negation normal form* (nnf, for short), that are usually expected to hold for a logic closed under negation.

We start with the following very basic result, characterising the truth of formulas over the null and empty hyperteams.

Lemma 2.3 (Empty & Null Hyperteams). *The following hold true for every FO formula φ and hyperteam $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi))$:*

1. (a) $\mathcal{M}, \emptyset \not\models^{\exists\forall} \varphi$;
 (b) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$, where $\emptyset \in \mathfrak{X}$;
2. (a) $\mathcal{M}, \emptyset \models^{\forall\exists} \varphi$;
 (b) $\mathcal{M}, \mathfrak{X} \not\models^{\forall\exists} \varphi$, where $\emptyset \in \mathfrak{X}$.

The preorder \sqsubseteq on hyperteams introduced above captures the intuitive notion of satisfaction strength *w.r.t* FO formulas. Basically, if $\mathfrak{X}_1 \sqsubseteq \mathfrak{X}_2$, the hyperteam \mathfrak{X}_1 satisfies, *w.r.t* the $\exists\forall$ (*resp.*, $\forall\exists$) semantic relation, less (*resp.*, more) formulas than the hyperteam \mathfrak{X}_2 . Actually, a stronger version of this property holds, when the \sqsubseteq -preorder is restricted to the set of free variables of the formula. This property is trivial for atomic formulas and can easily be proved by structural induction for the non-atomic ones.

Theorem 2.1 (Hyperteam Refinement). *Let φ be an FO formula and $\mathfrak{X}, \mathfrak{X}' \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi))$ two hyperteams with $\mathfrak{X} \sqsubseteq_{\text{FREE}(\varphi)} \mathfrak{X}'$. Then:*

1. if $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$ then $\mathcal{M}, \mathfrak{X}' \models^{\exists\forall} \varphi$;
2. if $\mathcal{M}, \mathfrak{X}' \models^{\forall\exists} \varphi$ then $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi$.

As an immediate consequence, we obtain the following result.

Corollary 2.1 (Hyperteam Equivalence). *Let φ be an FO formula and $\mathfrak{X}, \mathfrak{X}' \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi))$ two hyperteams with $\mathfrak{X} \equiv_{\text{FREE}(\varphi)} \mathfrak{X}'$. Then:*

$$\mathcal{M}, \mathfrak{X} \models^{\alpha} \varphi \text{ iff } \mathcal{M}, \mathfrak{X}' \models^{\alpha} \varphi.$$

Since, by definition, an FO sentence φ satisfies $\text{FREE}(\varphi) = \emptyset$, we can test its truth by just looking at its satisfaction *w.r.t* the trivial hyperteam $\{\{\emptyset\}\}$, as every proper hyperteam is equivalent to $\{\{\emptyset\}\}$ on the empty set of variables.

Corollary 2.2 (Sentence Satisfiability). *Let φ be an FO sentence. Then, φ is α -satisfiable iff $\mathcal{M}, \{\{\emptyset\}\} \models^{\alpha} \varphi$, for some \mathcal{S} -structure \mathcal{M} .*

As mentioned in Example 2.3, swapping the players of a hyperteam \mathfrak{X} , i.e., switching the alternation flag, and swapping the choices of the players, i.e., dualising \mathfrak{X} , have the same effect as far as satisfaction is concerned. Recall in addition that, by Lemma 2.1, the dualisation enjoys the involution property. Consequently, dualising both the alternation flag α and the hyperteam \mathfrak{X} preserves truth of formulas. These observations are formalised by the following result.

Theorem 2.2 (Double Dualisation). *For every FO formula φ , alternation flag α and hyperteam $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi))$, it holds that $\mathcal{M}, \mathfrak{X} \models^{\alpha} \varphi$ iff $\mathcal{M}, \overline{\mathfrak{X}} \models^{\alpha} \varphi$ iff $\mathcal{M}, \overline{\overline{\mathfrak{X}}} \models^{\overline{\alpha}} \varphi$.*

The above property also grants that formulas satisfiability, implication, and equivalence do not depend on the specific interpretation α of hyperteam semantics: a positive answer for α implies the same for $\overline{\alpha}$. This *invariance* corresponds to the intuition that the truth of a sentence, as well as the concept of logical consequence and equivalence, do not depend on the point of view of the specific player. One can also see this as a consequence of the symmetric treatment of Eloise and Abelard in the semantics.

Corollary 2.3 (Interpretation Invariance). *Let φ and ϕ be FO formulas. Then, φ is $\exists\forall$ -satisfiable iff φ is $\forall\exists$ -satisfiable. Also, $\varphi \Rightarrow^{\exists\forall} \phi$ iff $\varphi \Rightarrow^{\forall\exists} \phi$ and $\varphi \equiv^{\exists\forall} \phi$ iff $\varphi \equiv^{\forall\exists} \phi$.*

Given the game-theoretic nature of hyperteam semantics and negation, hyperteam semantics does not naturally entail *logical determinacy*, i.e., the property stating that a model either satisfies a formula or its negation, *w.r.t* the same semantic relation. However, it satisfies the *game-theoretic determinacy* stated below, which corresponds to the following intuition: if a player cannot prove the truth of a formula, then the other player can prove the truth of its negation.

Corollary 2.4 (Game-Theoretic Determinacy). *Let φ be an FO formula, α an alternation flag and $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi))$ a hyperteam. Then:*

1. *either $\mathcal{M}, \mathfrak{X} \models^{\alpha} \varphi$ or $\mathcal{M}, \mathfrak{X} \models^{\overline{\alpha}} \neg\varphi$;*
2. *either $\mathcal{M}, \mathfrak{X} \models^{\alpha} \varphi$ or $\mathcal{M}, \overline{\mathfrak{X}} \models^{\alpha} \neg\varphi$.*

Since, as observed above, the truth of sentences can be tested against the trivial hyperteam $\{\{\emptyset\}\}$, regardless of the specific alternation flag α , the classic *law of excluded middle* does hold at least for all FO sentences. In the following, we denote with $\mathcal{M} \models \varphi$ the fact that a sentence φ is both $\exists\forall$ -satisfied and $\forall\exists$ -satisfied by $\{\{\emptyset\}\}$ in \mathcal{M} .

Corollary 2.5 (Law of Excluded Middle). *Let φ be an FO sentence. Then, either $\mathcal{M} \models \varphi$ or $\mathcal{M} \models \neg\varphi$.*

Thanks to the above properties, we can establish the following elementary Boolean laws, which, in turn, allow for a canonical representation of formulas in nnf, as stated in Corollary 2.6.

Theorem 2.3 (Boolean Laws). *Let φ_1, φ_2 and φ be FO formulas. Then:*

1. (a) $\neg\perp \equiv \top$;
 (b) $\neg\top \equiv \perp$;
 (c) $\varphi \equiv \neg\neg\varphi$;
2. (a) $\varphi \wedge \perp \equiv \perp \wedge \varphi \equiv \perp$;
 (b) $\varphi \wedge \top \equiv \top \wedge \varphi \equiv \varphi$;
3. (a) $\varphi \vee \top \equiv \top \vee \varphi \equiv \top$;
 (b) $\varphi \vee \perp \equiv \perp \vee \varphi \equiv \varphi$;
4. (a) $\varphi_1 \wedge \varphi_2 \equiv \varphi_2 \wedge \varphi_1$;
 (b) $\varphi_1 \vee \varphi_2 \equiv \varphi_2 \vee \varphi_1$;
5. (a) $\varphi_1 \wedge \varphi_2 \Rightarrow \varphi_1$;
 (b) $\varphi_1 \wedge (\varphi \wedge \varphi_2) \equiv (\varphi_1 \wedge \varphi) \wedge \varphi_2$;
6. (a) $\varphi_1 \Rightarrow \varphi_1 \vee \varphi_2$;
 (b) $\varphi_1 \vee (\varphi \vee \varphi_2) \equiv (\varphi_1 \vee \varphi) \vee \varphi_2$;
7. (a) $\varphi_1 \wedge \varphi_2 \equiv \neg(\neg\varphi_1 \vee \neg\varphi_2)$;
 (b) $\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2)$;
8. (a) $\exists x\varphi \equiv \neg(\forall x\neg\varphi)$;
 (b) $\forall x\varphi \equiv \neg(\exists x\neg\varphi)$.

Corollary 2.6 (Negation Normal Form). *Every FO formula is equivalent to an FO formula in nnf for the hyperteam semantics.*

For technical convenience, we shall now generalise the extension operator to *quantifier prefixes* \wp (finite sequences of $\exists x$ or $\forall x$), whose set is denoted by \mathbf{Q} . Notice that, *without loss of generality*, we only consider prefixes where each variable x is quantified at most once. With $\text{VAR}(\wp)$, we denote the set of variables quantified in \wp . Given a hyperteam

\mathfrak{X} and an alternation flag α , the operator $\text{EXT}_\alpha(\mathfrak{X}, \wp)$ corresponds to iteratively applying the extension operator to \mathfrak{X} , for all quantifiers occurring in \wp , in that specific order. To this end, we first introduce the notion of *coherence* of a quantifier symbol $Q \in \{\exists, \forall\}$ with an alternation flag $\alpha \in \{\exists\forall, \forall\exists\}$ as follows: Q is α -coherent if either $\alpha = \exists\forall$ and $Q = \exists$ or $\alpha = \forall\exists$ and $Q = \forall$. Now, the application of a quantifier Qx to \mathfrak{X} , denoted by $\text{EXT}_\alpha(\mathfrak{X}, Qx)$, follows the semantics of quantifiers, as defined in Items 7 and 8 of Definition 2.1. More precisely, it just corresponds to the extension of \mathfrak{X} with x , when Q is α -coherent. Conversely, when Q is $\bar{\alpha}$ -coherent, we need to dualise the extension with x of the dual of \mathfrak{X} . Formally:

$$\text{EXT}_\alpha(\mathfrak{X}, Qx) \stackrel{\text{def}}{=} \begin{cases} \text{EXT}(\mathfrak{X}, x), & \text{if } Q \text{ is } \alpha\text{-coherent;} \\ \overline{\text{EXT}(\overline{\mathfrak{X}}, x)}, & \text{otherwise.} \end{cases}$$

The operator naturally lifts to arbitrary quantification prefixes \wp :

1. $\text{EXT}_\alpha(\mathfrak{X}, \epsilon) \stackrel{\text{def}}{=} \mathfrak{X}$;
2. $\text{EXT}_\alpha(\mathfrak{X}, Qx\wp) \stackrel{\text{def}}{=} \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, Qx), \wp)$.

We also define $\text{EXT}_\alpha(\wp) \stackrel{\text{def}}{=} \text{EXT}_\alpha(\{\{\emptyset\}\}, \wp)$. A simple structural induction on a quantifier prefix $\wp \in \mathbf{Q}$, shows that a hyperteam \mathfrak{X} α -satisfies a formula $\wp\phi$ iff its α -extension *w.r.t* \wp α -satisfies its subformula ϕ .

Theorem 2.4 (Prefix Extension). *Let $\wp\phi$ be an FO formula, where $\wp \in \mathbf{Q}$ is a quantifier prefix and ϕ is an arbitrary FO formula. Then, $\mathcal{M}, \mathfrak{X} \models^\alpha \wp\phi$ iff $\mathcal{M}, \text{EXT}_\alpha(\mathfrak{X}, \wp) \models^\alpha \phi$, for all hyperteam $\mathfrak{X} \in \text{HypTeams}_\supseteq(\text{FREE}(\wp\phi))$.*

2.4 Adequacy

In this section, we show that the hyperteam semantics is *adequate*, i.e., it is a *conservative extension*, precisely capturing Tarski's satisfaction for FO.

Formally, we prove that Definition 2.1 corresponds to the classic Tarskian satisfaction of Definition 1.3. This *FO adequacy* property holds trivially for atomic formulas and, in order to extend it to the remaining FO components, we make use of the following three lemmata, which take care of dualisation, quantifiers, and binary Boolean connectives, respectively.

As extensively discussed before, the dualisation swaps the role of the two players, while still preserving the original dependencies among their choices. Indeed, if a FO property is satisfied by a hyperteam *w.r.t* a given alternation flag, it is satisfied by its dual version *w.r.t* the dual flag, as formally stated in the lemma below (recall that \models_{FO} denotes the usual FO semantic relation).

Lemma 2.4 (FO Dualisation). *The following equivalences hold, for all FO formulas φ and hyperteams $\mathfrak{X} \in \text{HypTeams}_{\sqsupseteq}(\text{FREE}(\varphi))$.*

1. *Statements 1a and 1b are equivalent:*

- (a) *there exists a team $X \in \mathfrak{X}$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$, for all assignments $\chi \in X$;*
- (b) *for all teams $X \in \overline{\mathfrak{X}}$, there exists an assignment $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$.*

2. *Statements 2a and 2b are equivalent:*

- (a) *for all teams $X \in \mathfrak{X}$, there exists an assignment $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$;*
- (b) *there exists a team $X \in \overline{\mathfrak{X}}$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$, for all assignments $\chi \in X$.*

The following lemma states that the extension operator provides an adequate semantics for classic FO quantifications. Statement 1 considers Eloise’s choices, when the interpretation of the hyperteam is $\exists\forall$, while Statement 2 takes care of Abelard’s choices, when the interpretation is the dual $\forall\exists$.

Lemma 2.5 (FO Quantifiers). *The following equivalences hold, for all FO formulas φ , variables $x \in \text{Var}$, and hyperteams $\mathfrak{X} \in \text{HypTeams}_{\sqsupseteq}(\text{FREE}(\varphi) \setminus \{x\})$.*

1. *Statements 1a and 1b are equivalent:*

- (a) *there exists a team $X \in \mathfrak{X}$ such that $\mathcal{M}, \chi \models_{\text{FO}} \exists x. \varphi$, for all $\chi \in X$;*
- (b) *there exists a team $X \in \text{EXT}(\mathfrak{X}, x)$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$, for all $\chi \in X$.*

2. *Statements 2a and 2b are equivalent:*

- (a) *for all teams $X \in \mathfrak{X}$, there exists $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \forall x. \varphi$;*
- (b) *for all teams $X \in \text{EXT}(\mathfrak{X}, x)$, there exists $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$.*

Finally, the partition operator precisely mimics the semantics of the binary Boolean connectives when the correct interpretation of the underlying hyperteam is considered.

Lemma 2.6 (FO Boolean Connectives). *The following equivalences hold, for all FO formulas φ_1 and φ_2 and hyperteams $\mathfrak{X} \in \text{HypTeams}_{\sqsupseteq}(\text{FREE}(\varphi_1) \cup \text{FREE}(\varphi_2))$.*

1. Statements 1a and 1b are equivalent:

- (a) there exists a team $X \in \mathfrak{X}$ such that $\mathcal{M}, \chi \models_{FO} \varphi_1 \wedge \varphi_2$, for all $\chi \in X$;
- (b) for each bipartition $(\mathfrak{X}_1, \mathfrak{X}_2) \in PART(\mathfrak{X})$, there exist an index $i \in \{1, 2\}$ and a team $X \in \mathfrak{X}_i$ such that $\mathcal{M}, \chi \models_{FO} \varphi_i$, for all $\chi \in X$.

2. Statements 2a and 2b are equivalent:

- (a) for all teams $X \in \mathfrak{X}$, there exists $\chi \in X$ such that $\mathcal{M}, \chi \models_{FO} \varphi_1 \vee \varphi_2$;
- (b) there exists a bipartition $(\mathfrak{X}_1, \mathfrak{X}_2) \in PART(\mathfrak{X})$ such that, for all indexes $i \in \{1, 2\}$ and teams $X \in \mathfrak{X}_i$, it holds that $\mathcal{M}, \chi \models_{FO} \varphi_i$, for some $\chi \in X$.

We can now state the FO adequacy property for hypertteam semantics.

Theorem 2.5 (FO Adequacy). *For all FO formulas φ and hypertteams $\mathfrak{X} \in \text{HypTeams}_{\subseteq}(FREE(\varphi))$, it holds that:*

- 1. $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$ iff there exists a team $X \in \mathfrak{X}$ such that, for all assignments $\chi \in X$, it holds that $\mathcal{M}, \chi \models_{FO} \varphi$;
- 2. $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi$ iff, for all teams $X \in \mathfrak{X}$, there exists an assignment $\chi \in X$ such that $\mathcal{M}, \chi \models_{FO} \varphi$.

2.5 Skolemisation with hypertteams

We now introduce a *meta-level interpretation* of the quantifiers by means of a Herbrand-Skolem semantics extending the compositional one based on hypertteams, which results to be essential for many results in the next chapters.

The game-theoretic interpretation of the quantifiers $\exists x$ and $\forall x$ implicitly identifies strategies for Eloise and Abelard. The *meta extension* we propose here makes these strategies explicit, by augmenting the logic with the two quantifiers, $\Sigma^{+W}x$ and $\Pi^{+W}x$ ranging over W -uniform response functions (that can be seen as Herbrand/Skolem function [Bus98a]). The set W collects the variables of the hypertteam and the variables quantified before x . Intuitively, $\Sigma^{+W}x. \varphi$ ensures the existence of a W -uniform response function (as a Skolem function) assigning to x values that satisfy φ , while $\Pi^{+W}x. \varphi$ verifies φ , for all values assigned to x by some W -uniform response function (as a Herbrand function).

Definition 2.2 (Meta Syntax). *The Meta Extension (Meta, for short) is the set of formulas built according to Definition 1.2 extended as follows, where $x \in V$:*

$$\varphi \stackrel{\text{def}}{=} FO \mid \Sigma^{+W}x. \varphi \mid \Pi^{+W}x. \varphi.$$

To keep track of the Herbrand/Skolem functions already quantified, we use a *function assignment* $\mathcal{F} \in \text{FunAsg} \stackrel{\text{def}}{=} V \rightarrow \text{Rsp}$ mapping each variable $x \in V \stackrel{\text{def}}{=} \text{DOM}(\mathcal{F})$ to a function $\mathcal{F}(x) \in \text{Rsp}$. To extend a hyperteam $\mathfrak{X} \in \text{HypTeams}(U)$ with \mathcal{F} , we make use of the *extension operator* $\text{EXT}(\mathfrak{X}, \mathcal{F}) \stackrel{\text{def}}{=} \{\text{EXT}(X, \mathcal{F}) \mid X \in \mathfrak{X}\}$, where

1. $\text{EXT}(X, \mathcal{F}) \stackrel{\text{def}}{=} \{\chi \in \text{CYL}(X, V) \mid \forall x \in V \setminus U, \chi(x) = \mathcal{F}(x)(\chi)\}$ is the extension of the team X over the variables in V , so that the value $\chi(x)$ given by an assignment χ to each (not yet assigned) variable $x \in V \setminus U$ is coherent with the one prescribed by $\mathcal{F}(x)$ and
2. $\text{CYL}(X, V) \stackrel{\text{def}}{=} \{\chi \in \text{Asg}(U \cup V) \mid \chi|_U \in X\}$ is the cylindrification of a team $X \in \text{Teams}(U)$ w.r.t the set of variables $V \setminus U$.

Finally, a function assignment $\mathcal{F} \in \text{FunAsg}$ is *acyclic* if there is an *acyclic dependency context* $\iota \in V \rightarrow 2^V$, with $\text{DOM}(\mathcal{F}) \subseteq \text{DOM}(\iota)$, such that $\mathcal{F}(x) \in \text{Rsp}_{+\iota(x)}$ for all variables $x \in \text{DOM}(\mathcal{F})$, where by *dependency context* we mean any partial function $\iota \in \text{Var} \rightarrow 2^{\text{Var}}$ (intuitively, a dependency context keep in memory the dependencies of each variables).

Definition 2.3 (Meta Semantics). *The hyperteam semantic relation $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \varphi$ for Meta is inductively defined as follows, for all Meta formulas φ , function assignments $\mathcal{F} \in \text{FunAsg}$, hyperteams $\mathfrak{X} \in \text{HypTeams}_{\geq}(\text{FREE}(\varphi) \setminus \text{DOM}(\mathcal{F}))$, and alternation flags $\alpha \in \{\exists\forall, \forall\exists\}$:*

- 1,2,4-8) All FO cases, but those ones of the atomic relations, are defined by lifting, in the obvious way, the corresponding items of Definition 2.1 to function assignments, i.e., the latter play no role;
3. (a) $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\exists\forall} P(\vec{x})$ if there exists a team $X \in \text{EXT}(\mathfrak{X}, \mathcal{F})$ such that, for all assignments $\chi \in X$, it holds that $\chi(\vec{x}) \in P^{\mathcal{M}}$;
- (b) $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\forall\exists} P(\vec{x})$ if, for all teams $X \in \text{EXT}(\mathfrak{X}, \mathcal{F})$, there exists an assignment $\chi \in X$ such that $\chi(\vec{x}) \in P^{\mathcal{M}}$;
9. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \Sigma^{+W} x. \phi$ if $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models_{\text{Meta}}^\alpha \phi$, for some $F \in \text{Rsp}_{+W}$;
10. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \Pi^{+W} x. \phi$ if $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models_{\text{Meta}}^\alpha \phi$, for all $F \in \text{Rsp}_{+W}$.

Essentially, to evaluate an atomic formula $P(\vec{x})$, we extend \mathfrak{X} with the functions dictated by \mathcal{F} and then we check the assignments following the alternation given by the flag $\alpha \in \{\forall\exists, \exists\forall\}$ as in plain FO. Indeed, Item 3 above can be re-stated in the following equivalent form, which allows for a unified treatment of the alternation flags:

$$\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha P(\vec{x}), \text{ if } \mathcal{M}, \text{EXT}(\mathfrak{X}, \mathcal{F}) \models^\alpha P(\vec{x}),$$

where the second occurrence of the satisfaction relation \models^α refers to the hyperteam semantic relation for FO, as per Item 3 of Definition 2.1. The semantics of the *meta quantifiers* $\Sigma^{+W}x$ and $\Pi^{+W}x$ is the classic second-order one, where the functions chosen at the meta level are stored in the assignment \mathcal{F} .

The notions of satisfaction, implication, and equivalence, given at the end of Section 2.2 immediately lift to **Meta**. In addition, all relevant results proved for FO in Section 2.3 clearly lift to the **Meta** semantics of FO formulas. These results are, indeed, proved in this generalised form in 2.D. In particular, satisfaction in FO and in **Meta** coincide.

Proposition 2.2. $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$ iff $\mathcal{M}, \emptyset, \mathfrak{X} \models_{\text{Meta}}^\alpha \varphi$, for every FO formula φ and hyperteam $\mathfrak{X} \in \text{HypTeams}_{\geq}(\text{FREE}(\varphi))$.

At first glance, the semantic rule for the meta quantifiers might seem to mimic the corresponding quantifier rule of DIF (and TL), as in both cases a choice of a response function is involved. However, unlike in DIF (and TL), the application of the functions to the hyperteam is delayed until the evaluation of an atomic formula. This makes the behaviour of quantifications in the two semantics diverge significantly.

The following lemma characterises the connection between the compositional semantics of first-order quantifications $\exists x$ and $\forall x$ and the corresponding choice of a Skolem/Herbrand function.

Lemma 2.7 (Extension Interpretation). *The following four equivalences hold true, for all hypertteams $\mathfrak{X} \in \text{HypTeams}(V)$ over $V \subseteq \text{Var}$, properties $\Psi \subseteq \text{Asg}(V \cup \{x\})$ over $V \cup \{x\}$ with $x \in \text{Var} \setminus V$, flag $\alpha \in \{\exists\forall, \forall\exists\}$ and quantifier symbols $Q \in \{\exists, \forall\}$.*

1. *Statements 1a and 1b are equivalent, whenever Q is α -coherent:*
 - (a) *there exists $X' \in \text{EXT}_\alpha(\mathfrak{X}, Qx)$ such that $X' \subseteq \Psi$;*
 - (b) *there exist $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ and $X \in \mathfrak{X}$ such that $\text{EXT}(X, F, x) \subseteq \Psi$.*
2. *Statements 2a and 2b are equivalent, whenever Q is α -coherent:*
 - (a) *for all $X' \in \text{EXT}_\alpha(\mathfrak{X}, Qx)$, it holds that $X' \cap \Psi \neq \emptyset$;*
 - (b) *for all $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ and $X \in \mathfrak{X}$, it holds that $\text{EXT}(X, F, x) \cap \Psi \neq \emptyset$.*
3. *Statements 3a and 3b are equivalent, whenever Q is $\bar{\alpha}$ -coherent:*
 - (a) *there exists $X' \in \text{EXT}_\alpha(\mathfrak{X}, Qx)$ such that $X' \subseteq \Psi$;*
 - (b) *for all $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$, it holds that $\text{EXT}(X, F, x) \subseteq \Psi$, for some $X \in \mathfrak{X}$.*
4. *Statements 4a and 4b are equivalent, whenever Q is $\bar{\alpha}$ -coherent:*

- (a) for all $X' \in \text{EXT}_\alpha(\mathfrak{X}, Qx)$, it holds that $X' \cap \Psi \neq \emptyset$;
- (b) there is $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ such that $\text{EXT}(X, F, x) \cap \Psi \neq \emptyset$, for all $X \in \mathfrak{X}$.

Equivalences 1 and 4, when $Q = \exists$, implicitly state that an existential quantification can always be simulated by an existential choice of a suitable Skolem function, independently of the alternation flag α for the hyperteam. Dually, Equivalences 2 and 3, when $Q = \forall$, state that a universal quantification can be simulated by a universal choice of a suitable Herbrand function, again regardless of α . These observations can be formulated in Meta as follows.

Theorem 2.6 (Quantifier Interpretation). *The following equivalences hold true, for all FO formulas ϕ , variables $x \in \text{Var}$, acyclic function assignments $\mathcal{F} \in \text{FunAsg}$, and hyperteams $\mathfrak{X} \in \text{HypTeams}_\subseteq((\text{FREE}(\phi) \setminus \{x\}) \setminus \text{DOM}(\mathcal{F}))$ with $x \notin \text{VAR}(\mathfrak{X})$:*

1. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \exists x. \phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \Sigma^{+\text{VAR}(\mathfrak{X})} x. \phi$;
2. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \forall x. \phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \Pi^{+\text{VAR}(\mathfrak{X})} x. \phi$.

Given an FO formula $\wp\phi$ with quantifier prefix $\wp \in \mathbf{Q}$ and FO subformula ϕ , we can convert each quantification in \wp , from inside out, into the corresponding meta quantification, by suitably iterating the result reported above. The meta quantifiers in the obtained prefix are in reverse order with respect to the order of corresponding standard quantifiers in the original prefix. To formalise this idea, we introduce, given an FO formula φ the *Herbrand-Skolem prefix* function HSP_φ as follows:

1. $\text{HSP}_\varphi(\varepsilon) \stackrel{\text{def}}{=} \varepsilon$;
2. $\text{HSP}_\varphi(\wp. \exists x) \stackrel{\text{def}}{=} \Sigma^{+\text{FREE}(\varphi) \cup \text{VAR}(\wp)} x. \text{HSP}_\varphi(\wp)$;
3. $\text{HSP}_\varphi(\wp. \forall x) \stackrel{\text{def}}{=} \Pi^{+\text{FREE}(\varphi) \cup \text{VAR}(\wp)} x. \text{HSP}_\varphi(\wp)$.

We can show that $\wp\phi \equiv \text{HSP}_\varphi(\wp)\phi$, by exploiting Theorem 3.4. This conversion resembles a merging of the standard Skolem/Herbrand-isation procedures [Hei67; Bus98a] that convert a FO sentence either into an equi-satisfiable/equi-valid FO sentence without existential/universal quantifiers, or into an equivalent SO formula.

Theorem 2.7 (Herbrand-Skolem Theorem). *Let $\wp_1\wp_2\phi$ be an FO formula in pnf with quantifier prefix $\wp_1\wp_2 \in \mathbf{Q}$ and FO subformula ϕ . Then, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \wp_1\wp_2\phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \text{HSP}_\varphi(\wp_2)\wp_1\phi$, for all acyclic function assignments $\mathcal{F} \in \text{FunAsg}$ with $\text{DOM}(\mathcal{F}) \cap \text{DEP}(\wp_1\wp_2\phi) = \emptyset$ and hyperteams $\mathfrak{X} \in \text{HypTeams}_\subseteq(\text{FREE}(\wp_1\wp_2\phi) \setminus \text{DOM}(\mathcal{F}))$ with $\text{VAR}(\mathfrak{X}) \cap \text{VAR}(\wp_1\wp_2) = \emptyset$ and $\text{DOM}(\mathcal{F}) \cap \text{VAR}(\wp_1\wp_2) = \emptyset$.*

2.6 Conclusion

In this chapter we have generalized teams in such a way that the choices of both players are recorded in the semantic structure *w.r.t* which formulas are evaluated. This approach leads to the notion of *hyperteam*, defined as a set of teams, which provides a two-level structure, where each level is intuitively associated with one of the two players and encodes the uncertainty that the opponent has about the actual choices up to that stage of the play. From another perspective, the structure can be viewed as encoding all the possible plays in the underlying evaluation game, comprising the choices of one player as well as the possible responses of the opponent. With all this information at hand, then, we can easily obtain the plays of the dual game, namely the one in which the two players exchange their roles. The change of roles between the players, in turn, precisely corresponds to the game-theoretic interpretation of negation. This allows us to include negation to the logic in a very natural way and, at the same time, recover the law of the excluded middle, which is lost in IFL, by avoiding undetermined sentences, and have a fully symmetric treatment of the independence constraints on the universal and existential quantifiers.

The symmetric treatment of quantifiers enables uniformity constraints on both existential and universal quantifiers. This form of logical symmetry allows for the definition of a new logic, *Alternating Dependence/Independence Friendly Logic*, (ADIF for short) that subsume DIF and that we will investigate in the next chapter.

2.A Proofs of Section 2.1

Before each proof of a theorem, we display its dependency graph: the vertices are the results used to prove the theorem (they can be lemmas, propositions, other theorems, etc). There is an edge from Result 1 to Result 2 iff Result 1 is explicitly used in Result 2's proof.

Let $W \subseteq \text{Var}$ and $\mathfrak{X} \in \text{HypTeams}$. For a team $X \in \mathfrak{X}_{|W}$, we denote by $X \upharpoonright^W$ one (arbitrarily chosen) of the teams $Y \in \mathfrak{X}$ such that $Y_{|W} = X$.

Lemma 2.1 (Dualisation I). *For all hyperteams $\mathfrak{X} \in \text{HypTeams}$ and set of variables W , it holds that $\mathfrak{X} \equiv_W \overline{\overline{\mathfrak{X}}}$. In addition, $\mathfrak{X} \subseteq \overline{\overline{\mathfrak{X}}}$, if \mathfrak{X} is proper.*

Proof. First, observe that, by Proposition 2.1, $\overline{\overline{\mathfrak{X}}} \equiv \mathfrak{X}$ holds for every non-proper hyper-team \mathfrak{X} .

Next, we show that $\mathfrak{X} \subseteq \overline{\overline{\mathfrak{X}}}$, for a proper hyper-team \mathfrak{X} . Let $X \in \mathfrak{X}$. Observe that, since \mathfrak{X} is proper, $X' \cap X \neq \emptyset$ for all $X' \in \overline{\overline{\mathfrak{X}}}$. For every $\chi \in X$, fix a choice function $\Gamma_\chi \in \text{Chc}(\mathfrak{X})$ such that $\Gamma_\chi(X) = \chi \in X$. Now, consider $\overline{\Gamma} \in \text{Chc}(\overline{\overline{\mathfrak{X}}})$ such that $\overline{\Gamma}(\text{img}(\Gamma_\chi)) = \chi$ for all $\chi \in X$, and $\overline{\Gamma}(X') \in X \cap X'$ for all the other teams $X' \in \overline{\overline{\mathfrak{X}}} \setminus \{\text{img}(\Gamma_\chi) \mid \chi \in X\}$. Clearly, $X = \text{img}(\overline{\Gamma}) \in \overline{\overline{\mathfrak{X}}}$, hence $\mathfrak{X} \subseteq \overline{\overline{\mathfrak{X}}}$.

Since $\mathfrak{X} \subseteq \overline{\overline{\mathfrak{X}}}$ implies $\mathfrak{X} \sqsubseteq \overline{\overline{\mathfrak{X}}}$, it suffices to prove that $\overline{\overline{\mathfrak{X}}} \sqsubseteq \mathfrak{X}$ holds to obtain $\mathfrak{X} \equiv \overline{\overline{\mathfrak{X}}}$. To this end, let $\overline{X'} = \text{img}(\overline{\Gamma}) \in \overline{\overline{\mathfrak{X}}}$, for some $\overline{\Gamma} \in \text{Chc}(\overline{\overline{\mathfrak{X}}})$. We show that there is $X \in \mathfrak{X}$ such that $X \subseteq \overline{X'}$. Assume, towards a contradiction, that this is not the case, i.e., for all $X \in \mathfrak{X}$ there is $\chi_X \in X \setminus \overline{X'}$. Then, define $\Gamma \in \text{Chc}(\mathfrak{X})$ as: $\Gamma(X) = \chi_X$ for all $X \in \mathfrak{X}$. Clearly, $\overline{\Gamma}(\text{img}(\Gamma)) \notin \overline{X'}$, thus raising a contradiction. Now, the thesis follows from the observation that $\mathfrak{X} \equiv \overline{\overline{\mathfrak{X}}}$ is equivalent to $\mathfrak{X} \equiv_{\text{Var}} \overline{\overline{\mathfrak{X}}}$, which implies $\mathfrak{X} \equiv_W \overline{\overline{\mathfrak{X}}}$, due to $W \subseteq \text{Var}$. \square

Lemma 2.2 (Dualisation II). *The following equivalences hold true, for all hyperteams $\mathfrak{X} \in \text{HypTeams}$ and properties $\Psi \subseteq \text{Asg}$.*

1. *Statements 1a and 1b are equivalent:*

- (a) *there exists a team $X \in \mathfrak{X}$ (resp., $X \in \overline{\overline{\mathfrak{X}}}$) such that $X \subseteq \Psi$;*
- (b) *for all teams $X' \in \overline{\overline{\mathfrak{X}}}$ (resp., $X' \in \mathfrak{X}$), it holds that $X' \cap \Psi \neq \emptyset$.*

2. *Statements 2a and 2b are equivalent:*

- (a) *there exists a team $X \in \mathfrak{X}$ such that $X \cap \Psi \neq \emptyset$;*
- (b) *there exists a team $X' \in \overline{\overline{\mathfrak{X}}}$ such that $X' \cap \Psi \neq \emptyset$.*

3. *Statements 3a and 3b are equivalent:*

- (a) for all teams $X \in \mathfrak{X}$, it holds that $X \subseteq \Psi$;
- (b) for all teams $X' \in \overline{\mathfrak{X}}$, it holds that $X' \subseteq \Psi$.

Proof. We consider the three equivalences separately.

1. First, we show that there exists a team $X \in \mathfrak{X}$ such that $X \subseteq \Psi$ if and only if for all teams $X' \in \overline{\mathfrak{X}}$, it holds that $X' \cap \Psi \neq \emptyset$.
 - (only-if) Let X' be a generic element of $\overline{\mathfrak{X}}$. Thus, $X' = \text{img}(\Gamma)$ for some $\Gamma \in \text{Chc}(\mathfrak{X})$. Thus, $\Gamma(X) \in X \cap X'$. The thesis follows from $X \subseteq \Psi$.
 - (if) By Proposition 2.1, if $\overline{\mathfrak{X}} = \emptyset$, then $\emptyset \in \mathfrak{X}$, and the thesis immediately follows since $\emptyset \subseteq \Psi$. If, instead $\overline{\mathfrak{X}} \neq \emptyset$, then assume, towards a contradiction, that for all $X \in \mathfrak{X}$, there is $\chi_X \in X \setminus \Psi$. Define $\Gamma \in \text{Chc}(\mathfrak{X})$ as: $\Gamma(X) = \chi_X$, for all $X \in \mathfrak{X}$. Since $\text{img}(\Gamma) \in \overline{\mathfrak{X}}$ and $\text{img}(\Gamma) \cap \Psi = \emptyset$, we get a contradiction.

The rest of the claim, i.e., there exists a team $X' \in \overline{\mathfrak{X}}$ such that $X' \subseteq \Psi$ if and only if for all teams $X \in \mathfrak{X}$, it holds that $X \cap \Psi \neq \emptyset$, follows from above and the fact that $\mathfrak{X} \equiv \overline{\overline{\mathfrak{X}}}$ (Lemma 2.1).
2. (only-if) Consider $\Gamma \in \text{Chc}(\mathfrak{X})$ such that $\Gamma(X) \in X \cap \Psi$. The thesis follows from $\Gamma(X) \in \text{img}(\Gamma) \in \overline{\mathfrak{X}}$.
 - (if) Let $X' = \text{img}(\Gamma) \in \overline{\mathfrak{X}}$, for some $\Gamma \in \text{Chc}(\mathfrak{X})$, be such that $X' \cap \Psi \neq \emptyset$ and let $\bar{\chi} \in X' \cap \Psi$. Thus, there is $X \in \mathfrak{X}$ such that $\Gamma(X) = \bar{\chi} \in X$, which means that $X \cap \Psi \neq \emptyset$, hence the thesis.
3. The claim follows by instantiating Ψ with $\text{Asg} \setminus \Psi$ in the previous claim, and observing that 3a and 3b correspond to the negations of 2a and 2b, respectively. \square

2.B Proofs of Section 2.2

Lemma 2.3 (Empty & Null Hyperteams). *The following hold true for every FO formula φ and hyperteam $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi))$:*

1. (a) $\mathcal{M}, \emptyset \not\models^{\exists\forall} \varphi$;
- (b) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$, where $\emptyset \in \mathfrak{X}$;
2. (a) $\mathcal{M}, \emptyset \models^{\forall\exists} \varphi$;
- (b) $\mathcal{M}, \mathfrak{X} \not\models^{\forall\exists} \varphi$, where $\emptyset \in \mathfrak{X}$.

Proof. The claim follows from the more general Lemma 2.8, reported in Section 2.D, by instantiating \mathcal{F} with the empty function \emptyset . \square

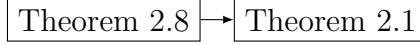


Figure 2.2 – Dependency graph of Theorem 2.1.

Theorem 2.1 (Hyperteam Refinement). *Let φ be an FO formula and $\mathfrak{X}, \mathfrak{X}' \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi))$ two hyperteams with $\mathfrak{X} \sqsubseteq_{\text{FREE}(\varphi)} \mathfrak{X}'$. Then:*

1. *if $\mathcal{M}, \mathfrak{X} \models^{\exists^{\forall}} \varphi$ then $\mathcal{M}, \mathfrak{X}' \models^{\exists^{\forall}} \varphi$;*
2. *if $\mathcal{M}, \mathfrak{X}' \models^{\forall^{\exists}} \varphi$ then $\mathcal{M}, \mathfrak{X} \models^{\forall^{\exists}} \varphi$.*

Proof. The claim follows from the more general Theorem 2.8, reported in Section 2.D, by instantiating both \mathcal{F} and ι with the empty function \emptyset . \square

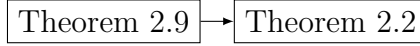


Figure 2.3 – Dependency graph of Theorem 2.2.

Theorem 2.2 (Double Dualisation). *For every FO formula φ , alternation flag α and hyperteam $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi))$, it holds that $\mathcal{M}, \mathfrak{X} \models^{\alpha} \varphi$ iff $\mathcal{M}, \overline{\mathfrak{X}} \models^{\alpha} \varphi$ iff $\mathcal{M}, \overline{\overline{\mathfrak{X}}} \models^{\overline{\alpha}} \varphi$.*

Proof. The claim follows from the more general Theorem 2.9, reported in Section 2.D, by instantiating \mathcal{F} with the empty function \emptyset . \square

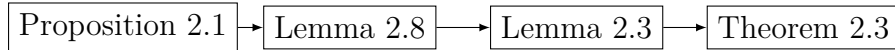


Figure 2.4 – Dependency graph of Theorem 2.3.

Theorem 2.3 (Boolean Laws). *Let φ_1, φ_2 and φ be FO formulas. Then:*

1. (a) $\neg \perp \equiv \top$;
- (b) $\neg \top \equiv \perp$;
- (c) $\varphi \equiv \neg \neg \varphi$;
2. (a) $\varphi \wedge \perp \equiv \perp \wedge \varphi \equiv \perp$;

- (b) $\varphi \wedge \top \equiv \top \wedge \varphi \equiv \varphi$;
3. (a) $\varphi \vee \top \equiv \top \vee \varphi \equiv \top$;
- (b) $\varphi \vee \perp \equiv \perp \vee \varphi \equiv \varphi$;
4. (a) $\varphi_1 \wedge \varphi_2 \equiv \varphi_2 \wedge \varphi_1$;
- (b) $\varphi_1 \vee \varphi_2 \equiv \varphi_2 \vee \varphi_1$;
5. (a) $\varphi_1 \wedge \varphi_2 \Rightarrow \varphi_1$;
- (b) $\varphi_1 \wedge (\varphi \wedge \varphi_2) \equiv (\varphi_1 \wedge \varphi) \wedge \varphi_2$;
6. (a) $\varphi_1 \Rightarrow \varphi_1 \vee \varphi_2$;
- (b) $\varphi_1 \vee (\varphi \vee \varphi_2) \equiv (\varphi_1 \vee \varphi) \vee \varphi_2$;
7. (a) $\varphi_1 \wedge \varphi_2 \equiv \neg(\neg\varphi_1 \vee \neg\varphi_2)$;
- (b) $\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2)$;
8. (a) $\exists x\varphi \equiv \neg(\forall x\neg\varphi)$;
- (b) $\forall x\varphi \equiv \neg(\exists x\neg\varphi)$.

Proof. Proving that an equivalence (resp., implication) $\varphi_1 \equiv \varphi_2$ (resp., $\varphi_1 \Rightarrow \varphi_2$) holds true amounts to showing that both $\varphi_1 \equiv^{\exists\forall} \varphi_2$ and $\varphi_1 \equiv^{\forall\exists} \varphi_2$ (resp., $\varphi_1 \Rightarrow^{\exists\forall} \varphi_2$ and $\varphi_1 \Rightarrow^{\forall\exists} \varphi_2$) hold true. However, as a consequence of Theorem 2.2, we have that $\varphi_1 \equiv^\alpha \varphi_2$ iff $\varphi_1 \equiv^{\bar{\alpha}} \varphi_2$ (resp., $\varphi_1 \Rightarrow^\alpha \varphi_2$ iff $\varphi_1 \Rightarrow^{\bar{\alpha}} \varphi_2$) for all $\alpha \in \{\exists\forall, \forall\exists\}$. Therefore, for every claim in the statement of the theorem, it is enough to focus on one of the two alternation flags $\exists\forall$ and $\forall\exists$ only. We could avoid the use of Theorem 2.2 by proving each claim for both alternation flag. However, this would not be interesting as for all claims, the arguments for both flags are the same.¹ In the following, when proving an equivalence $\varphi_1 \equiv \varphi_2$ (resp., implication $\varphi_1 \Rightarrow \varphi_2$), we assume $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi_1) \cup \text{FREE}(\varphi_2))$.

1. (a) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \neg\perp \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \not\models^{\forall\exists} \perp \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \neq \emptyset \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \models^{\exists\forall} \top$.
 - (b) $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \neg\top \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \not\models^{\exists\forall} \top \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} = \emptyset \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \models^{\forall\exists} \perp$.
 - (c) $\mathcal{M}, \mathfrak{X} \models^\alpha \neg\neg\varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \not\models^{\bar{\alpha}} \neg\varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \models^\alpha \varphi$.
2. (a) First, we prove that $\varphi \wedge \perp \equiv \perp$ holds. To this end, we show that if $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi \wedge \perp$, then $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \perp$, and vice versa. By semantics, $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi \wedge \perp$ implies that for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$, it holds that $\mathcal{M}, \mathfrak{X}_1 \models^{\exists\forall} \varphi$ or $\mathcal{M}, \mathfrak{X}_2 \models^{\exists\forall} \perp$. In particular, since $(\emptyset, \mathfrak{X}) \in \text{PART}(\mathfrak{X})$ and, by Item 1a of Lemma 2.3, $\mathcal{M}, \emptyset \not\models^{\exists\forall} \varphi$,

1. Nonetheless, this is why Theorem 2.2 does not occur in the dependency graph of Theorem 2.3.

we have that $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \perp$. Conversely, $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \perp$ means that $\emptyset \in \mathfrak{X}$. Thus, for every $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$, it holds that $\emptyset \in \mathfrak{X}_1$ or $\emptyset \in \mathfrak{X}_2$. Thanks to Item 1b of Lemma 2.3, we have $\mathcal{M}, \mathfrak{X}_1 \models^{\exists\forall} \varphi$ or $\mathcal{M}, \mathfrak{X}_2 \models^{\exists\forall} \perp$, which, by semantics of \wedge , implies $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi \wedge \perp$. To conclude, observe that $\varphi \wedge \perp \equiv \perp \wedge \varphi$ holds, due to commutativity of \wedge , formally proved below (Item 4a).

- (b) First, we prove that $\varphi \wedge \top \equiv \varphi$ holds. To this end, we show that if $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi \wedge \top$, then $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$, and vice versa. By semantics, $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi \wedge \top$ implies that for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$, it holds that $\mathcal{M}, \mathfrak{X}_1 \models^{\exists\forall} \varphi$ or $\mathcal{M}, \mathfrak{X}_2 \models^{\exists\forall} \top$. In particular, since $(\mathfrak{X}, \emptyset) \in \text{PART}(\mathfrak{X})$ and, by Item 1a of Lemma 2.3, $\mathcal{M}, \emptyset \not\models^{\exists\forall} \top$, we have that $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$. Conversely, assume $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$ and let $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$. If $\mathfrak{X}_1 = \mathfrak{X}$, then $\mathcal{M}, \mathfrak{X}_1 \models^{\exists\forall} \varphi$; if $\mathfrak{X}_1 \neq \mathfrak{X}$, then $\mathfrak{X}_2 \neq \emptyset$, and thus, by semantics of \top , it holds that $\mathcal{M}, \mathfrak{X}_2 \models^{\exists\forall} \top$. Therefore, for every $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$, it holds that $\mathcal{M}, \mathfrak{X}_1 \models^{\exists\forall} \varphi$ or $\mathcal{M}, \mathfrak{X}_2 \models^{\exists\forall} \top$, which, by semantics of \wedge , implies $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi \wedge \top$. To conclude, observe that $\varphi \wedge \top \equiv \top \wedge \varphi$ holds, due to commutativity of \wedge , formally proved below (Item 4a).
3. (a) First, we prove that $\varphi \vee \top \equiv \top$ holds. To this end, we show that if $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi \vee \top$, then $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \top$, and vice versa. By semantics, $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi \vee \top$ implies that there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that $\mathcal{M}, \mathfrak{X}_1 \models^{\forall\exists} \varphi$ and $\mathcal{M}, \mathfrak{X}_2 \models^{\forall\exists} \top$. By Item 2b of Lemma 2.3, it must be $\emptyset \notin \mathfrak{X}_i$, for each $i \in \{1, 2\}$, and thus $\emptyset \notin \mathfrak{X}$, which, by semantics of \top , implies $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \top$. Conversely, assume $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \top$. The claim follows from the fact that $(\emptyset, \mathfrak{X}) \in \text{PART}(\mathfrak{X})$ is such that $\mathcal{M}, \emptyset \models^{\forall\exists} \varphi$ (by Item 2a of Lemma 2.3) and $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \top$ (by assumption), which implies that $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi \vee \top$. To conclude, observe that $\varphi \vee \top \equiv \top \vee \varphi$ holds, due to commutativity of \vee , formally proved below (Item 4b).
- (b) First, we prove that $\varphi \vee \perp \equiv \varphi$ holds. To this end, we show that if $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi \vee \perp$, then $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi$, and vice versa. By semantics, $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi \vee \perp$ implies that there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that $\mathcal{M}, \mathfrak{X}_1 \models^{\forall\exists} \varphi$ and $\mathcal{M}, \mathfrak{X}_2 \models^{\forall\exists} \perp$. From the latter, it follows $\mathfrak{X}_2 = \emptyset$, meaning that $\mathfrak{X}_1 = \mathfrak{X}$. Therefore, we have $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi$. Conversely, assume $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi$. The claim follows from the fact that $(\mathfrak{X}, \emptyset) \in \text{PART}(\mathfrak{X})$ is such that $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi$ (by assumption) and $\mathcal{M}, \emptyset \models^{\forall\exists} \perp$ (by semantics of \perp), which implies that $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi \vee \perp$. To conclude, observe that $\varphi \vee \perp \equiv \perp \vee \varphi$ holds, due to commutativity of \vee , formally proved below (Item 4b).
4. Both Items 4a and 4b follow from the observation that $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ iff

- $(\mathfrak{X}_2, \mathfrak{X}_1) \in \text{PART}(\mathfrak{X})$.
5. (a) If $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi_1 \wedge \varphi_2$, then for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$, it holds that $\mathcal{M}, \mathfrak{X}_1 \models^{\exists\forall} \varphi_1$ or $\mathcal{M}, \mathfrak{X}_2 \models^{\exists\forall} \varphi_2$. In particular, since $(\mathfrak{X}, \emptyset) \in \text{PART}(\mathfrak{X})$, we have that $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi_1$.
- (b) The claim follows from the observation that partitioning is associative.
6. (a) Assume $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi_1$. The claim follows from the fact that $(\mathfrak{X}, \emptyset) \in \text{PART}(\mathfrak{X})$ is such that $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi_1$ and $\mathcal{M}, \emptyset \models^{\forall\exists} \varphi_2$ (by Item 2a of Lemma 2.3), which implies that $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi_1 \vee \varphi_2$.
- (b) The claim follows from the observation that partitioning is associative.
7. (a) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \neg(\neg\varphi_1 \vee \neg\varphi_2) \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \not\models^{\forall\exists} \neg\varphi_1 \vee \neg\varphi_2 \Leftrightarrow$ it does not hold that $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \neg\varphi_1 \vee \neg\varphi_2 \stackrel{\text{sem.}}{\Leftrightarrow}$ there is no $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that $\mathcal{M}, \mathfrak{X}_1 \models^{\forall\exists} \neg\varphi_1$ and $\mathcal{M}, \mathfrak{X}_2 \models^{\forall\exists} \neg\varphi_2 \Leftrightarrow$ for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ it holds that $\mathcal{M}, \mathfrak{X}_1 \not\models^{\forall\exists} \neg\varphi_1$ or $\mathcal{M}, \mathfrak{X}_2 \not\models^{\forall\exists} \neg\varphi_2 \stackrel{\text{sem.}}{\Leftrightarrow}$ for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ it holds that $\mathcal{M}, \mathfrak{X}_1 \models^{\exists\forall} \varphi_1$ or $\mathcal{M}, \mathfrak{X}_2 \models^{\exists\forall} \varphi_2 \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi_1 \wedge \varphi_2$.
- (b) $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \neg(\neg\varphi_1 \wedge \neg\varphi_2) \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \not\models^{\exists\forall} \neg\varphi_1 \wedge \neg\varphi_2 \Leftrightarrow$ it does not hold that $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \neg\varphi_1 \wedge \neg\varphi_2 \stackrel{\text{sem.}}{\Leftrightarrow}$ there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that $\mathcal{M}, \mathfrak{X}_1 \not\models^{\exists\forall} \neg\varphi_1$ and $\mathcal{M}, \mathfrak{X}_2 \not\models^{\exists\forall} \neg\varphi_2 \stackrel{\text{sem.}}{\Leftrightarrow}$ there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that $\mathcal{M}, \mathfrak{X}_1 \models^{\forall\exists} \varphi_1$ and $\mathcal{M}, \mathfrak{X}_2 \models^{\forall\exists} \varphi_2 \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi_1 \vee \varphi_2$.
8. (a) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \neg(\forall x \neg\varphi) \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \not\models^{\forall\exists} \forall x \neg\varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \text{EXT}(\mathfrak{X}, x) \not\models^{\forall\exists} \neg\varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \text{EXT}(\mathfrak{X}, x) \models^{\exists\forall} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \models^{\exists\forall} \exists x \varphi$.
- (b) $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \neg(\exists x \neg\varphi) \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \not\models^{\exists\forall} \exists x \neg\varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \text{EXT}(\mathfrak{X}, x) \not\models^{\exists\forall} \neg\varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \text{EXT}(\mathfrak{X}, x) \models^{\forall\exists} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathfrak{X} \models^{\forall\exists} \forall x \varphi$. \square

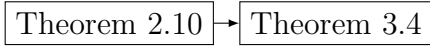


Figure 2.5 – Dependency graph of Theorem 3.4.

Theorem 3.4 (Prefix Extension). *Let $\wp\phi$ be an FO formula, where $\wp \in \mathbf{Q}$ is a quantifier prefix and ϕ is an arbitrary FO formula. Then, $\mathcal{M}, \mathfrak{X} \models^\alpha \wp\phi$ iff $\mathcal{M}, \text{EXT}_\alpha(\mathfrak{X}, \wp) \models^\alpha \phi$, for all hypertteams $\mathfrak{X} \in \text{HypTeams}_\supset(\text{FREE}(\wp\phi))$.*

Proof. The claim follows from the more general Theorem 2.10, reported in Section 2.D, by instantiating \mathcal{F} with the empty function \emptyset . \square

2.C Proofs of Section 2.4

Lemma 2.4 (FO Dualisation). *The following equivalences hold, for all FO formulas φ and hyperteams $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi))$.*

1. *Statements 1a and 1b are equivalent:*

- (a) *there exists a team $X \in \mathfrak{X}$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$, for all assignments $\chi \in X$;*
- (b) *for all teams $X \in \overline{\mathfrak{X}}$, there exists an assignment $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$.*

2. *Statements 2a and 2b are equivalent:*

- (a) *for all teams $X \in \mathfrak{X}$, there exists an assignment $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$;*
- (b) *there exists a team $X \in \overline{\mathfrak{X}}$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$, for all assignments $\chi \in X$.*

Proof. The first equivalence follows from Lemma 2.2, Item 1, by letting $\Psi = \{\chi \in \text{Asg}(V) \mid \mathcal{M}, \chi \models_{\text{FO}} \varphi \text{ and } \text{FREE}(\varphi) \subseteq V\}$. The second equivalence follows from the first one and from $\mathfrak{X} \equiv \overline{\overline{\mathfrak{X}}}$ (Lemma 2.1). \square

Lemma 2.5 (FO Quantifiers). *The following equivalences hold, for all FO formulas φ , variables $x \in \text{Var}$, and hyperteams $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi) \setminus \{x\})$.*

1. *Statements 1a and 1b are equivalent:*

- (a) *there exists a team $X \in \mathfrak{X}$ such that $\mathcal{M}, \chi \models_{\text{FO}} \exists x. \varphi$, for all $\chi \in X$;*
- (b) *there exists a team $X \in \text{EXT}(\mathfrak{X}, x)$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$, for all $\chi \in X$.*

2. *Statements 2a and 2b are equivalent:*

- (a) *for all teams $X \in \mathfrak{X}$, there exists $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \forall x. \varphi$;*
- (b) *for all teams $X \in \text{EXT}(\mathfrak{X}, x)$, there exists $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$.*

Proof. (1a \Rightarrow 1b) Let $X \in \mathfrak{X}$ be such that $\mathcal{M}, \chi \models_{\text{FO}} \exists x \varphi$ holds for every $\chi \in X$. By the standard FO semantics, for every $\chi \in X$, there is an element $v_\chi \in \mathbb{D}$ such that $\mathcal{M}, \chi[x \mapsto v_\chi] \models_{\text{FO}} \varphi$. Let $F \in \text{Rsp}$ be such that $F(\chi) = v_\chi$ for every $\chi \in X$ and let $X_F = \{\chi[x \mapsto F(\chi)] : \chi \in X\}$. Since $X_F \in \text{EXT}(\mathfrak{X}, x)$ and $\mathcal{M}, \chi \models_{\text{FO}} \varphi$ holds for every $\chi \in X_F$, the thesis holds.

(1b \Rightarrow 1a) Let $X_F = \{\chi[x \mapsto F(\chi)] : \chi \in X\} \in \text{EXT}(\mathfrak{X}, x)$, for some $X \in \mathfrak{X}$ and $F \in \text{Rsp}$, be such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$ holds for every $\chi \in X_F$. Clearly, by the standard FO semantics, this implies that $\mathcal{M}, \chi \models_{\text{FO}} \exists x \varphi$ holds for every $\chi \in X$, hence the thesis.

(2a \Leftrightarrow 2b) By statement 1 of this lemma, we have that 1a is false if and only if 1b is false ($\text{not}(1a) \Leftrightarrow \text{not}(1b)$, for short). By instantiating, in this last equivalence, φ with $\neg\varphi$, we have $1a' \Leftrightarrow 1b'$, where $1a'$ and $1b'$ are abbreviations for, respectively:

- for all teams $X \in \mathfrak{X}$, there is $\chi \in X$ such that $\mathcal{M}, \chi \not\models_{\text{FO}} \exists x \neg\varphi$;
- for all teams $X \in \text{EXT}(\mathfrak{X}, x)$, there is $\chi \in X$ such that $\mathcal{M}, \chi \not\models_{\text{FO}} \neg\varphi$.

By applying natural semantics for negation and the duality of \exists and \forall , it is straightforward to see that $1a'$ and $1b'$ correspond to 2a and 2b, respectively, hence the thesis. \square

Lemma 2.6 (FO Boolean Connectives). *The following equivalences hold, for all FO formulas φ_1 and φ_2 and hypertteams $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi_1) \cup \text{FREE}(\varphi_2))$.*

1. *Statements 1a and 1b are equivalent:*

- (a) *there exists a team $X \in \mathfrak{X}$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi_1 \wedge \varphi_2$, for all $\chi \in X$;*
- (b) *for each bipartition $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$, there exist an index $i \in \{1, 2\}$ and a team $X \in \mathfrak{X}_i$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi_i$, for all $\chi \in X$.*

2. *Statements 2a and 2b are equivalent:*

- (a) *for all teams $X \in \mathfrak{X}$, there exists $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi_1 \vee \varphi_2$;*
- (b) *there exists a bipartition $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that, for all indexes $i \in \{1, 2\}$ and teams $X \in \mathfrak{X}_i$, it holds that $\mathcal{M}, \chi \models_{\text{FO}} \varphi_i$, for some $\chi \in X$.*

Proof. (1a \Rightarrow 1b) Let $X \in \mathfrak{X}$ be such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi_1 \wedge \varphi_2$ holds for every $\chi \in X$ and consider an arbitrary pair $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$. Since $(\mathfrak{X}_1, \mathfrak{X}_2)$ is a partition of \mathfrak{X} , either $X \in \mathfrak{X}_1$ or $X \in \mathfrak{X}_2$: in the former case, let $i = 1$; in the latter, let $i = 2$. Since $X \in \mathfrak{X}_i$ and $\mathcal{M}, \chi \models_{\text{FO}} \varphi_i$ holds for every $\chi \in X$, the thesis holds.

(1b \Rightarrow 1a) Consider the hyperteam $\mathfrak{X}'_1 = \{X \in \mathfrak{X} : \forall \chi \in X . \mathcal{M}, \chi \models_{\text{FO}} \varphi_1\}$ and the pair $(\mathfrak{X}_1 \stackrel{\text{def}}{=} \mathfrak{X} \setminus \mathfrak{X}'_1, \mathfrak{X}_2 \stackrel{\text{def}}{=} \mathfrak{X}'_1) \in \text{PART}(\mathfrak{X})$. Observe that, by definition of \mathfrak{X}_1 , there is no $X \in \mathfrak{X}_1$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi_1$ holds for every $\chi \in X$. Thus, by 1b, there must exist $X \in \mathfrak{X}_2$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi_2$ holds for every $\chi \in X$. By definition of \mathfrak{X}_2 , it also holds that $\mathcal{M}, \chi \models_{\text{FO}} \varphi_1$ for every $\chi \in X$, hence the thesis.

(2a \Leftrightarrow 2a) By statement 1 of this lemma, we have that 1a is false if and only if 1b is false ($\text{not}(1a) \Leftrightarrow \text{not}(1b)$, for short). By instantiating, in this last equivalence, φ_1 with $\neg\varphi_1$ and φ_2 with $\neg\varphi_2$, we have $1a' \Leftrightarrow 1b'$, where $1a'$ and $1b'$ are abbreviations for, respectively:

- for all teams $X \in \mathfrak{X}$, there is $\chi \in X$ such that $\mathcal{M}, \chi \not\models_{\text{FO}} \neg\varphi_1 \wedge \neg\varphi_2$;

- there exists a pair of hyperteams $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that, for all indexes $i \in \{1, 2\}$ and teams $X \in \mathfrak{X}_i$, there exists an assignment $\chi \in X$ for which it holds that $\mathcal{M}, \chi \not\models_{\text{FO}} \neg\varphi_i$.

By applying natural semantics of negation and De Morgan’s laws, it is straightforward to see that $1a'$ and $1b'$ correspond to $2a$ and $2a$, respectively, hence the thesis. \square

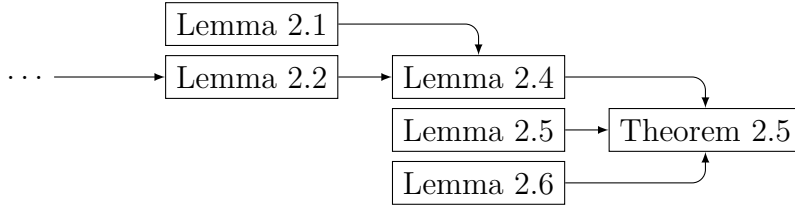


Figure 2.6 – Dependency graph of Theorem 2.5.

Theorem 2.5 (FO Adequacy). *For all FO formulas φ and hyperteams $\mathfrak{X} \in \text{HypTeams}_{\geq}(\text{FREE}(\varphi))$, it holds that:*

1. $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$ iff there exists a team $X \in \mathfrak{X}$ such that, for all assignments $\chi \in X$, it holds that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$;
2. $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi$ iff, for all teams $X \in \mathfrak{X}$, there exists an assignment $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$.

Proof. Both Items 1 and 2 are proved together, by induction on the structure of the formula.

- If φ is an atomic formula, i.e., it is \perp or \top , or it has the form $P(\vec{x})$, then the claims immediately follow from the semantics (Definition 2.1, Items 1 to 3)
- If $\varphi = \neg\phi$, then we have, by semantics, $\mathcal{M}, \mathfrak{X} \models^{\alpha} \varphi$ if and only if $\mathcal{M}, \mathfrak{X} \not\models^{\bar{\alpha}} \phi$. If $\alpha = \exists\forall$, then, by inductive hypothesis, it is not the case that for every $X \in \mathfrak{X}$ there is $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \phi$, which amounts to say that there is $X \in \mathfrak{X}$ such that for every $\chi \in X$ it holds $\mathcal{M}, \chi \not\models_{\text{FO}} \phi$, from which the thesis follows. If, instead, $\alpha = \forall\exists$, then, by inductive hypothesis, there is no $X \in \mathfrak{X}$ such that for every $\chi \in X$ it holds $\mathcal{M}, \chi \models_{\text{FO}} \phi$, which amounts to say that for every $X \in \mathfrak{X}$ there is $\chi \in X$ such that $\mathcal{M}, \chi \not\models_{\text{FO}} \phi$, from which the thesis follows.
- If $\varphi = \varphi_1 \wedge \varphi_2$ and $\alpha = \exists\forall$, then we have, by semantics, $\mathcal{M}, \mathfrak{X} \models^{\alpha} \varphi$ if and only if for every $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ it holds that $\mathcal{M}, \mathfrak{X}_1 \models^{\alpha} \varphi_1$ or $\mathcal{M}, \mathfrak{X}_2 \models^{\alpha} \varphi_2$. By

inductive hypothesis, this amounts to say that for every $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ there is $i \in \{1, 2\}$ and $X \in \mathfrak{X}_i$ such that for every $\chi \in X$ it holds $\mathcal{M}, \chi \models_{\text{FO}} \varphi_i$. The thesis follows from Lemma 2.6, Item 1.

If $\varphi = \varphi_1 \wedge \varphi_2$ and $\alpha = \forall\exists$, then we have, by semantics, $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$ if and only if $\mathcal{M}, \overline{\mathfrak{X}} \models^{\overline{\alpha}} \varphi$. By proceeding as before, i.e., by applying semantics, inductive hypothesis, and Lemma 2.6, Item 1, we have that there is $X' \in \overline{\mathfrak{X}}$ such that for every $\chi' \in X'$ it holds $\mathcal{M}, \chi' \models_{\text{FO}} \varphi$. The thesis follows from Lemma 2.4, Item 2.

- If $\varphi = \varphi_1 \vee \varphi_2$ and $\alpha = \forall\exists$, then we have, by semantics, $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$ if and only if there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that $\mathcal{M}, \mathfrak{X}_1 \models^\alpha \varphi_1$ and $\mathcal{M}, \mathfrak{X}_2 \models^\alpha \varphi_2$. By inductive hypothesis, this amounts to say that there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that for every $i \in \{1, 2\}$ and $X \in \mathfrak{X}_i$ there is $\chi \in X$ for which it holds $\mathcal{M}, \chi \models_{\text{FO}} \varphi_i$. The thesis follows from Lemma 2.6, Item 2.

If $\varphi = \varphi_1 \vee \varphi_2$ and $\alpha = \exists\forall$, then we have, by semantics, $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$ if and only if $\mathcal{M}, \overline{\mathfrak{X}} \models^{\overline{\alpha}} \varphi$. By proceeding as before, i.e., by applying semantics, inductive hypothesis, and Lemma 2.6, Item 2, we have that for every $X' \in \overline{\mathfrak{X}}$ there is $\chi' \in X'$ such that $\mathcal{M}, \chi' \models_{\text{FO}} \varphi$. The thesis follows from Lemma 2.4, Item 1.

- If $\varphi = \exists x\phi$ and $\alpha = \exists\forall$, then we have, by semantics, $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$ if and only if $\mathcal{M}, \text{EXT}(\mathfrak{X}, x) \models^\alpha \phi$. By inductive hypothesis, this amounts to say that there is $X \in \text{EXT}_{\text{FREE}(\phi) \setminus \{x\}}(\mathfrak{X}, x)$ such that for every $\chi \in X$ it holds $\mathcal{M}, \chi \models_{\text{FO}} \phi$. The thesis follows from Lemma 2.5, Item 1.

If $\varphi = \exists x\phi$ and $\alpha = \forall\exists$, then we have, by semantics, $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$ if and only if $\mathcal{M}, \overline{\mathfrak{X}} \models^{\overline{\alpha}} \varphi$. By proceeding as before, i.e., by applying semantics, inductive hypothesis, and Lemma 2.5, Item 1, we have that there is $X' \in \overline{\mathfrak{X}}$ such that for every $\chi' \in X'$ it holds $\mathcal{M}, \chi' \models_{\text{FO}} \varphi$. The thesis follows from Lemma 2.4, Item 2.

- If $\varphi = \forall x\phi$ and $\alpha = \forall\exists$, then we have, by semantics, $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$ if and only if $\mathcal{M}, \text{EXT}(\mathfrak{X}, x) \models^\alpha \phi$. By inductive hypothesis, this amounts to say that for every $X \in \text{EXT}(\mathfrak{X}, x)$ there is $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \phi$. The thesis follows from Lemma 2.5, Item 2.

If $\varphi = \forall x\phi$ and $\alpha = \exists\forall$, then we have, by semantics, $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$ if and only if $\mathcal{M}, \overline{\mathfrak{X}} \models^{\overline{\alpha}} \varphi$. By proceeding as before, i.e., by applying semantics, inductive hypothesis, and Lemma 2.5, Item 2, we have that for every $X' \in \overline{\mathfrak{X}}$ there is $\chi' \in X'$ such that $\mathcal{M}, \chi' \models_{\text{FO}} \varphi$. The thesis follows from Lemma 2.4, Item 1. \square

2.D Proofs of Section 2.5

Lemma 2.8 (Generalised Empty & Null Hyperteams). *The following hold true for every Meta formula φ , function assignment $\mathcal{F} \in \text{FunAsg}$, and hyperteam $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi) \setminus \text{DOM}(\mathcal{F}))$.*

1. (a) $\mathcal{M}, \mathcal{F}, \emptyset \not\models^{\exists\forall} \varphi$;
 (b) $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$, where $\emptyset \in \mathfrak{X}$;
2. (a) $\mathcal{M}, \mathcal{F}, \emptyset \models^{\forall\exists} \varphi$;
 (b) $\mathcal{M}, \mathcal{F}, \mathfrak{X} \not\models^{\forall\exists} \varphi$, where $\emptyset \in \mathfrak{X}$.

Proof. We proceed by structural induction on the size of φ . For the four inductive cases concerning the two binary Boolean connectives and the two standard quantifiers, it is useful to recall that, thanks to Proposition 2.1, $\bar{\emptyset} = \{\emptyset\}$ and $\bar{\mathfrak{X}} = \emptyset$ iff $\emptyset \in \mathfrak{X}$.

- [**Base case** $\varphi = \perp$] Both subitems of Item 1 directly follow from the meta-variant – $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \perp$ iff $\emptyset \in \mathfrak{X}$ – of Item 1a of Definition 2.1. Similarly, Item 2 follows from the variant – $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \perp$ iff $\mathfrak{X} = \emptyset$ – of Item 1b of the same definition.
- [**Base case** $\varphi = \top$] Both subitems of Item 2 directly follow from the meta-variant – $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \top$ iff $\emptyset \notin \mathfrak{X}$ – of Item 2a of Definition 2.1. Similarly, Item 1 follows from the variant – $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \top$ iff $\mathfrak{X} \neq \emptyset$ – of Item 2b of the same definition.
- [**Base case** $\varphi = \mathbf{P}(\vec{x})$] By observing that $\text{EXT}(\emptyset, \mathcal{F}) = \emptyset$ and $\emptyset \in \mathfrak{X}$ iff $\emptyset \in \text{EXT}(\mathfrak{X}, \mathcal{F})$, it is easy to see that Items 1 and 2 immediately follows from Items 3a and 3b of Definition 2.3, respectively.
- [**Inductive case** $\varphi = \neg\phi$] Item 1a (*resp.*, Item 1b, Item 2a, and Item 2b) follows from the meta-variant – $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\alpha} \neg\phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \not\models^{\bar{\alpha}} \phi$ – of Item 4 of Definition 2.1 and Item 2a (*resp.*, Item 2b, Item 1a, and Item 1b) of the inductive hypothesis applied to ϕ .
- [**Inductive case** $\varphi = \phi_1 \wedge \phi_2$] Items 2a and 2b directly follow from Items 1a and 1b, respectively, via the meta-variant – $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \phi_1 \wedge \phi_2$ iff $\mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\exists\forall} \phi_1 \wedge \phi_2$ – of Item 5b of Definition 2.1. We can therefore focus on the latter two.
- [**Item 1a**] By the meta-variant of Item 5a of Definition 2.1, it holds that $\mathcal{M}, \mathcal{F}, \emptyset \not\models^{\exists\forall} \varphi$ iff there exists a partitioning $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\emptyset)$ such that $\mathcal{M}, \mathcal{F}, \mathfrak{X}_1 \not\models^{\exists\forall} \phi_1$ and $\mathcal{M}, \mathcal{F}, \mathfrak{X}_2 \not\models^{\exists\forall} \phi_2$. Now, from the inductive hypothesis applied to ϕ_1 and ϕ_2 , it follows that $\mathcal{M}, \mathcal{F}, \emptyset \not\models^{\exists\forall} \phi_1$ and $\mathcal{M}, \mathcal{F}, \emptyset \not\models^{\exists\forall} \phi_2$. Moreover, $(\emptyset, \emptyset) \in \text{PART}(\emptyset)$. Thus, the thesis clearly holds.

- **[Item 1b]** By the meta-variant of Item 5a of Definition 2.1, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$ iff, for all partitioning $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X}_1 \models^{\exists\forall} \phi_1$ or $\mathcal{M}, \mathcal{F}, \mathfrak{X}_2 \models^{\exists\forall} \phi_2$, where $\emptyset \in \mathfrak{X}$. Now, from the inductive hypothesis applied to ϕ_1 and ϕ_2 , it follows that $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\exists\forall} \phi_1$ and $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\exists\forall} \phi_2$, for every hyperteam \mathfrak{X}' such that $\emptyset \in \mathfrak{X}'$. Moreover, for every partitioning $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$, one can observe that $\emptyset \in \mathfrak{X}_1$ or $\emptyset \in \mathfrak{X}_2$. Thus, the thesis clearly holds.
- **[Inductive case $\varphi = \phi_1 \vee \phi_2$]** Items 1a and 1b directly follow from Items 2a and 2b, respectively, via the meta-variant $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \phi_1 \vee \phi_2$ iff $\mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\forall\exists} \phi_1 \vee \phi_2$ – of Item 6a of Definition 2.1. We can therefore focus on the latter two.
 - **[Item 2a]** By the meta-variant of Item 6b of Definition 2.1, it holds that $\mathcal{M}, \mathcal{F}, \emptyset \models^{\forall\exists} \varphi$ iff there exists a partitioning $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\emptyset)$ such that $\mathcal{M}, \mathcal{F}, \mathfrak{X}_1 \models^{\forall\exists} \phi_1$ and $\mathcal{M}, \mathcal{F}, \mathfrak{X}_2 \models^{\forall\exists} \phi_2$. Now, by the inductive hypothesis applied to ϕ_1 and ϕ_2 , it follows that $\mathcal{M}, \mathcal{F}, \emptyset \models^{\forall\exists} \phi_1$ and $\mathcal{M}, \mathcal{F}, \emptyset \models^{\forall\exists} \phi_2$. Moreover, $(\emptyset, \emptyset) \in \text{PART}(\emptyset)$. Thus, the thesis clearly holds.
 - **[Item 2b]** By the meta-variant of Item 6b of Definition 2.1, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \not\models^{\forall\exists} \varphi$ iff, for all partitioning $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X}_1 \not\models^{\forall\exists} \phi_1$ or $\mathcal{M}, \mathcal{F}, \mathfrak{X}_2 \not\models^{\forall\exists} \phi_2$, where $\emptyset \in \mathfrak{X}$. Now, by the inductive hypothesis applied to ϕ_1 and ϕ_2 , it follows that $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \not\models^{\forall\exists} \phi_1$ and $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \not\models^{\forall\exists} \phi_2$, for every hyperteam \mathfrak{X}' such that $\emptyset \in \mathfrak{X}'$. Moreover, for every partitioning $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$, one can observe that $\emptyset \in \mathfrak{X}_1$ or $\emptyset \in \mathfrak{X}_2$. Thus, the thesis clearly holds.
- **[Inductive case $\varphi = \exists x\phi$]** Items 2a and 2b directly follow from Items 1a and 1b, respectively, via the meta-variant $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \exists x\phi$ iff $\mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\exists\forall} \exists x\phi$ – of Item 7b of Definition 2.1. We can therefore focus on the latter two.
 - **[Item 1a]** By the meta-variant of Item 7a of Definition 2.1, it holds that $\mathcal{M}, \mathcal{F}, \emptyset \not\models^{\exists\forall} \varphi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}(\emptyset, x) \not\models^{\exists\forall} \phi$. Now, by the inductive hypothesis on ϕ , it follows that $\mathcal{M}, \mathcal{F}, \emptyset \not\models^{\exists\forall} \phi$. Moreover, $\text{EXT}(\emptyset, x) = \emptyset$. Thus, the thesis clearly holds.
 - **[Item 1b]** By the meta-variant of Item 7a of Definition 2.1, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}(\mathfrak{X}, x) \models^{\exists\forall} \phi$, where $\emptyset \in \mathfrak{X}$. Now, by the inductive hypothesis on ϕ , it follows that $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\exists\forall} \phi$, for each hyperteam \mathfrak{X}' with $\emptyset \in \mathfrak{X}'$. Moreover, $\emptyset \in \text{EXT}(\mathfrak{X}, x)$. Thus, the thesis clearly holds.
- **[Inductive case $\varphi = \forall x\phi$]** Items 1a and 1b directly follow from Items 2a and 2b,

- respectively, via the meta-variant – $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \forall x\phi$ iff $\mathcal{M}, \mathcal{F}, \overline{\mathfrak{X}} \models^{\forall\exists} \forall x\phi$ – of Item 8a of Definition 2.1. We can therefore focus on the latter two.
- **[Item 2a]** By the meta-variant of Item 8b of Definition 2.1, it holds that $\mathcal{M}, \mathcal{F}, \emptyset \models^{\forall\exists} \varphi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}(\emptyset, x) \models^{\forall\exists} \phi$. Now, by the inductive hypothesis on ϕ , it follows that $\mathcal{M}, \mathcal{F}, \emptyset \models^{\forall\exists} \phi$. Moreover, $\text{EXT}(\emptyset, x) = \emptyset$. Thus, the thesis clearly holds.
 - **[Item 2b]** By the meta-variant of Item 8b of Definition 2.1, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \not\models^{\forall\exists} \varphi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}(\mathfrak{X}, x) \not\models^{\forall\exists} \phi$, where $\emptyset \in \mathfrak{X}$. Now, by the inductive hypothesis on ϕ , it follows that $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \not\models^{\forall\exists} \phi$, for each hyperteam \mathfrak{X}' with $\emptyset \in \mathfrak{X}'$. Moreover, $\emptyset \in \text{EXT}(\mathfrak{X}, x)$. Thus, the thesis clearly holds.
 - **[Inductive case $\varphi = \Sigma^{+W}x\phi$]** Since the semantics of the existential meta quantifier does not depend on the alternation flag α , we consider the two satisfaction (*resp.*, non-satisfaction) cases altogether.
 - **[Items 1a and 2b]** By Item 9 of Definition 2.3, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \not\models^{\alpha} \Sigma^{+W}x\phi$ iff, for all functions $F \in \text{Rsp}_{+W}$, it holds that $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \not\models^{\alpha} \phi$. Now, by the inductive hypothesis on ϕ , it follows that $\mathcal{M}, \mathcal{F}', \mathfrak{X} \not\models^{\alpha} \phi$, for every function assignment \mathcal{F}' , where either $\alpha = \exists\forall$ and $\mathfrak{X} = \emptyset$ or $\alpha = \forall\exists$ and $\emptyset \in \mathfrak{X}$. Thus, the thesis clearly holds.
 - **[Items 1b and 2a]** By Item 9 of Definition 2.3, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\alpha} \Sigma^{+W}x\phi$ iff there exists a function $F \in \text{Rsp}_{+W}$ such that $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^{\alpha} \phi$. Now, by the inductive hypothesis on ϕ , it follows that $\mathcal{M}, \mathcal{F}', \mathfrak{X} \models^{\alpha} \phi$, for every function assignment \mathcal{F}' , where either $\alpha = \forall\exists$ and $\mathfrak{X} = \emptyset$ or $\alpha = \exists\forall$ and $\emptyset \in \mathfrak{X}$. Thus, the thesis clearly holds.
 - **[Inductive case $\varphi = \Pi^{+W}x\phi$]** Since the semantics of the universal meta quantifier does not depend on the alternation flag α , we consider the two satisfaction (*resp.*, non-satisfaction) cases altogether.
 - **[Items 1a and 2b]** By Item 10 of Definition 2.3, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \not\models^{\alpha} \Pi^{+W}x\phi$ iff there exists a function $F \in \text{Rsp}_{+W}$ such that $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \not\models^{\alpha} \phi$. Now, by the inductive hypothesis on ϕ , it follows that $\mathcal{M}, \mathcal{F}', \mathfrak{X} \not\models^{\alpha} \phi$, for every function assignment \mathcal{F}' , where either $\alpha = \exists\forall$ and $\mathfrak{X} = \emptyset$ or $\alpha = \forall\exists$ and $\emptyset \in \mathfrak{X}$. Thus, the thesis clearly holds.
 - **[Items 1b and 2a]** By Item 10 of Definition 2.3, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\alpha} \Pi^{+W}x\phi$ iff, for all functions $F \in \text{Rsp}_{+W}$, it holds that $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^{\alpha} \phi$. Now, by the inductive hypothesis on ϕ , it follows that $\mathcal{M}, \mathcal{F}', \mathfrak{X} \models^{\alpha} \phi$, for every

function assignment \mathcal{F}' , where either $\alpha = \forall\exists$ and $\mathfrak{X} = \emptyset$ or $\alpha = \exists\forall$ and $\emptyset \in \mathfrak{X}$. Thus, the thesis clearly holds. \square

The following result states *monotonicity* of the dualization, extension, and partition operators *w.r.t* the preorder \sqsubseteq .

Lemma 2.9 (Monotonicity I). *Let $\mathfrak{X}, \mathfrak{X}' \in \text{HypTeams}$ be two hyperteams with $\mathfrak{X} \sqsubseteq \mathfrak{X}'$. Then, the following hold true:*

1. $\overline{\mathfrak{X}'} \sqsubseteq \overline{\mathfrak{X}}$;
2. (a) $\mathfrak{X} = \text{EXT}(\mathfrak{X}, x)_{\text{VAR}(\mathfrak{X})}$;
 (b) $\text{EXT}(\mathfrak{X}, x) \sqsubseteq \text{EXT}(\mathfrak{X}', x)$, with $x \in \text{Var}$;
3. for every $(\mathfrak{X}'_1, \mathfrak{X}'_2) \in \text{PART}(\mathfrak{X}')$, there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that $\mathfrak{X}_1 \sqsubseteq \mathfrak{X}'_1$ and $\mathfrak{X}_2 \sqsubseteq \mathfrak{X}'_2$.

Proof. 1) By $\mathfrak{X} \sqsubseteq \mathfrak{X}'$, there is a function $f : \mathfrak{X} \rightarrow \mathfrak{X}'$ such that $f(X) \subseteq X$ for all $X \in \mathfrak{X}$. In order to prove the claim, consider a generic team $X' \in \overline{\mathfrak{X}'}$. We have to show that there is $X \in \overline{\mathfrak{X}}$ such that $X \subseteq X'$. By the definition of $\overline{\mathfrak{X}'}$, we have that $X' = \text{img}(\Gamma')$, for some $\Gamma' \in \text{Chc}(\mathfrak{X}')$. We define $\Gamma \in \text{Chc}(\mathfrak{X})$ as: $\Gamma(X) = \Gamma'(f(X))$ for all $X \in \mathfrak{X}$. Clearly, $\text{img}(\Gamma) \subseteq (\text{img}(\Gamma')) = X'$. Since $\text{img}(\Gamma) \in \overline{\mathfrak{X}}$, the thesis holds.

2a) The claim follows from the fact that for every $F \in \text{Rsp}$, $\chi \in \text{Asg}$, and $x \in \text{Var}$, it holds that $\chi = \chi[x \mapsto F(\chi)]_{\text{VAR}(\chi)}$, which implies $X = \text{EXT}(X, F, x)_{\text{VAR}(X)}$ for every $X \in \mathfrak{X}$ and $F \in \text{Rsp}$, and the claim follows.

2b) By $\mathfrak{X} \sqsubseteq \mathfrak{X}'$, there is a function $f : \mathfrak{X} \rightarrow \mathfrak{X}'$ such that $f(X) \subseteq X$ for all $X \in \mathfrak{X}$. In order to prove the claim, take a generic team $\hat{X} \in \text{EXT}(\mathfrak{X}, x)$. Thus, $\hat{X} = \text{EXT}(X, F, x) = \{\chi[x \mapsto F(\chi)] \mid \chi \in X\}$, for some $X \in \mathfrak{X}$ and $F \in \text{Rsp}$. Let $X' = f(X) \in \mathfrak{X}'$. Clearly, $X' = f(X) \subseteq X$. Moreover, $\text{EXT}(X', F, x) \in \text{EXT}(\mathfrak{X}', x)$, since $F \in \text{Rsp}$. And finally, it is immediate that $\text{EXT}(X', F, x) \subseteq \text{EXT}(X, F, x) = \hat{X}$.

3) By $\mathfrak{X} \sqsubseteq \mathfrak{X}'$, there is a function $f : \mathfrak{X} \rightarrow \mathfrak{X}'$ such that $f(X) \subseteq X$ for all $X \in \mathfrak{X}$. Let $(\mathfrak{X}'_1, \mathfrak{X}'_2) \in \text{PART}(\mathfrak{X}')$ and define $\mathfrak{X}_i = \{X \in \mathfrak{X} \mid f(X) \in \mathfrak{X}'_i\}$ for $i \in \{1, 2\}$. It is immediate to see that $\mathfrak{X}_i \sqsubseteq \mathfrak{X}'_i$ ($i \in \{1, 2\}$). \square

Lemma 2.10 (Extension Monotonicity). *For all function assignments $\mathcal{F} \in \text{FunAsg}$, and hyperteams $\mathfrak{X}_1, \mathfrak{X}_2 \in \text{HypTeams}$, where $\mathfrak{X}_1 \sqsubseteq \mathfrak{X}_2$, it holds that $\text{EXT}(\mathfrak{X}_1, \mathcal{F}) \sqsubseteq \text{EXT}(\mathfrak{X}_2, \mathcal{F})$.*

Proof. Let $X_1 \in \text{EXT}(\mathfrak{X}_1, \mathcal{F})$. We show that there is $X_2 \in \text{EXT}(\mathfrak{X}_2, \mathcal{F})$ such that $X_2 \subseteq X_1$. By $X_1 \in \text{EXT}(\mathfrak{X}_1, \mathcal{F})$, it holds that $X_1 = \text{EXT}(X'_1, \mathcal{F})$ for some $X'_1 \in \mathfrak{X}_1$. By $\mathfrak{X}_1 \sqsubseteq \mathfrak{X}_2$, there is $X'_2 \in \mathfrak{X}_2$ such that $X'_2 \subseteq X'_1$. Thus, $\text{EXT}(X'_2, \mathcal{F}) \in \text{EXT}(\mathfrak{X}_2, \mathcal{F})$. From $X'_2 \subseteq X'_1$, it follows that $\text{EXT}(X'_2, \mathcal{F}) \subseteq \text{EXT}(X'_1, \mathcal{F}) = X_1$. Hence the thesis. \square

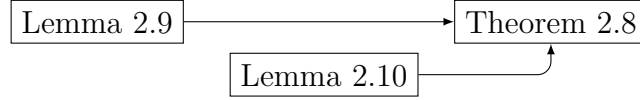


Figure 2.7 – Dependency graph of Theorem 2.8.

Theorem 2.8 (Generalised Hyperteam Refinement). *The following hold true for every Meta formula φ , function assignment $\mathcal{F} \in \text{FunAsg}$, function $\iota : \text{DOM}(\iota) \rightarrow 2^{\text{Var}}$, with $\text{DOM}(\mathcal{F}) \subseteq \text{DOM}(\iota)$, and hyperteam $\mathfrak{X}, \mathfrak{X}' \in \text{HypTeams}_{\sqsupseteq}(\text{FREE}(\varphi) \setminus \text{DOM}(\mathcal{F}))$, with $\mathcal{F}(x) \in \text{Rsp}_{+\iota(x)}$, for all $x \in \text{DOM}(\mathcal{F})$, and $\mathfrak{X} \sqsubseteq \mathfrak{X}'$:*

1. *if $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$ then $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\exists\forall} \varphi$;*
2. *if $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\forall\exists} \varphi$ then $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \varphi$.*

Proof. Due to $\mathfrak{X} \sqsubseteq \mathfrak{X}'$, there is a function $f : \mathfrak{X} \rightarrow \mathfrak{X}'$, such that $f(X) \subseteq X$ for every $X \in \mathfrak{X}$. The claim is proved by induction on the structure of the formula and the alternation flag α .

- If $\varphi = \perp$, then $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$ implies $\emptyset \in \mathfrak{X}$. By $\mathfrak{X} \sqsubseteq \mathfrak{X}'$, we have $\emptyset \in \mathfrak{X}'$. Thus, $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\exists\forall} \varphi$.
On the other hand, we also have that $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\forall\exists} \varphi$ implies $\mathfrak{X}' = \emptyset$. By $\mathfrak{X} \sqsubseteq \mathfrak{X}'$, we have $\mathfrak{X} = \emptyset$. Thus, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \varphi$.
- If $\varphi = \top$, then $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$ implies $\mathfrak{X} \neq \emptyset$. By $\mathfrak{X} \sqsubseteq \mathfrak{X}'$, we have $\mathfrak{X}' \neq \emptyset$. Thus, $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\exists\forall} \varphi$.
On the other hand, we also have that $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\forall\exists} \varphi$ implies $\emptyset \notin \mathfrak{X}'$. By $\mathfrak{X} \sqsubseteq \mathfrak{X}'$, we have $\emptyset \notin \mathfrak{X}$. Thus, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \varphi$.
- If $\varphi = \text{P}(\vec{x})$, then $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$ implies the existence of a team $X \in \text{EXT}(\mathfrak{X}, \mathcal{F})$ such that, for all assignments $\chi \in X$, it holds that $\chi(\vec{x}) \in \text{P}^{\mathcal{M}}$. By $\mathfrak{X} \sqsubseteq \mathfrak{X}'$ and Lemma 2.10, for all $x \in \text{DOM}(\mathcal{F})$, we have that $\text{EXT}(\mathfrak{X}, \mathcal{F}) \sqsubseteq \text{EXT}(\mathfrak{X}', \mathcal{F})$, and thus there is a team $X' \in \text{EXT}(\mathfrak{X}', \mathcal{F})$ such that $X' \subseteq X$, which implies $X'_{\upharpoonright \vec{x}} \subseteq X_{\upharpoonright \vec{x}}$, since $\vec{x} \subseteq \text{FREE}(\varphi)$. The thesis follows from the fact that $\chi(\vec{x}) \in \text{P}^{\mathcal{M}}$ if and only if $\chi_{\upharpoonright \vec{x}}(\vec{x}) \in \text{P}^{\mathcal{M}}$ holds, for every $\chi \in \text{Asg}$.

- On the other hand, we also have that $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\forall\exists} \varphi$ implies that for all teams $X' \in \text{EXT}(\mathfrak{X}', \mathcal{F})$, there exists an assignment $\chi' \in X'$ such that $\chi'(\vec{x}) \in \mathbb{P}^{\mathcal{M}}$. By $\mathfrak{X} \sqsubseteq \mathfrak{X}'$ and Lemma 2.10, we have that $\text{EXT}(\mathfrak{X}, \mathcal{F}) \sqsubseteq \text{EXT}(\mathfrak{X}', \mathcal{F})$, and thus for every team $X \in \text{EXT}(\mathfrak{X}, \mathcal{F})$ there is a team $X' \in \text{EXT}(\mathfrak{X}', \mathcal{F})$ such that $X \subseteq X'|_W$. The thesis follows from the same argument used above.
- If $\varphi = \neg\phi$, then $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$ implies $\mathcal{M}, \mathcal{F}, \mathfrak{X} \not\models^{\forall\exists} \phi$. By inductive hypothesis, this implies $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \not\models^{\forall\exists} \phi$, which amounts to $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\exists\forall} \varphi$.
On the other hand, we also have that $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\forall\exists} \varphi$ implies $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \not\models^{\exists\forall} \phi$. By inductive hypothesis, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \not\models^{\exists\forall} \phi$, which amounts to $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \varphi$.
 - Let $\varphi = \phi_1 \wedge \phi_2$. We assume $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$ and we show that $\mathcal{M}, \mathcal{F}, \mathfrak{X}'_1 \models^{\exists\forall} \phi_1$ or $\mathcal{M}, \mathcal{F}, \mathfrak{X}'_2 \models^{\exists\forall} \phi_2$ holds for all $(\mathfrak{X}'_1, \mathfrak{X}'_2) \in \text{PART}(\mathfrak{X}')$. To this end, let $(\mathfrak{X}'_1, \mathfrak{X}'_2) \in \text{PART}(\mathfrak{X}')$. By Lemma 2.9, item 3, there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that $\mathfrak{X}_1 \sqsubseteq \mathfrak{X}'_1$ and $\mathfrak{X}_2 \sqsubseteq \mathfrak{X}'_2$, and, by the semantics of \wedge , we have that $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ implies that $\mathcal{M}, \mathcal{F}, \mathfrak{X}_1 \models^{\exists\forall} \phi_1$ or $\mathcal{M}, \mathcal{F}, \mathfrak{X}_2 \models^{\exists\forall} \phi_2$. Finally, by inductive hypothesis it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X}'_1 \models^{\exists\forall} \phi_1$ or $\mathcal{M}, \mathcal{F}, \mathfrak{X}'_2 \models^{\exists\forall} \phi_2$.
On the other hand, we also have that $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\forall\exists} \varphi$ if and only if $\mathcal{M}, \mathcal{F}, \overline{\mathfrak{X}'} \models^{\exists\forall} \varphi$. By inductive hypothesis and Lemma 2.9, Item 1, this implies $\mathcal{M}, \mathcal{F}, \overline{\mathfrak{X}} \models^{\exists\forall} \varphi$, which amounts to $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \varphi$.
 - Let $\varphi = \phi_1 \vee \phi_2$. In this case, we first prove the second item of the claim. We assume $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\forall\exists} \varphi$ and we show that there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that $\mathcal{M}, \mathcal{F}, \mathfrak{X}_1 \models^{\forall\exists} \phi_1$ and $\mathcal{M}, \mathcal{F}, \mathfrak{X}_2 \models^{\forall\exists} \phi_2$. By the semantics of \vee , we have that there is $(\mathfrak{X}'_1, \mathfrak{X}'_2) \in \text{PART}(\mathfrak{X}')$ such that $\mathcal{M}, \mathcal{F}, \mathfrak{X}'_1 \models^{\forall\exists} \phi_1$ and $\mathcal{M}, \mathcal{F}, \mathfrak{X}'_2 \models^{\forall\exists} \phi_2$. By Lemma 2.9, item 3, there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that $\mathfrak{X}_1 \sqsubseteq \mathfrak{X}'_1$ and $\mathfrak{X}_2 \sqsubseteq \mathfrak{X}'_2$. Finally, by inductive hypothesis, $\mathcal{M}, \mathcal{F}, \mathfrak{X}_1 \models^{\forall\exists} \phi_1$ and $\mathcal{M}, \mathcal{F}, \mathfrak{X}_2 \models^{\forall\exists} \phi_2$.
On the other hand, we also have that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$ if and only if $\mathcal{M}, \mathcal{F}, \overline{\mathfrak{X}} \models^{\forall\exists} \varphi$. By inductive hypothesis and Lemma 2.9, Item 1, this implies $\mathcal{M}, \mathcal{F}, \overline{\mathfrak{X}'} \models^{\forall\exists} \varphi$, which amounts to $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\exists\forall} \varphi$.
 - If $\varphi = \exists x\phi$, then $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$ implies $\mathcal{M}, \mathcal{F}, \text{EXT}(\mathfrak{X}, x) \models^{\exists\forall} \phi$. By Lemma 2.9, item 2b, we have $\text{EXT}(\mathfrak{X}, x) \sqsubseteq \text{EXT}(\mathfrak{X}', x)$. From the inductive hypothesis, it follows $\mathcal{M}, \mathcal{F}, \text{EXT}(\mathfrak{X}', x) \models^{\exists\forall} \phi$, which amounts to $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\exists\forall} \varphi$.
On the other hand, we also have, by semantics, $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\forall\exists} \varphi$ if and only if $\mathcal{M}, \mathcal{F}, \overline{\mathfrak{X}'} \models^{\exists\forall} \varphi$. By inductive hypothesis and Lemma 2.9, Item 1, this implies $\mathcal{M}, \mathcal{F}, \overline{\mathfrak{X}} \models^{\exists\forall} \varphi$, which amounts to $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \varphi$.
 - If $\varphi = \forall x\phi$, then $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\forall\exists} \varphi$ implies $\mathcal{M}, \mathcal{F}, \text{EXT}(\mathfrak{X}', x) \models^{\forall\exists} \phi$.

- By Lemma 2.9, item 2b, we have $\text{EXT}(\mathfrak{X}, x) \sqsubseteq \text{EXT}(\mathfrak{X}', x)$. From the inductive hypothesis, it follows $\mathcal{M}, \mathcal{F}, \text{EXT}(\mathfrak{X}, x) \models^{\forall\exists} \phi$, which amounts to $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \phi$. On the other hand, we also have, by semantics, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \phi$ if and only if $\mathcal{M}, \mathcal{F}, \overline{\mathfrak{X}} \models^{\forall\exists} \phi$. By inductive hypothesis and Lemma 2.9, Item 1, this implies $\mathcal{M}, \mathcal{F}, \overline{\mathfrak{X}'} \models^{\forall\exists} \phi$, which amounts to $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\exists\forall} \phi$.
- If $\phi = \Sigma^{+W}x\phi$, then $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \phi$ implies $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^{\exists\forall} \phi$, for some function $F \in \text{Rsp}_{+W}$. By inductive hypothesis, we have $\mathcal{M}, \mathcal{F}[x \mapsto F], \overline{\mathfrak{X}'} \models^{\forall\exists} \phi$, from which $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\exists\forall} \phi$ follows.
- On the other hand, we also have that $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\forall\exists} \phi$ implies $\mathcal{M}, \mathcal{F}[x \mapsto F], \overline{\mathfrak{X}'} \models^{\forall\exists} \phi$, for some function $F \in \text{Rsp}_{+W}$. By inductive hypothesis, we have $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^{\forall\exists} \phi$, from which $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \phi$ follows.
- Finally, let $\phi = \Pi^{+W}x\phi$. Then, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \phi$ implies $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^{\exists\forall} \phi$, for all functions $F \in \text{Rsp}_{+W}$. By inductive hypothesis, we have that $\mathcal{M}, \mathcal{F}[x \mapsto F], \overline{\mathfrak{X}'} \models^{\forall\exists} \phi$ holds for all functions $F \in \text{Rsp}_{+W}$, which amounts to $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\exists\forall} \phi$. On the other hand, we also have that $\mathcal{M}, \mathcal{F}, \mathfrak{X}' \models^{\forall\exists} \phi$ which implies $\mathcal{M}, \mathcal{F}[x \mapsto F], \overline{\mathfrak{X}'} \models^{\forall\exists} \phi$, for all functions $F \in \text{Rsp}_{+W}$. By inductive hypothesis, we have that $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^{\forall\exists} \phi$ holds for all functions $F \in \text{Rsp}_{+W}$, which amounts to $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \phi$. \square

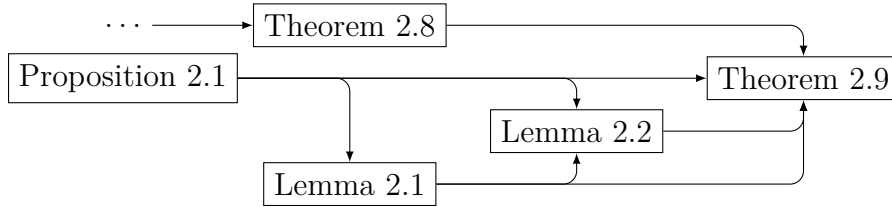


Figure 2.8 – Dependency graph of Theorem 2.9.

Theorem 2.9 (Generalized Double Dualisation). *For every FO formula ϕ , function assignment $\mathcal{F} \in \text{FunAsg}$, and hypertteam $\mathfrak{X} \in \text{HypTeams}_{\geq}(\text{FREE}(\phi) \setminus \text{DOM}(\mathcal{F}))$, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\alpha} \phi$ iff $\mathcal{M}, \mathcal{F}, \overline{\mathfrak{X}} \models^{\alpha} \phi$. Moreover, if \mathcal{F} is acyclic, then it also holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\alpha} \phi$ iff $\mathcal{M}, \mathcal{F}, \overline{\mathfrak{X}} \models^{\bar{\alpha}} \phi$.*

Proof. The fact that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\alpha} \phi$ iff $\mathcal{M}, \mathcal{F}, \overline{\mathfrak{X}} \models^{\alpha} \phi$ immediately follows from $\mathfrak{X} \equiv \overline{\overline{\mathfrak{X}}}$, for every function $\iota \in \text{Var} \rightarrow 2^{\text{Var}}$ (Lemma 2.1), and Theorem 2.8.

We turn now to proving that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \varphi$ iff $\mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\bar{\alpha}} \varphi$. As a preliminary result, notice that if \mathcal{F} is acyclic, then for every $X \subseteq \text{Asg}(U)$, for some $U \subseteq \text{Var}$, there is a bijection τ between X and $\text{EXT}(X, \mathcal{F})$, with $\tau(\chi)|_U = \chi$. Consequently, it holds that $\text{EXT}(\bar{\mathfrak{X}}, \mathcal{F}) = \overline{\text{EXT}(\mathfrak{X}, \mathcal{F})}$. The proof is done by case analysis of the syntax of the formula.

- If $\varphi = \perp$, then we have:
 - $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \emptyset \in \mathfrak{X} \stackrel{\text{Prop. 2.1}}{\Leftrightarrow} \bar{\mathfrak{X}} = \emptyset \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\forall\exists} \varphi$, and
 - $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathfrak{X} = \emptyset \stackrel{\text{Prop. 2.1}}{\Leftrightarrow} \mathfrak{X} \equiv \emptyset \stackrel{\text{Lemma 2.1}}{\Leftrightarrow} \bar{\bar{\mathfrak{X}}} \equiv \emptyset \stackrel{\text{Prop. 2.1}}{\Leftrightarrow} \bar{\bar{\mathfrak{X}}} = \emptyset \stackrel{\text{Prop. 2.1}}{\Leftrightarrow} \emptyset \in \bar{\bar{\mathfrak{X}}} \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\bar{\mathfrak{X}}} \models^{\exists\forall} \varphi$.
- If $\varphi = \top$, then we have:
 - $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathfrak{X} \neq \emptyset \stackrel{\text{Prop. 2.1}}{\Leftrightarrow} \mathfrak{X} \neq \emptyset \stackrel{\text{Lemma 2.1}}{\Leftrightarrow} \bar{\bar{\mathfrak{X}}} \neq \emptyset \stackrel{\text{Prop. 2.1}}{\Leftrightarrow} \bar{\bar{\mathfrak{X}}} \neq \emptyset \stackrel{\text{Prop. 2.1}}{\Leftrightarrow} \emptyset \notin \bar{\bar{\mathfrak{X}}} \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\bar{\mathfrak{X}}} \models^{\forall\exists} \varphi$, and
 - $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \emptyset \notin \mathfrak{X} \stackrel{\text{Prop. 2.1}}{\Leftrightarrow} \bar{\mathfrak{X}} \neq \emptyset \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\exists\forall} \varphi$.
- If $\varphi = P(\vec{x})$, then the claim follows from the semantics, Lemma 2.2, Item 1, and the fact that $\text{EXT}(\bar{\mathfrak{X}}, \mathcal{F}) = \overline{\text{EXT}(\mathfrak{X}, \mathcal{F})}$.
- If $\varphi = \neg\psi$, then we have: $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \mathfrak{X} \not\models^{\bar{\alpha}} \psi \stackrel{\text{ind.hp.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^\alpha \psi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\bar{\alpha}} \varphi$.
- If $\varphi = \varphi_1 \wedge \varphi_2$, then we have:
 - $\mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\forall\exists} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\exists\forall} \varphi \stackrel{\text{Thm. 2.9 (part 1)}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$, and
 - $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\exists\forall} \varphi$.
- If $\varphi = \varphi_1 \vee \varphi_2$, then we have:
 - $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\forall\exists} \varphi$, and
 - $\mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\exists\forall} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\forall\exists} \varphi \stackrel{\text{Thm. 2.9 (part 1)}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \varphi$.
- If $\varphi = \exists x\phi$, then we have:
 - $\mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\forall\exists} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\exists\forall} \varphi \stackrel{\text{Thm. 2.9 (part 1)}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi$, and
 - $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\exists\forall} \varphi$.
- If $\varphi = \forall x\phi$, then we have:
 - $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\forall\exists} \varphi$;
 - $\mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\exists\forall} \varphi \stackrel{\text{sem.}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\forall\exists} \varphi \stackrel{\text{Thm. 2.9 (part 1)}}{\Leftrightarrow} \mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \varphi$. □

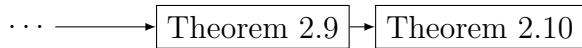


Figure 2.9 – Dependency graph of Theorem 2.10.

Theorem 2.10 (Generalized Prefix Extension). *Let $\wp\phi$ be an FO formula, where $\wp \in \mathbf{Q}$ is a quantifier prefix and ϕ is an arbitrary FO formula. Then, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \wp\phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\mathfrak{X}, \wp) \models^\alpha \phi$, for all acyclic function assignments $\mathcal{F} \in \text{FunAsg}$ and hyperteams $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\wp\phi) \setminus \text{DOM}(\mathcal{F}))$.*

Proof. We proceed by induction on the structure of the quantification prefix $\wp \in \mathbf{Q}$.

- [**Base case** $\wp = \varepsilon$] Since $\text{EXT}_\alpha(\mathfrak{X}, \varepsilon) = \mathfrak{X}$, there is really nothing to prove as the statement is trivially true.
- [**Inductive case** $\wp = Qx\wp'$] We proceed by a case analysis on the coherence of the quantifier Q with the alternation flag α .
 - [**Q is α -coherent**] By the meta-variants of Items 7a and 8b of Definition 2.1, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \wp\phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}(\mathfrak{X}, x) \models^\alpha \wp'\phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\mathfrak{X}, Qx) \models^\alpha \wp'\phi$. Now, by the inductive hypothesis, it follows that $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\mathfrak{X}, Qx) \models^\alpha \wp'\phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, Qx), \wp') \models^\alpha \phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\mathfrak{X}, \wp) \models^\alpha \phi$, which concludes the proof of this case.
 - [**Q is $\bar{\alpha}$ -coherent**] By the meta-variants of Items 7b and 8a of Definition 2.1, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \wp\phi$ iff $\mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\bar{\alpha}} \wp\phi$. Now, by the meta-variants of Items 7a and 8b of the same definition, $\mathcal{M}, \mathcal{F}, \bar{\mathfrak{X}} \models^{\bar{\alpha}} \wp\phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}(\bar{\mathfrak{X}}, x) \models^{\bar{\alpha}} \wp'\phi$. Thanks to Theorem 2.9, $\mathcal{M}, \mathcal{F}, \text{EXT}(\bar{\mathfrak{X}}, x) \models^{\bar{\alpha}} \wp'\phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}(\bar{\mathfrak{X}}, x) \models^\alpha \wp'\phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\bar{\mathfrak{X}}, Qx) \models^\alpha \wp'\phi$. Summing up, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \wp\phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\mathfrak{X}, Qx) \models^\alpha \wp'\phi$. At this point, by the inductive hypothesis, it follows that $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\mathfrak{X}, Qx) \models^\alpha \wp'\phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, Qx), \wp') \models^\alpha \phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\mathfrak{X}, \wp) \models^\alpha \phi$, which concludes the proof of this case as well. \square

Lemma 2.7 (Extension Interpretation). *The following four equivalences hold true, for all hyperteams $\mathfrak{X} \in \text{HypTeams}(V)$ over $V \subseteq \text{Var}$, properties $\Psi \subseteq \text{Asg}(V \cup \{x\})$ over $V \cup \{x\}$ with $x \in \text{Var} \setminus V$, flag $\alpha \in \{\exists\forall, \forall\exists\}$ and quantifier symbols $Q \in \{\exists, \forall\}$.*

1. *Statements 1a and 1b are equivalent, whenever Q is α -coherent:*

- (a) *there exists $X' \in \text{EXT}_\alpha(\mathfrak{X}, Qx)$ such that $X' \subseteq \Psi$;*
- (b) *there exist $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ and $X \in \mathfrak{X}$ such that $\text{EXT}(X, F, x) \subseteq \Psi$.*

2. *Statements 2a and 2b are equivalent, whenever Q is α -coherent:*

- (a) *for all $X' \in \text{EXT}_\alpha(\mathfrak{X}, Qx)$, it holds that $X' \cap \Psi \neq \emptyset$;*
- (b) *for all $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ and $X \in \mathfrak{X}$, it holds that $\text{EXT}(X, F, x) \cap \Psi \neq \emptyset$.*

3. Statements 3a and 3b are equivalent, whenever Q is $\bar{\alpha}$ -coherent:

(a) there exists $X' \in \text{EXT}_\alpha(\mathfrak{X}, Qx)$ such that $X' \subseteq \Psi$;

(b) for all $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$, it holds that $\text{EXT}(X, F, x) \subseteq \Psi$, for some $X \in \mathfrak{X}$.

4. Statements 4a and 4b are equivalent, whenever Q is $\bar{\alpha}$ -coherent:

(a) for all $X' \in \text{EXT}_\alpha(\mathfrak{X}, Qx)$, it holds that $X' \cap \Psi \neq \emptyset$;

(b) there is $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ such that $\text{EXT}(X, F, x) \cap \Psi \neq \emptyset$, for all $X \in \mathfrak{X}$.

Proof. We first prove Items 1 and 2 altogether, where Q is α -coherent, and then we proceed with the remaining ones separately. In particular, for these last two, we make use, given an arbitrary function $F \in \text{Rsp}_{+W}$, of the auxiliary notation $\text{PRJ}(\Psi, F, x) \stackrel{\text{def}}{=} \{\chi \in \text{Asg}(V) \mid \chi[x \mapsto F(\chi)] \in \Psi\}$ satisfying the following two properties, for every team $X \in \text{Teams}(V)$:

1. $\text{EXT}(X, F, x) \subseteq \Psi$ iff $X \subseteq \text{PRJ}(\Psi, F, x)$;
2. $\text{EXT}(X, F, x) \cap \Psi \neq \emptyset$ iff $X \cap \text{PRJ}(\Psi, F, x) \neq \emptyset$.

— **[Items 1 and 2]** By definition of the extension function, when Q is α -coherent, we have that

$$\text{EXT}_\alpha(\mathfrak{X}, Qx) = \text{EXT}(\mathfrak{X}, x) = \{\text{EXT}(X, F, x) \mid X \in \mathfrak{X}, F \in \text{Rsp}\}.$$

Thus, for every possible team $X' \in \text{Teams}(V \cup \{x\})$, it holds that $X' \in \text{EXT}_\alpha(\mathfrak{X}, Qx)$ iff there exists a function $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ and a team $X \in \mathfrak{X}$ such that $X' = \text{EXT}(X, F, x)$. Hence, both equivalences immediately follows.

— **[Item 3]** Since Q is $\bar{\alpha}$ -coherent, $\text{EXT}_\alpha(\mathfrak{X}, Qx) = \overline{\text{EXT}(\bar{\mathfrak{X}}, x)}$, and thus Condition 3a holds iff there is a team $X' \in \overline{\text{EXT}(\bar{\mathfrak{X}}, x)}$ such that $X' \subseteq \Psi$. By Item 1 of Lemma 2.2, this holds iff for all teams $X' \in \text{EXT}(\bar{\mathfrak{X}}, x) = \text{EXT}_\alpha(\bar{\mathfrak{X}}, Qx)$, it holds that $X' \cap \Psi \neq \emptyset$. Thanks to Item 2, the latter is true iff for all functions $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ and teams $X \in \bar{\mathfrak{X}}$, it holds that $\text{EXT}(X, F, x) \cap \Psi \neq \emptyset$, and thus $X \cap \text{PRJ}(\Psi, F, x) \neq \emptyset$. At this point, again by Item 1 of Lemma 2.2, for all $X \in \bar{\mathfrak{X}}$, it holds that $X \cap \text{PRJ}(\Psi, F, x) \neq \emptyset$ iff there exists a team $X \in \mathfrak{X}$ such that $X \subseteq \text{PRJ}(\Psi, F, x)$, and thus $\text{EXT}(X, F, x) \subseteq \Psi$. Therefore, the following equivalence concludes the proof: for all functions $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ and teams $X \in \bar{\mathfrak{X}}$, it holds that $X \cap \text{PRJ}(\Psi, F, x) \neq \emptyset$ iff for all functions $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ there exists a team $X \in \mathfrak{X}$ such that $\text{EXT}(X, F, x) \subseteq \Psi$, which coincides with Condition 3b.

- **[Item 4]** Since Q is $\bar{\alpha}$ -coherent, $\text{EXT}_\alpha(\mathfrak{X}, Qx) = \overline{\text{EXT}(\bar{\mathfrak{X}}, x)}$, and thus Condition 4a holds iff for all teams $X' \in \overline{\text{EXT}(\bar{\mathfrak{X}}, x)}$, it holds that $X' \cap \Psi \neq \emptyset$. By Item 1 of Lemma 2.2, this holds iff there exists a team $X \in \text{EXT}(\bar{\mathfrak{X}}, x) = \text{EXT}_\alpha(\bar{\mathfrak{X}}, \bar{Q}x)$ such that $X \subseteq \Psi$. Thanks to Item 1, the latter is true iff there exist a function $F \in \text{Rsp}_{+\text{VAR}(\bar{\mathfrak{X}})}$ and a team $X \in \bar{\mathfrak{X}}$ such that $\text{EXT}(X, F, x) \subseteq \Psi$, and thus $X \subseteq \text{PRJ}(\Psi, F, x)$. At this point, again by Item 1 of Lemma 2.2, there exists a team $X \in \bar{\mathfrak{X}}$ such that $X \subseteq \text{PRJ}(\Psi, F, x)$ iff for all teams $X' \in \bar{\mathfrak{X}}$, it holds that $X' \cap \text{PRJ}(\Psi, F, x) \neq \emptyset$, and thus $\text{EXT}(X', F, x) \cap \Psi \neq \emptyset$. Therefore, the following equivalence concludes the proof: there exist a function $F \in \text{Rsp}_{+\text{VAR}(\bar{\mathfrak{X}})}$ and a team $X \in \bar{\mathfrak{X}}$ such that $X \subseteq \text{PRJ}(\Psi, F, x)$ iff there exists a function $F \in \text{Rsp}_{+\text{VAR}(\bar{\mathfrak{X}})}$ such that for all teams $X' \in \bar{\mathfrak{X}}$, it holds that $\text{EXT}(X', F, x) \cap \Psi \neq \emptyset$, which coincides with Condition 4b. \square

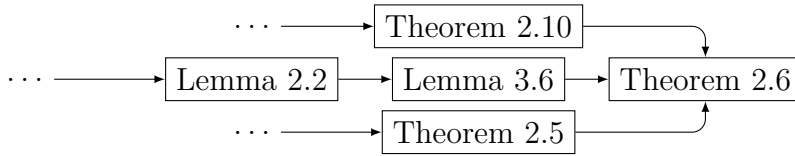


Figure 2.10 – Dependency graph of Theorem 2.6.

Theorem 2.6 (Quantifier Interpretation). *The following equivalences hold true, for all FO formulas ϕ , variables $x \in \text{Var}$, acyclic function assignments $\mathcal{F} \in \text{FunAsg}$, and hyperteams $\mathfrak{X} \in \text{HypTeams}_\subseteq((\text{FREE}(\phi) \setminus \{x\}) \setminus \text{DOM}(\mathcal{F}))$ with $x \notin \text{VAR}(\mathfrak{X})$:*

1. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\alpha} \exists x. \phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\alpha} \Sigma^{+\text{VAR}(\mathfrak{X})} x. \phi$;
2. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\alpha} \forall x. \phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\alpha} \Pi^{+\text{VAR}(\mathfrak{X})} x. \phi$.

Proof. First, observe that, by a generalisation of Theorem 2.5 to **Meta**, the following two equivalences hold true, where we define $\llbracket \phi \rrbracket \stackrel{\text{def}}{=} \{\chi \in \text{Asg}(V) \mid \text{FREE}(\phi) \subseteq \text{Vand } \mathcal{M}, \chi \models_{\text{FO}} \phi\}$ for every FO formula ϕ and acyclic function assignments $\mathcal{F} \in \text{FunAsg}$:

1. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \phi$ iff $X \subseteq \llbracket \phi \rrbracket$, for some team $X \in \text{EXT}(\mathfrak{X}, \mathcal{F})$;
2. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \phi$ iff $X \cap \llbracket \phi \rrbracket \neq \emptyset$, for all teams $X \in \text{EXT}(\mathfrak{X}, \mathcal{F})$.

which are equivalent to the following, respectively:

- $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \phi$ iff $\text{EXT}(X, \mathcal{F}) \subseteq \llbracket \phi \rrbracket$, for some team $X \in \mathfrak{X}$;
- $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \phi$ iff $\text{EXT}(X, \mathcal{F}) \cap \llbracket \phi \rrbracket \neq \emptyset$, for all teams $X \in \mathfrak{X}$.

For technical convenience, given $U \subseteq \text{Var}$ and $\Psi \subseteq \text{Asg}(U \cup \text{DOM}(\mathcal{F}))$, let us introduce the notation $\text{PRJ}(\Psi, U, \mathcal{F}) \stackrel{\text{def}}{=} \{\chi \in \Psi \mid \forall x \in \text{DOM}(\mathcal{F}) \setminus U, \chi(x) = \mathcal{F}(x)(\chi)\}_{\upharpoonright U}$. Thanks to the assumption of \mathcal{F} being acyclic, the following two properties hold, for every team $X \in \text{Teams}(U)$:

1. $\text{EXT}(X, \mathcal{F}) \subseteq \Psi$ iff $X \subseteq \text{PRJ}(\Psi, U, \mathcal{F})$;
2. $\text{EXT}(X, \mathcal{F}) \cap \Psi \neq \emptyset$ iff $X \cap \text{PRJ}(\Psi, U, \mathcal{F}) \neq \emptyset$.

In the light of this notation, we can rewrite the last two equivalences above as follows:

- $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \phi$ iff $X \subseteq \text{PRJ}(\llbracket \phi \rrbracket, \text{VAR}(X), \mathcal{F})$, for some team $X \in \mathfrak{X}$;
- $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \phi$ iff $X \cap \text{PRJ}(\llbracket \phi \rrbracket, \text{VAR}(X), \mathcal{F}) \neq \emptyset$, for all teams $X \in \mathfrak{X}$.

By applying to a formula $Qx\phi$, where $Q \in \{\exists, \forall\}$, a combination of Theorem 2.10 and what we have just derived, we obtain the two equivalences below:

1. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} Qx\phi$ iff there exists a team $X \in \text{EXT}_{\exists\forall}(\mathfrak{X}, Qx)$ such that $X \subseteq \text{PRJ}(\llbracket \phi \rrbracket, \text{VAR}(X), \mathcal{F})$;
2. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} Qx\phi$ iff, for all teams $X \in \text{EXT}_{\forall\exists}(\mathfrak{X}, Qx)$, it holds that $X \cap \text{PRJ}(\llbracket \phi \rrbracket, \text{VAR}(X), \mathcal{F}) \neq \emptyset$.

At this point, we proceed by a case analysis on the type of quantifier Q and the alternation flag α , where we exploit the fact that for every function $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$,

$$\text{EXT}(\text{EXT}(X, F, x), \mathcal{F}) = \text{EXT}(X, \mathcal{F}[x \mapsto F]).$$

Notice that, since \mathcal{F} is acyclic, $x \notin +\text{VAR}(\mathfrak{X})$, and $\text{DOM}(\mathcal{F}) \cap +\text{VAR}(\mathfrak{X}) = \emptyset$, it holds that $\mathcal{F}[x \mapsto F]$ is acyclic as well.

- [$Q = \exists$ & $\alpha = \exists\forall$] By Equivalence 1) and Item 1 of Lemma 2.7, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \exists x\phi$ iff there exist a function $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ and a team $X \in \mathfrak{X}$ such that $\text{EXT}(X, F, x) \subseteq \text{PRJ}(\llbracket \phi \rrbracket, \text{VAR}(X) \cup \{x\}, \mathcal{F})$, and thus $\text{EXT}(\text{EXT}(X, F, x), \mathcal{F}) \subseteq \llbracket \phi \rrbracket$. This means that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \exists x\phi$ iff there exist a function $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ and a team $X \in \mathfrak{X}$ such that $\text{EXT}(X, \mathcal{F}[x \mapsto F]) \subseteq \llbracket \phi \rrbracket$ iff there exists a function $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ such that $X' \subseteq \llbracket \phi \rrbracket$, for some team $X' \in \text{EXT}(\mathfrak{X}, \mathcal{F}[x \mapsto F])$. By Equivalence 1), the latter statement can be rewritten as: there exists a function $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ such that $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^{\exists\forall} \phi$; this in turn is equivalent to $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \Sigma^{+\text{VAR}(\mathfrak{X})} x\phi$, due to Item 9 of Definition 2.3. This concludes the proof of Item 1 for $\alpha = \exists\forall$.
- [$Q = \exists$ & $\alpha = \forall\exists$] By Equivalence 2) and Item 4 of Lemma 2.7, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \exists x\phi$ iff there exists a function $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ such that, for all teams $X \in \mathfrak{X}$,

- it holds true that $\text{EXT}(X, F, x) \cap \text{PRJ}(\llbracket \phi \rrbracket, \text{VAR}(X) \cup \{x\}, \mathcal{F}) \neq \emptyset$, and thus $\text{EXT}(\text{EXT}(X, F, x), \mathcal{F}) \cap \llbracket \phi \rrbracket \neq \emptyset$. This means that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \exists x \phi$ iff there exists a function $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ such that, for all teams $X \in \mathfrak{X}$, it holds that $\text{EXT}(X, \mathcal{F}[x \mapsto F]) \cap \llbracket \phi \rrbracket \neq \emptyset$ iff there exists a function $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ such that $X' \cap \llbracket \phi \rrbracket \neq \emptyset$, for all teams $X' \in \text{EXT}(\mathfrak{X}, \mathcal{F}[x \mapsto F])$. By Equivalence 2), the latter statement can be rewritten as: there exists a function $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ such that $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^{\forall\exists} \phi$; this in turn is equivalent to $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \Sigma^{+\text{VAR}(\mathfrak{X})} x \phi$, due to Item 9 of Definition 2.3. This concludes the proof of Item 1 for $\alpha = \forall\exists$.
- $[Q = \forall \ \& \ \alpha = \exists\forall]$ By Equivalence 1) and Item 3 of Lemma 2.7, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \forall x \phi$ iff, for all functions $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$, there exists a team $X \in \mathfrak{X}$ such that $\text{EXT}(X, F, x) \subseteq \text{PRJ}(\llbracket \phi \rrbracket, \text{VAR}(X) \cup \{x\}, \mathcal{F})$, and thus $\text{EXT}(\text{EXT}(X, F, x), \mathcal{F}) \subseteq \llbracket \phi \rrbracket$. This means that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \forall x \phi$ iff, for all functions $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$, there exists a team $X \in \mathfrak{X}$ such that $\text{EXT}(X, \mathcal{F}[x \mapsto F]) \subseteq \llbracket \phi \rrbracket$ iff, for all functions $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$, it holds that $X' \subseteq \llbracket \phi \rrbracket$, for some team $X' \in \text{EXT}(\mathfrak{X}, \mathcal{F}[x \mapsto F])$. By Equivalence 1), the latter statement can be rewritten as: for all functions $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$, it holds that $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^{\exists\forall} \phi$; this in turn is equivalent to $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\exists\forall} \Pi^{+\text{VAR}(\mathfrak{X})} x \phi$, due to Item 10 of Definition 2.3. This concludes the proof of Item 2 for $\alpha = \exists\forall$.
 - $[Q = \forall \ \& \ \alpha = \forall\exists]$ By Equivalence 2) and Item 2 of Lemma 2.7, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \forall x \phi$ iff, for all functions $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ and teams $X \in \mathfrak{X}$, it holds that $\text{EXT}(X, F, x) \cap \text{PRJ}(\llbracket \phi \rrbracket, \text{VAR}(\mathfrak{X}) \cup \{x\}, \mathcal{F}) \neq \emptyset$, and thus $\text{EXT}(\text{EXT}(X, F, x), \mathcal{F}) \cap \llbracket \phi \rrbracket \neq \emptyset$. This means that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \forall x \phi$ iff, for all functions $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$ and teams $X \in \mathfrak{X}$, it holds that $\text{EXT}(X, \mathcal{F}[x \mapsto F]) \cap \llbracket \phi \rrbracket \neq \emptyset$ iff, for all functions $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$, it holds that $X' \cap \llbracket \phi \rrbracket \neq \emptyset$, for all teams $X' \in \text{EXT}(\mathfrak{X}, \mathcal{F}[x \mapsto F])$. By Equivalence 2), the latter statement can be rewritten as: for all functions $F \in \text{Rsp}_{+\text{VAR}(\mathfrak{X})}$, it holds that $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^{\forall\exists} \phi$; this in turn is equivalent to $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^{\forall\exists} \Pi^{+\text{VAR}(\mathfrak{X})} x \phi$, due to Item 10 of Definition 2.3. This concludes the proof of Item 2 for $\alpha = \forall\exists$. □

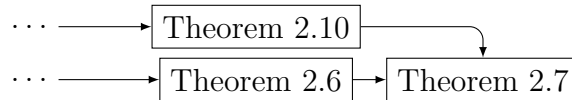


Figure 2.11 – Dependency graph of Theorem 2.7.

Theorem 2.7 (Herbrand-Skolem Theorem). *Let $\wp_1\wp_2\phi$ be an FO formula in pnf with quantifier prefix $\wp_1\wp_2 \in \mathbf{Q}$ and FO subformula ϕ . Then, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \wp_1\wp_2\phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \text{HSP}_\varphi(\wp_2)\wp_1\phi$, for all acyclic function assignments $\mathcal{F} \in \text{FunAsg}$ with $\text{DOM}(\mathcal{F}) \cap \text{DEP}(\wp_1\wp_2\phi) = \emptyset$ and hyperteams $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\wp_1\wp_2\phi) \setminus \text{DOM}(\mathcal{F}))$ with $\text{VAR}(\mathfrak{X}) \cap \text{VAR}(\wp_1\wp_2) = \emptyset$ and $\text{DOM}(\mathcal{F}) \cap \text{VAR}(\wp_1\wp_2) = \emptyset$.*

Proof. First, thanks to Corollary 2.1, we can assume that $\text{VAR}(\mathfrak{X}) = \text{FREE}(\wp_1\wp_2\phi) \setminus \text{DOM}(\mathcal{F})$. The proof proceeds by structural induction on the quantifier prefix $\wp_2 \in \mathbf{Q}$.

- [Base case $\wp_2 = \varepsilon$] Since $\text{HSP}_\varphi(\wp_2) = \varepsilon$, there is really nothing to prove as the statement is trivially true.
- [Inductive case $\wp_2 = \wp'Qx$] By Theorem 2.10, it holds that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \wp_1\wp_2\phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp')$ $\models^\alpha Qx\phi$. A case analysis on the type of quantifier is now required.
 - [$Q = \exists$] By Item 1 of Theorem 2.6, $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp')$ $\models^\alpha \exists x\phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp')$ $\models^\alpha \Sigma^{+\text{VAR}(\text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp'))}x\phi$, since ϕ is a quantifier-free FO formula. Thus, by Item 9 of Definition 2.3, we have that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \wp_1\wp_2\phi$ iff there exists a function $F \in \text{Rsp}_{+\text{VAR}(\text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp'))}$ such that $\mathcal{M}, \mathcal{F}[x \mapsto F], \text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp')$ $\models^\alpha \phi$. Again by Theorem 2.10, $\mathcal{M}, \mathcal{F}[x \mapsto F], \text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp')$ $\models^\alpha \phi$ is equivalent to $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^\alpha \wp_1\wp'\phi$, which in turn, by the inductive hypothesis applied to $\wp_1\wp'\phi$, is equivalent to $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^\alpha \text{HSP}_\varphi(\wp')\wp_1\phi$. Summing up, we have $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \wp_1\wp_2\phi$ iff there exists a function $F \in \text{Rsp}_{+\text{VAR}(\text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp'))}$ such that $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^\alpha \text{HSP}_\varphi(\wp')\wp_1\phi$. At this point, again by Item 9 of Definition 2.3, we obtain $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \wp_1\wp_2\phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \Sigma^{+\text{VAR}(\text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp'))}x\text{HSP}_\varphi(\wp')\wp_1\phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \text{HSP}_\varphi(\wp_2)\wp_1\phi$, where the latter equivalence is due to the definition of the HSP_φ function satisfying the equality $\text{HSP}_\varphi(\wp_2) = \text{HSP}_\varphi(\wp'\exists x) = \Sigma^{+\text{VAR}(\text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp'))}x\text{HSP}_\varphi(\wp')$. This concludes the proof of the existential case.
 - [$Q = \forall$] By Item 2 of Theorem 2.6, $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp')$ $\models^\alpha \forall x\phi$ iff $\mathcal{M}, \mathcal{F}, \text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp')$ $\models^\alpha \Pi^{+\text{VAR}(\text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp'))}x\phi$, since ϕ is a quantifier-free FO formula. Thus, by Item 10 of Definition 2.3, we have that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \wp_1\wp_2\phi$ iff, for all functions $F \in \text{Rsp}_{+\text{VAR}(\text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp'))}$, it holds that $\mathcal{M}, \mathcal{F}[x \mapsto F], \text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp')$ $\models^\alpha \phi$. By Theorem 2.10, $\mathcal{M}, \mathcal{F}[x \mapsto F], \text{EXT}_\alpha(\mathfrak{X}, \wp_1\wp')$ $\models^\alpha \phi$ is equivalent to $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^\alpha \wp_1\wp'\phi$, which in turn, by the inductive hypothesis applied to $\wp_1\wp'\phi$, is equivalent to $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^\alpha \text{HSP}_\varphi(\wp')\wp_1\phi$. Summing up, we have $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \wp_1\wp_2\phi$ iff, for all functions

$F \in \mathbf{Rsp}_{+\text{VAR}(\text{EXT}_\alpha(\mathfrak{x}, \wp_1 \wp'))}$, it holds that $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models^\alpha \text{HSP}_\varphi(\wp') \wp_1 \phi$. At this point, again by Item 10 of Definition 2.3, we obtain $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \wp_1 \wp_2 \phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \Pi^{+\text{VAR}(\text{EXT}_\alpha(\mathfrak{x}, \wp_1 \wp'))} x \text{HSP}_\varphi(\wp') \wp_1 \phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \text{HSP}_\varphi(\wp_2) \wp_1 \phi$, where the latter equivalence is due to the definition of the HSP_φ function satisfying the equality $\text{HSP}_\varphi(\wp_2) = \text{HSP}_\varphi(\wp' \forall x) = \Pi^{+\text{VAR}(\text{EXT}_\alpha(\mathfrak{x}, \wp_1 \wp'))} x \text{HSP}_\varphi(\wp')$. This concludes the proof of the universal case. \square

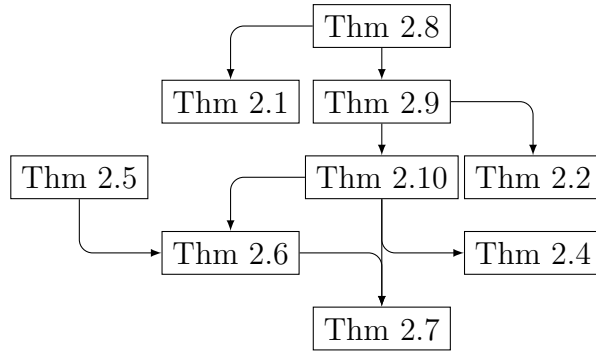


Figure 2.12 – Dependency graph of all Theorems.

ALTERNATING DEPENDENCE/INDEPENDENCE- FRIENDLY LOGIC

The hyperteam semantics is naturally suited to restrict quantifiers with uniformity constraints. In a DIF fashion, we can focus on response functions that are uniform with respect to a dependence constraints. For this chapter, we consider the same dependences/independences as in DIF, i.e. the value of a quantified variables depends on (or is independent from) the values of a set of variables W given at the quantification. We then define *Alternating Dependence/Independence-Friendly Logic* (ADIF for short), that generalizes DIF with negations. More involved dependencies will be considered in Chapter 4 and Chapter 5.

The chapter is organized as follows. In Section 3.1, we present the syntax and semantics of ADIF and also define **Meta-ADIF**, the **Meta** variant of ADIF. Then, in Section 3.2, we compare ADIF with FO, DIF and SO. Finally, in Section 3.3, we define the game-theoretic semantics of ADIF.

3.1 A symmetrical extension of DIF

Formally, given a dependence $\pm W$, a variable x and a hyperteam \mathfrak{X} , we define the $\pm W$ -uniform extension of \mathfrak{X} by $\text{EXT}_{\pm W}(\mathfrak{X}, x) = \{\text{EXT}(X, F, x) \mid F \in \text{Rsp}_{\pm W}\}$ (recall that $\text{Rsp}_{\pm W}$ has been defined in Definition 1.4).

Example 3.1. *Let $\mathfrak{X} = \{X_1 = \{\chi_1, \chi_2\}, X_2 = \{\chi_1, \chi_3\}\}$ be a hyperteam. To extend \emptyset -uniformly \mathfrak{X} with variable x over the structure domain $\mathbb{D} = \{0, 1\}$, one needs to extend*

each team in \mathfrak{X} with the two \emptyset -uniform (i.e., constant) functions $F_0(\chi) = 0$ and $F_1(\chi) = 1$:

$$\text{EXT}_{\emptyset}(\mathfrak{X}, x) = \left\{ \begin{array}{l} \text{EXT}(X_1, F_0, x) = \{\chi_1[x \mapsto 0], \chi_2[x \mapsto 0]\} \\ \text{EXT}(X_1, F_1, x) = \{\chi_1[x \mapsto 1], \chi_2[x \mapsto 1]\} \\ \text{EXT}(X_2, F_0, x) = \{\chi_1[x \mapsto 0], \chi_3[x \mapsto 0]\} \\ \text{EXT}(X_2, F_1, x) = \{\chi_1[x \mapsto 1], \chi_3[x \mapsto 1]\} \end{array} \right\}.$$

This new operator allows us to define *Alternating Dependence/Independence-Friendly Logic* (or ADIF for short) as follows.

3.1.1 Syntax.

The syntax of ADIF extends the one of FO by allowing dependence constraints specification on quantifiers. It can be seen as the closure of the DIF syntax with negation. For the whole chapter, we assume a relational signature \mathcal{S} (i.e. a signature with no function symbols). The following defines the syntax of ADIF .

Definition 3.1. *An ADIF formula is defined inductively as follows, where P is a predicate symbol, \vec{x} is a vector of variables of size $\text{AR}(P)$, x is a variable and W is a set of variables.*

$$\varphi := \perp \mid \top \mid P(\vec{x}) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists^{\pm W} x\varphi \mid \forall^{\pm W} x\varphi$$

Predicative logics usually rely on a notion of *free placeholder* to correctly define the meaning of a formula and ADIF is no exception. In ADIF, however, we distinguish between *support* and *free variables*. Specifically, support variables are the ones occurring in some atom $P(\vec{x})$ that needs to be assigned a value in order to evaluate the truth of the formula. The free variables, instead, also include those occurring in some dependence/independence constraint. By $\text{SUP}: \text{ADIF} \rightarrow 2^{\text{Var}}$ we denote the function collecting all support variables $\text{SUP}(\varphi)$ of a formula φ , defined as follows:

- $\text{SUP}(\perp), \text{SUP}(\top) \stackrel{\text{def}}{=} \emptyset$;
- $\text{SUP}(P(\vec{x})) \stackrel{\text{def}}{=} \vec{x}$;
- $\text{SUP}(\neg\varphi) \stackrel{\text{def}}{=} \text{SUP}(\varphi)$;
- $\text{SUP}(\varphi_1 \odot \varphi_2) \stackrel{\text{def}}{=} \text{SUP}(\varphi_1) \cup \text{SUP}(\varphi_2)$, for all connective symbols $\odot \in \{\wedge, \vee\}$;
- $\text{SUP}(Q^{\pm W} x\varphi) \stackrel{\text{def}}{=} \text{SUP}(\varphi) \setminus \{x\}$, for all quantifier symbols $Q \in \{\exists, \forall\}$.

The free-variable function $\text{FREE}: \text{ADIF} \rightarrow 2^{\text{Var}}$ is defined similarly, except for the quantifier case, which is reported in the following:

- $\text{FREE}(Q^{\pm W}x\varphi) \stackrel{\text{def}}{=} (\text{FREE}(\varphi) \setminus \{x\}) \cup \llbracket \pm W \rrbracket$, if $x \in \text{FREE}(\varphi)$, and
- $\text{FREE}(Q^{\pm W}x\varphi) \stackrel{\text{def}}{=} \text{FREE}(\varphi)$, otherwise, for all quantifier symbols $Q \in \{\exists, \forall\}$,

with $\llbracket \pm W \rrbracket$ denoting the set W , for the symbol ‘+’, and its complement $\text{Var} \setminus W$, for the symbol ‘-’.

Obviously, it holds that $\text{SUP}(\varphi) \subseteq \text{FREE}(\varphi)$. A *sentence* ϕ is a formula such that $\text{FREE}(\phi) = \emptyset$. If $\text{SUP}(\phi) = \emptyset$, but $\text{FREE}(\phi) \neq \emptyset$, then ϕ is just a *pseudo sentence*. As an example, $\phi = \forall^{+\emptyset}x\exists^{+\emptyset}y(x = y)$ is a sentence, while $\phi' = \forall^{+\emptyset}x\exists^{+z}y(x = y)$ is a pseudo sentence, since $\text{SUP}(\phi') = \emptyset$, but $\text{FREE}(\phi') = \{z\}$. Another example of pseudo sentence is $\phi'' = \forall x\exists^{-x}y(x = y)$. In general, every formula with empty support and containing a quantifier of the form $Q^{-W}x$ is clearly a pseudo sentence. We also define $\exists x\varphi \stackrel{\text{def}}{=} \exists^{+W}x\varphi$ and $\forall x\varphi \stackrel{\text{def}}{=} \forall^{+W}x\varphi$, where $W \stackrel{\text{def}}{=} \text{SUP}(\varphi) \setminus \{x\}$. Notice that, with the last two quantifiers, FO is a syntactic fragment of ADIF and furthermore, for FO formulas, it holds that $\text{SUP}(\varphi) = \text{FREE}(\varphi)$. As we shall show in Section 3.2.1 by resorting to the result presented in Section 2.4, this fragment semantically corresponds to classic FO as defined in Section 1.1. Similarly, we shall later identify a richer fragment of ADIF that semantically corresponds to IFL as formalised in Section 1.2.

We define *alternating dependence logic* (ADF for short) as the fragment where only \exists^{+W} and \forall^{+W} quantifiers are considered and symmetrically, *alternating independence-friendly logic* (AIF for short) is the fragment where only \exists^{-W} and \forall^{-W} quantifiers are considered.

Before giving the formal definition of the compositional semantics, it is worth providing just few examples of properties expressible in ADIF.

Let us picture a two-turn game where Player 1, who chooses first, controls the variable x and Player 2, who chooses second, controls y . Let $\psi(x, y)$ be the goal of Player 2 and consider the following two ADF sentences:

$$\phi_1 \stackrel{\text{def}}{=} \forall x\exists^{+x}y\psi(x, y); \quad \phi_2 \stackrel{\text{def}}{=} \exists x\forall^{+x}y\neg\psi(x, y).$$

Sentence ϕ_1 , whenever true, requires Player 2, in this case Eloise, to be able to respond to every choice for x made by Player 1, in this case Abelard, so that goal $\psi(x, y)$ is always satisfied. This corresponds to the existence of a winning strategy for Eloise, namely a strategy that wins every induced play in the game, for the objective $\psi(x, y)$. On the contrary, with inverted roles, the truth of ϕ_2 ensures that there is a choice of Eloise such that, no matter what Abelard chooses, $\psi(x, y)$ cannot be achieved. This means that

Abelard cannot have a winning strategy for $\psi(x, y)$. If ϕ_2 is false, instead, it is Abelard who has a winning strategy for $\psi(x, y)$, while the falsity of ϕ_1 ensures the existence of a choice of Abelard such that, no matter what Eloise chooses, $\psi(x, y)$ cannot be achieved. Note that both sentences belong to the FO fragment introduced above and their semantics also corresponds to the Tarskian one. However, the ADF sentences

$$\phi_3 \stackrel{\text{def}}{=} \forall x \exists^{+\emptyset} y \psi(x, y); \quad \phi_4 \stackrel{\text{def}}{=} \exists x \forall^{+\emptyset} y \neg \psi(x, y)$$

add imperfect information to the picture. Sentence ϕ_3 still postulates the existence of a winning strategy for Eloise, but this time also requires that, when making the choice for y , the player has no access to any information and, in particular, to the value chosen for x by the opponent. We call such a strategy \emptyset -uniform. Similarly, ϕ_4 , when true, witnesses the non-existence of such a \emptyset -uniform winning strategy for Abelard. The ADIF pseudo sentences

$$\phi_5 \stackrel{\text{def}}{=} \forall x \exists^{-x} y \psi(x, y); \quad \phi_6 \stackrel{\text{def}}{=} \exists x \forall^{-x} y \neg \psi(x, y)$$

have a very similar meaning to ϕ_3 and ϕ_4 , respectively, with the exception that y , while still required to be independent of x , may now depend on any variable different from x . Indeed, $\text{FREE}(\phi_5) = \text{FREE}(\phi_6) = \text{Var} \setminus x$, hence, in principle, y can depend on any of these free variables. As a general rule, a quantifier $Q^{-W}w$ occurring inside a formula φ allows w to depend on any free variable of φ that is not in the set W .

Consider now a three-turn game, extending the ϕ_4 one, where, after the move of Player 2, Player 1 chooses the value for another variable under its control, let us call this z . In the following, we consider a goal subformula that is a conjunction of the two goals $\psi_1(x, y)$ and $\psi_2(y, z)$. The interpretation of the ADF sentence

$$\phi_7 \stackrel{\text{def}}{=} \exists x \forall^{+\emptyset} y \exists^{+x} z (\psi_1(x, y) \wedge \psi_2(y, z))$$

is a bit more involved. First of all, it states that Player 2, i.e., Abelard, cannot see the choice made for x . In addition, while Player 1, i.e., Eloise, is not aware of y , she has access to the value previously chosen for x by herself. The sentence, whenever true, ensures the existence of a choice by Eloise which ensures that Abelard cannot prevent $\psi_1(x, y)$ from happening, no matter what he chooses. Moreover, Eloise can respond to any of these latter choices for y and win objective $\psi_2(y, z)$ by only looking at the value of x . This means that

Abelard is not able to prevent $\psi_1(x, y)$ and, at the same time, Eloise has an x -uniform strategy to win $\psi_2(y, z)$.

3.1.2 Semantics

The semantics relation of ADIF is the same as the hyperteam semantics of Definition 2.1 except for the rules for quantifiers.

Definition 3.2 (Hyperteam Semantics). *The ADIF semantic relation $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$ is the same as Definition 2.1 except for Items 7 and 8 that are replaced with the following with ϕ an ADIF formulas, $\mathfrak{X} \in \text{HypTeams}_\subseteq(\text{FREE}(\varphi))$ a hyperteam and x a variable:*

- γ' (a) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \exists^{\pm W} x\phi$ if $\mathcal{M}, \text{EXT}_{\pm W}(\mathfrak{X}, x) \models^{\exists\forall} \phi$;
- (b) $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \exists^{\pm W} x\phi$ if $\mathcal{M}, \bar{\mathfrak{X}} \models^{\exists\forall} \exists^{\pm W} x\phi$;
- 3.2' (a) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \forall^{\pm W} x\phi$ if $\mathcal{M}, \bar{\mathfrak{X}} \models^{\forall\exists} \forall^{\pm W} x\phi$;
- (b) $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \forall^{\pm W} x\phi$ if $\mathcal{M}, \text{EXT}_{\pm W}(\mathfrak{X}, x) \models^{\forall\exists} \phi$.

3.1.3 Examples

To familiarise with the proposed compositional semantics of ADIF, we now present few examples of evaluation of formulas via a step by step unravelling of all the semantic rules involved.

Example 3.2. *Consider the sentence $\phi_4 = \exists x \forall^{+\emptyset} y \neg \psi(x, y)$ from above, where we instantiate $\psi(x, y)$ as $(x = y)$. We evaluate ϕ_4 in the binary structure $\mathcal{M} = \langle \{0, 1\}, \{=\mathcal{M}\} \rangle$ against the trivial hyperteam $\{\{\emptyset\}\}$. The alternation flag is of no consequence, since $\{\{\emptyset\}\}$ is self-dual (see Proposition 2.1), hence, we can choose $\alpha = \exists\forall$, without loss of generality. We want to check whether $\mathcal{M}, \{\{\emptyset\}\} \models^{\exists\forall} \phi_4$. The semantic rule for the existential quantifier $\exists x$ requires to compute the extension $\text{EXT}_\emptyset(\{\{\emptyset\}\}, x)$ of $\{\{\emptyset\}\}$. This results in*

$$\mathcal{M}, \{\{\emptyset\}\} \models^{\exists\forall} \exists x \forall^{+\emptyset} y \neg(x = y) \quad \text{iff} \quad \mathcal{M}, \mathfrak{X} \models^{\exists\forall} \forall^{+\emptyset} y \neg(x = y),$$

where $\mathfrak{X} = \{\{x:0\}, \{x:1\}\}$. The rule for the universal quantifier $\forall^{+\emptyset} y$ requires to dualise the hyperteam and switch the flag to $\forall\exists$. Since every team of \mathfrak{X} is a singleton, there is only one possible choice function, thus, the result is

$$\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \forall^{+\emptyset} y \neg(x = y) \quad \text{iff} \quad \mathcal{M}, \bar{\mathfrak{X}} \models^{\forall\exists} \forall^{+\emptyset} y \neg(x = y),$$

where $\bar{\mathfrak{X}} = \{\{x:0, x:1\}\}$. Now the quantifier $\forall^{+\emptyset}y$ and the alternation flag $\forall\exists$ are coherent, and we extend the hyperteam to obtain $\text{EXT}_{\emptyset}(\bar{\mathfrak{X}}, y)$, where only constant functions can be used for the extensions, since y cannot depend on x . The result is, then,

$$\mathcal{M}, \bar{\mathfrak{X}} \models^{\forall\exists} \forall^{+\emptyset}y \neg(x = y) \quad \text{iff} \quad \mathcal{M}, \text{EXT}_{\emptyset}(\bar{\mathfrak{X}}, y) \models^{\forall\exists} \neg(x = y),$$

where $\text{EXT}_{\emptyset}(\bar{\mathfrak{X}}, y) = \left\{ \left\{ \begin{array}{l} x:0 \ x:1 \\ y:0 \ y:0 \end{array} \right\}, \left\{ \begin{array}{l} x:0 \ x:1 \\ y:1 \ y:1 \end{array} \right\} \right\}$. The rule for the negation operation \neg dualises the flag and, in addition, requires the hyperteam $\text{EXT}_{\emptyset}(\bar{\mathfrak{X}}, y)$ not to satisfy the atom $(x = y)$ under $\exists\forall$. This means that every team in $\text{EXT}_{\emptyset}(\bar{\mathfrak{X}}, y)$ must contain an assignment that falsifies the atom. But this is indeed the case, since every team has an assignment χ such that $\chi(x) \neq \chi(y)$. Hence, ϕ_4 evaluates to true in \mathcal{M} against $\{\{\emptyset\}\}$. Observe that, on the contrary, the sentence $\phi_3 = \forall x \exists^{+\emptyset}y \psi(x, y)$ from above evaluates to false in \mathcal{M} against $\{\{\emptyset\}\}$, being equivalent to the negation of ϕ_4 . Indeed, in this case, following the semantic rules for the quantifiers, we would still end up with the same hyperteam $\text{EXT}_{\emptyset}(\bar{\mathfrak{X}}, y)$ against which we need to evaluate the subformula $x = y$. However, this time the alternation flag would be $\exists\forall$ and, as we already noted above, every team in $\text{EXT}_{\emptyset}(\bar{\mathfrak{X}}, y)$ contains one assignment falsifying $x = y$.

We anticipate here a game-theoretic intuition of truth and falsity in ADIF on the simpler case of sentences in *prenex normal form* and with a single alternation of quantifiers – Recall that a formula is in prenex normal form when it is of the form $\varphi\phi$ with φ a quantifier prefix and ϕ a quantifier-free ADIF formula. The interpretation of such sentences can be viewed as a challenge-response game, where the player associated with the first type of quantifier in the prefix is the challenger and the other one the responder. The idea is that for the responder to win the game, they must win the subformula (either satisfy it if she is the existential player or falsify it if he is the universal one) while adhering to some uniform strategy, i.e., a strategy compatible with the (in)dependence constraints on their variables. If they cannot, the challenger wins. In a sense, this satisfaction game places on the responder the burden of proof that they are able to successfully play according to the constraints and win the subformula. When the challenger wins the challenge-response game, then the formula is considered true if she is the existential player, and false if he is the universal one. This is why, for instance, the two sentences ϕ_4 and ϕ_3 , discussed in Section 3.1.1, are true and false, respectively. Indeed, in ϕ_4 the responder is the universal player controlling variable y . Since y cannot depend on anything, it must be chosen

uniformly regardless of the value of x . Clearly, that player does not have a uniform strategy that falsifies the subformula $\neg(x = y)$, which makes the sentence won by the existential player and, therefore, true. By a similar reasoning, the responder in ϕ_3 is the existential player controlling y and cannot access the value of x . Hence, that player does not have a uniform strategy to satisfy the subformula $(x = y)$ either. Therefore, the universal player, who is the challenger, wins the sentence, which makes it false.

For sentences in prenex normal form with more than one alternations, though, the truth and falsity conditions in ADIF become more complicated, since the two players may act both as a challenger and as a responder against different variables. In this case, one needs to take into consideration the uniformity constraints of both players and who is ultimately responsible for breaking the (in)dependence constraints to try and win the subformula. Here is also where the symmetry requirement on the players comes into play in a more significant way, as for both truth and falsity one needs to take into account the restrictions of the two players at the same time. We refer the reader to Section 3.3 for the full presentation of the game-theoretic semantics of ADIF, in which the intuitions discussed above are made precise.

Example 3.3. Consider the pseudo sentence $\phi_6 = \exists x \forall^{-x} y \neg \psi(x, y)$ from above, where again we instantiate $\psi(x, y)$ as $(x = y)$. The exact same reasoning followed in Example 3.2 shows that ϕ_6 is true in \mathcal{M} against the trivial hyperteam $\{\{\emptyset\}\}$. Consequently, the pseudo sentence $\phi_5 = \forall x \exists^{-x} y \psi(x, y)$ is false in \mathcal{M} against $\{\{\emptyset\}\}$, being equivalent to the negation of ϕ_6 . These two pseudo sentences, however, are not equivalent to the sentences ϕ_4 and ϕ_3 , respectively. To see this, let us evaluate ϕ_5 in \mathcal{M} against the hyperteam $\mathfrak{X} = \{\{z:0, z:1\}\}$ w.r.t the alternation flag $\alpha = \forall \exists$. Note that $z \in \text{FREE}(\phi_5) = \llbracket -x \rrbracket = \text{Var} \setminus \{x\}$. The semantic rule for $\forall x$ requires to compute the extension $\text{EXT}_\emptyset(\mathfrak{X}, x)$ of \mathfrak{X} . This results in

$$\mathcal{M}, \mathfrak{X} \models^{\forall \exists} \forall x \exists^{-x} y (x = y) \quad \text{iff} \quad \mathcal{M}, \text{EXT}_\emptyset(\mathfrak{X}, x) \models^{\forall \exists} \exists^{-x} y (x = y),$$

where $\text{EXT}_\emptyset(\mathfrak{X}, x) = \left\{ \left\{ \begin{array}{cc} z:0 & z:1 \\ x:0 & x:0 \end{array} \right\}, \left\{ \begin{array}{cc} z:0 & z:1 \\ x:1 & x:1 \end{array} \right\} \right\}$. The rule for $\exists^{-x} y$ requires to dualise the hyperteam and switch the flag to $\exists \forall$. Since both teams in $\text{EXT}_\emptyset(\mathfrak{X}, x)$ contain two assignments, there are four choice functions in total, leading to

$$\mathcal{M}, \text{EXT}_\emptyset(\mathfrak{X}, x) \models^{\forall \exists} \exists^{-x} y (x = y) \quad \text{iff} \quad \mathcal{M}, \overline{\text{EXT}_\emptyset(\mathfrak{X}, x)} \models^{\exists \forall} \exists^{-x} y (x = y), \text{ where}$$

$$\overline{\text{EXT}_\emptyset(\mathfrak{X}, x)} = \left\{ \begin{array}{l} X_1 = \quad X_2 = \quad X_3 = \quad X_4 = \\ \left\{ \begin{array}{l} z:0 \ z:0 \\ x:0 \ x:1 \end{array} \right\}, \left\{ \begin{array}{l} z:0 \ z:1 \\ x:0 \ x:1 \end{array} \right\}, \left\{ \begin{array}{l} z:1 \ z:0 \\ x:0 \ x:1 \end{array} \right\}, \left\{ \begin{array}{l} z:1 \ z:1 \\ x:0 \ x:1 \end{array} \right\} \end{array} \right\}.$$

The extension $\widehat{\mathfrak{X}} \stackrel{\text{def}}{=} \text{EXT}_{\text{Var} \setminus x}(\overline{\text{EXT}_\emptyset(\mathfrak{X}, x)}, y) = \text{EXT}_{\{z\}}(\overline{\text{EXT}_\emptyset(\mathfrak{X}, x)}, y)$ with the four functions that can only depend on z , happens to contains 12 teams. Each response functions is denoted by F_{ij} with $i, j \in \{0, 1\}$ and are defined by $F_{ij}(\chi) = \begin{cases} i & \text{if } \chi(z) = 0 \\ j & \text{otherwise} \end{cases}$

$$\text{EXT}_{-x}(\overline{\text{EXT}_\emptyset(\mathfrak{X}, x)}, y) = \left(\begin{array}{l} \text{EXT}(X_1, F_{00}, y) = \text{EXT}(X_2, F_{00}, y) = \text{EXT}(X_3, F_{00}, y) = \text{EXT}(X_4, F_{00}, y) = \\ \left\{ \begin{array}{l} z:0 \ z:0 \\ x:0, x:1 \\ y:0 \ y:0 \end{array} \right\}, \left\{ \begin{array}{l} z:0 \ z:1 \\ x:0, x:1 \\ y:0 \ y:0 \end{array} \right\}, \left\{ \begin{array}{l} z:1 \ z:0 \\ x:0, x:1 \\ y:0 \ y:0 \end{array} \right\}, \left\{ \begin{array}{l} z:1 \ z:1 \\ x:0, x:1 \\ y:0 \ y:0 \end{array} \right\} \\ \text{EXT}(X_1, F_{01}, y) = \text{EXT}(X_2, F_{01}, y) = \text{EXT}(X_3, F_{01}, y) = \text{EXT}(X_4, F_{01}, y) = \\ \left\{ \begin{array}{l} z:0 \ z:0 \\ x:0, x:1 \\ y:0 \ y:0 \end{array} \right\}, \left\{ \begin{array}{l} z:0 \ z:1 \\ x:0, x:1 \\ y:0 \ y:1 \end{array} \right\}, \left\{ \begin{array}{l} z:1 \ z:0 \\ x:0, x:1 \\ y:1 \ y:0 \end{array} \right\}, \left\{ \begin{array}{l} z:1 \ z:1 \\ x:0, x:1 \\ y:1 \ y:1 \end{array} \right\} \\ \text{EXT}(X_1, F_{10}, y) = \text{EXT}(X_2, F_{10}, y) = \text{EXT}(X_3, F_{10}, y) = \text{EXT}(X_4, F_{10}, y) = \\ \left\{ \begin{array}{l} z:0 \ z:0 \\ x:0, x:1 \\ y:1 \ y:1 \end{array} \right\}, \left\{ \begin{array}{l} z:0 \ z:1 \\ x:0, x:1 \\ y:1 \ y:0 \end{array} \right\}, \left\{ \begin{array}{l} z:1 \ z:0 \\ x:0, x:1 \\ y:0 \ y:1 \end{array} \right\}, \left\{ \begin{array}{l} z:1 \ z:1 \\ x:0, x:1 \\ y:0 \ y:0 \end{array} \right\} \\ \text{EXT}(X_1, F_{11}, y) = \text{EXT}(X_2, F_{00}, y) = \text{EXT}(X_3, F_{00}, y) = \text{EXT}(X_4, F_{11}, y) = \\ \left\{ \begin{array}{l} z:0 \ z:0 \\ x:0, x:1 \\ y:1 \ y:1 \end{array} \right\}, \left\{ \begin{array}{l} z:0 \ z:1 \\ x:0, x:1 \\ y:1 \ y:1 \end{array} \right\}, \left\{ \begin{array}{l} z:1 \ z:0 \\ x:0, x:1 \\ y:1 \ y:1 \end{array} \right\}, \left\{ \begin{array}{l} z:1 \ z:1 \\ x:0, x:1 \\ y:1 \ y:1 \end{array} \right\} \end{array} \right).$$

Notice that even if 16 teams are displayed, some are redundant: we have $\text{EXT}(X_1, F_{00}, y) = \text{EXT}(X_1, F_{01}, y)$ and $\text{EXT}(X_1, F_{11}, y) = \text{EXT}(X_1, F_{10}, y)$ and $\text{EXT}(X_4, F_{00}, y) = \text{EXT}(X_4, F_{10}, y)$ and $\text{EXT}(X_4, F_{11}, y) = \text{EXT}(X_4, F_{01}, y)$, which makes only 12 different teams.

We find $X \stackrel{\text{def}}{=} \text{EXT}(X_2, F_{01}, y) = \{\chi_1, \chi_2\}$, where $\chi_1(z) = \chi_1(x) = \chi_1(y) = 0$, $\chi_2(z) = \chi_2(x) = \chi_2(y) = 1$, and $F(\chi) = \chi(z)$. Now, the final step requires checking whether $\mathcal{M}, \widehat{\mathfrak{X}} \models^{\exists\forall} (x = y)$. Since every assignment in X satisfies $(x = y)$, the pseudo sentence is proved true in \mathcal{M} against \mathfrak{X} w.r.t the alternation flag $\alpha = \exists\forall$. Intuitively, z allows for

some leak of information and, by mimicking x , is able to pass the required information to y in order to satisfy the subformula. This leak of information is only possible because x is independent from z . As an immediate consequence, ϕ_6 evaluates to false in \mathcal{M} against \mathfrak{X} . Instead, it is possible to show that the evaluations of ϕ_3 and ϕ_4 remain unchanged on \mathfrak{X} , i.e., they are again false and true, respectively, due to the fact that they are sentences (this is a direct consequence of Corollary 2.1).

The above example should clarify the reasoning behind the choice of the name *pseudo sentences*, for those formulas φ with $\text{SUP}(\varphi) = \emptyset$, but $\text{FREE}(\varphi) \neq \emptyset$. As for sentences, a pseudo sentence can be verified against an arbitrary hyperteam; however, similarly to formulas, its truth may depend on the specific hyperteam.

Example 3.4. Consider the sentence $\phi_7 = \exists x \forall^{+\emptyset} y \exists^{+x} z (\psi_1(x, y) \wedge \psi_2(y, z))$ from above, where we instantiate $\psi_1(x, y)$ as $(x = y)$ and $\psi_2(y, z)$ as $(y = z)$. We evaluate this sentence in the same structure \mathcal{M} of the previous examples and the trivial hyperteam $\{\{\emptyset\}\}$. Observe also that ϕ_7 shares most of the quantifier prefix of sentence ϕ_4 in Example 3.2. As a consequence, by applying the same steps as before, we end up with the following equivalence:

$$\mathcal{M}, \{\{\emptyset\}\} \models^{\exists\forall} \phi_7 \text{ iff } \mathcal{M}, \text{EXT}_0(\overline{\mathfrak{X}}, y) \models^{\forall\exists} \exists^{+x} z (x = y) \wedge (y = z),$$

where $\text{EXT}_0(\overline{\mathfrak{X}}, y) = \left\{ \left\{ \begin{array}{cc} x:0 & x:1 \\ y:0 & y:0 \end{array} \right\}, \left\{ \begin{array}{cc} x:0 & x:1 \\ y:1 & y:1 \end{array} \right\} \right\}$. Applying the rule for $\exists^{+x} z$ requires dualisation first, leading to

$$\mathcal{M}, \{\{\emptyset\}\} \models^{\exists\forall} \phi_7 \text{ iff } \mathcal{M}, \overline{\text{EXT}_0(\overline{\mathfrak{X}}, y)} \models^{\exists\forall} \exists^{+x} z (x = y) \wedge (y = z), \text{ where}$$

$$\overline{\text{EXT}_0(\overline{\mathfrak{X}}, y)} = \left\{ \begin{array}{cccc} X_1 = & X_2 = & X_3 = & X_4 = \\ \left\{ \begin{array}{cc} x:0 & x:0 \\ y:0 & y:1 \end{array} \right\}, & \left\{ \begin{array}{cc} x:0 & x:1 \\ y:0 & y:1 \end{array} \right\}, & \left\{ \begin{array}{cc} x:1 & x:0 \\ y:0 & y:1 \end{array} \right\}, & \left\{ \begin{array}{cc} x:1 & x:1 \\ y:0 & y:1 \end{array} \right\} \end{array} \right\}.$$

The extension $\widehat{\mathfrak{X}} \stackrel{\text{def}}{=} \text{EXT}_{\{x\}}(\overline{\text{EXT}_0(\overline{\mathfrak{X}}, y)}, z)$ can only use functions that depend on x alone and there are four of them. Similarly to the previous example, the hyperteam $\widehat{\mathfrak{X}}$ ends up containing 16 teams. Among these teams one can find $X \stackrel{\text{def}}{=} \text{EXT}(X_2, F, z) = \{\chi_1, \chi_2\}$, where $\chi_1(x) = \chi_1(y) = \chi_1(z) = 0$, $\chi_2(x) = \chi_2(y) = \chi_2(z) = 1$, and $F(\chi) = \chi(x)$. Now, the final step requires checking whether $\mathcal{M}, \widehat{\mathfrak{X}} \models^{\exists\forall} (x = y) \wedge (y = z)$. By the rule for the conjunction connective, this is true if $\mathcal{M}, \widehat{\mathfrak{X}}_1 \models^{\exists\forall} (x = y)$ or $\mathcal{M}, \widehat{\mathfrak{X}}_2 \models^{\exists\forall} (y = z)$, for all

bipartitions $(\hat{\mathfrak{X}}_1, \hat{\mathfrak{X}}_2) \in \text{PART}(\hat{\mathfrak{X}})$. Obviously, any such partition would contain X either in $\hat{\mathfrak{X}}_1$ or in $\hat{\mathfrak{X}}_2$. Since every assignment in X satisfies both $(x = y)$ and $(y = z)$, the sentence is proved true in \mathcal{M} against $\{\{\emptyset\}\}$.

3.1.4 Properties

By the similarity between Definition 2.1 of hyperteam semantics for FO and Definition 3.2 of ADIF semantics, ADIF inherits all results stated in Section 2.3, such as Lemma 2.3 that works for every ADIF formula φ and any hyperteam $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{SUP}(\varphi))$.

Lemma 3.1 (Empty & Null Hyperteams). *The following hold true for every ADIF formula φ and hyperteam $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{SUP}(\varphi))$:*

1. (a) $\mathcal{M}, \emptyset \not\models^{\exists\forall} \varphi$;
 (b) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$, where $\emptyset \in \mathfrak{X}$;
2. (a) $\mathcal{M}, \emptyset \models^{\forall\exists} \varphi$;
 (b) $\mathcal{M}, \mathfrak{X} \not\models^{\forall\exists} \varphi$, where $\emptyset \in \mathfrak{X}$.

The core theorem that is Theorem 2.1, stating that the \sqsubseteq preorder on hyperteam corresponds to a semantic implication, is adapted below. Notice how the hyperteam are required only to be defined on the set of variables $\text{SUP}(\varphi)$ whereas the \sqsubseteq regards all variables in $\text{FREE}(\varphi)$.

Theorem 3.1 (Hyperteam Refinement). *Let φ be an ADIF formula and $\mathfrak{X}, \mathfrak{X}' \in \text{HypTeams}_{\supseteq}(\text{SUP}(\varphi))$ two hyperteam with $\mathfrak{X} \sqsubseteq_{\text{FREE}(\varphi)} \mathfrak{X}'$. Then:*

1. if $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$ then $\mathcal{M}, \mathfrak{X}' \models^{\exists\forall} \varphi$;
2. if $\mathcal{M}, \mathfrak{X}' \models^{\forall\exists} \varphi$ then $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi$.

To prove this theorem, we require an extension of the monotonicity results for \sqsubseteq on the various operations on hyperteam to the new $\text{EXT}_{\pm W}$ operation. The full proof is given in Section 3.A. This is the only result for which the proof differ from the previous chapter.

With this result, we can adapt Theorem 2.2 and Theorem 2.4 as follows.

Theorem 3.2 (Double Dualisation). *For every ADIF formula φ , alternation flag α and hyperteam $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{SUP}(\varphi))$, it holds that $\mathcal{M}, \mathfrak{X} \models^{\alpha} \varphi$ iff $\mathcal{M}, \overline{\overline{\mathfrak{X}}} \models^{\alpha} \varphi$ iff $\mathcal{M}, \overline{\mathfrak{X}} \models^{\bar{\alpha}} \varphi$.*

Theorem 3.3 (Prefix Extension). *Let $\wp\phi$ be an ADIF formula, where $\wp \in \mathbf{Q}$ is a quantifier prefix and ϕ is an arbitrary FO formula. Then, $\mathcal{M}, \mathfrak{X} \models^\alpha \wp\phi$ iff $\mathcal{M}, \text{EXT}_\alpha(\mathfrak{X}, \wp) \models^\alpha \phi$, for all hypertteams $\mathfrak{X} \in \text{HypTeams}_\supseteq(\text{SUP}(\wp\phi))$.*

Currently, we do not know whether ADIF does enjoy a *prenex normal form* (pnf, for short). For this reason, for the rest of this section and in Section 3.3, we shall mainly consider formulas that are already in pnf.

Open problem 1 (ADIF Prenex Normal Form). *Is every ADIF formula equivalent to an ADIF formula in pnf?*

As for the previous chapter, we now generalise the extension operator to quantifier prefixes \wp , whose set is denoted by \mathbf{Q} . Notice that, *without loss of generality*, we only consider prefixes where each variable x

1. is quantified at most once,
2. does not occur in the dependence/independence constraint set $\llbracket \pm W \rrbracket$ of its quantifier $Q^{\pm W}x$, and
3. cannot be quantified in the scope of a quantifier $Q^{\pm W}y$ whose dependence/independence constraint set $\llbracket \pm W \rrbracket$ includes x itself.

With $\text{VAR}(\wp)$ and DEP_\wp we denote the set of variables quantified in \wp and the union of all dependence/independence constraint sets occurring in \wp , respectively. Given a hyperteam \mathfrak{X} and an alternation flag α , the operator $\text{EXT}_\alpha(\mathfrak{X}, \wp)$ corresponds to iteratively applying the extension operator to \mathfrak{X} , for all quantifiers occurring in \wp , in that specific order. To this end, we first introduce the notion of *coherence* of a quantifier symbol $Q \in \{\exists, \forall\}$ with an alternation flag $\alpha \in \{\exists\forall, \forall\exists\}$ as follows: Q is α -coherent if either $\alpha = \exists\forall$ and $Q = \exists$ or $\alpha = \forall\exists$ and $Q = \forall$. Now, the application of a quantifier $Q^{\pm W}x$ to \mathfrak{X} , denoted by $\text{EXT}_\alpha(\mathfrak{X}, Q^{\pm W}x)$, follows the semantics of quantifiers, as defined in Definition 3.2. More precisely, it just corresponds to the extension of \mathfrak{X} with x , when Q is α -coherent. Conversely, when Q is $\bar{\alpha}$ -coherent, we need to dualise the extension with x of the dual of \mathfrak{X} . Formally:

$$\text{EXT}_\alpha(\mathfrak{X}, Q^{\pm W}x) \stackrel{\text{def}}{=} \begin{cases} \text{EXT}_{\pm W}(\mathfrak{X}, x), & \text{if } Q \text{ is } \alpha\text{-coherent;} \\ \overline{\text{EXT}_{\pm W}(\bar{\mathfrak{X}}, x)}, & \text{otherwise.} \end{cases}$$

The operator naturally lifts to arbitrary quantification prefixes \wp :

1. $\text{EXT}_\alpha(\mathfrak{X}, \epsilon) \stackrel{\text{def}}{=} \mathfrak{X}$;
2. $\text{EXT}_\alpha(\mathfrak{X}, Q^{\pm W} x \varphi) \stackrel{\text{def}}{=} \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, Q^{\pm W} x), \varphi)$.

We also define $\text{EXT}_\alpha(\varphi) \stackrel{\text{def}}{=} \text{EXT}_\alpha(\{\{\emptyset\}\}, \varphi)$. A simple structural induction on a quantifier prefix $\wp \in \mathbf{Q}$, shows that a hyperteam \mathfrak{X} α -satisfies a formula $\wp\phi$ iff its α -extension *w.r.t* \wp α -satisfies its subformula ϕ .

Theorem 3.4 (Prefix Extension). *Let $\wp\phi$ be an ADIF formula, where $\wp \in \mathbf{Q}$ is a quantifier prefix and ϕ is an arbitrary ADIF formula. Then, $\mathcal{M}, \mathfrak{X} \models^\alpha \wp\phi$ iff $\mathcal{M}, \text{EXT}_\alpha(\mathfrak{X}, \wp) \models^\alpha \phi$, for all hyperteam $\mathfrak{X} \in \text{HypTeams}_\subseteq(\text{FREE}(\wp\phi))$.*

3.1.5 Meta semantics

The Meta logic of Section 2.5 can be adapted for ADIF in Meta-ADIF by adding the meta quantifiers to the syntax of ADIF.

Definition 3.3 (Meta-ADIF Syntax). *The Meta Extension of ADIF (Meta-ADIF, for short) is the set of formulas built according to Definition 1.2 extended as follows, where $x \in V$:*

$$\varphi \stackrel{\text{def}}{=} \text{ADIF} \mid \Sigma^{+W} x. \varphi \mid \Pi^{+W} x. \varphi.$$

We need to redefine the support and the free variables to account for the meta quantifiers. The support variables $\text{SUP}(\varphi)$ of a Meta-ADIF formula φ simply generalize the definition for ADIF formulas as follows:

- $\text{SUP}(\perp), \text{SUP}(\top) \stackrel{\text{def}}{=} \emptyset$;
- $\text{SUP}(\mathbf{P}(\vec{x})) \stackrel{\text{def}}{=} \vec{x}$;
- $\text{SUP}(\neg\varphi) \stackrel{\text{def}}{=} \text{SUP}(\varphi)$;
- $\text{SUP}(\varphi_1 \odot \varphi_2) \stackrel{\text{def}}{=} \text{SUP}(\varphi_1) \cup \text{SUP}(\varphi_2)$, for all connective symbols $\odot \in \{\wedge, \vee, \rightarrow\}$;
- $\text{SUP}(Q^{\pm W} x \varphi) \stackrel{\text{def}}{=} \text{SUP}(\varphi) \setminus \{x\}$, for all quantifier symbols $Q \in \{\exists, \forall, \Sigma, \Pi\}$.

The definition of free variables is, instead, quite more intricate and requires the introduction of the following supplemental functions of *free variables under meta dependency context* $\text{FREE}: \text{Meta-ADIF} \times (\text{Var} \rightarrow 2^{\text{Var}}) \rightarrow 2^{\text{Var}}$ and *dependence variables under meta dependency context* $\text{DEP}: \text{Meta-ADIF} \times (\text{Var} \rightarrow 2^{\text{Var}}) \rightarrow 2^{\text{Var}}$ – recall that a dependency context is a partial function $\iota \in \text{Var} \rightarrow 2^{\text{Var}}$. The *transitive closure* of ι is a dependency context $\iota^* \in \text{DOM}(\iota) \rightarrow 2^{\text{Var}}$ such that, for each variable $x \in \text{DOM}(\iota)$ in its domain, $\iota^*(x)$ is the smallest set of variables such that

1. $\iota(x) \subseteq \iota^*(x)$ and

2. $\iota(y) \subseteq \iota^*(x)$, for all variables $y \in \iota^*(x) \cap \text{DOM}(\iota)$.

Finally, ι is *acyclic* if $x \notin \iota^*(x)$, for all variables $x \in \text{DOM}(\iota)$. The functions FREE and DEP can be defined in a mutual recursive fashion as follows.

- $\text{FREE}(\perp, \iota) \stackrel{\text{def}}{=} \text{FREE}(\top, \iota) \stackrel{\text{def}}{=} \emptyset$;
- $\text{FREE}(\mathbf{P}(\vec{x}), \iota) \stackrel{\text{def}}{=} \vec{x} \cup \bigcup \{\iota^*(x) \mid x \in \vec{x} \cap \text{DOM}(\iota)\}$;
- $\text{FREE}(\neg\varphi, \iota) \stackrel{\text{def}}{=} \text{FREE}(\varphi, \iota)$;
- $\text{FREE}(\varphi_1 \odot \varphi_2, \iota) \stackrel{\text{def}}{=} \text{FREE}(\varphi_1, \iota) \cup \text{FREE}(\varphi_2, \iota)$, for $\odot \in \{\wedge, \vee\}$;
- $\text{FREE}(Qx\varphi, \iota) \stackrel{\text{def}}{=} \begin{cases} \text{FREE}(\varphi, \iota') \setminus \{x\} & \text{if } x \in \text{FREE}(\varphi, \iota'), \\ \text{FREE}(\varphi, \iota') & \text{otherwise} \end{cases}$
 where $\iota' \stackrel{\text{def}}{=} \iota \setminus \{x\}$, for $Q \in \{\exists, \forall\}$;
- $\text{FREE}(Q^{\pm W}x\varphi, \iota) \stackrel{\text{def}}{=} \begin{cases} \text{FREE}(\varphi, \iota') & \text{if } x \in \text{DEP}(\varphi, \iota'), \\ \text{FREE}(\varphi, \iota') \setminus \{x\} & \text{otherwise} \end{cases}$
 where $\iota' \stackrel{\text{def}}{=} \iota[x \mapsto \llbracket \pm W \rrbracket]$, for $Q \in \{\Sigma, \Pi\}$.

Intuitively, a variable y can be free in a Meta-ADIF formula φ under a dependency context ι only for one (or more) of the following two reasons:

1. it is explicitly used in some relational symbol;
2. it occurs in the (transitive) dependency set $\iota^*(x)$ of some meta quantified variable x used in a relational symbol;

Notice that a meta quantifier of a variable x masks such a variable only if it does not appear in the set of dependence variables of its subformula.

- $\text{DEP}(\perp, \iota), \text{DEP}(\top, \iota), \text{DEP}(\mathbf{P}(\vec{x}), \iota) \stackrel{\text{def}}{=} \emptyset$;
- $\text{DEP}(\neg\varphi, \iota) \stackrel{\text{def}}{=} \text{DEP}(\varphi, \iota)$;
- $\text{DEP}(\varphi_1 \odot \varphi_2, \iota) \stackrel{\text{def}}{=} \text{DEP}(\varphi_1, \iota) \cup \text{DEP}(\varphi_2, \iota)$, for $\odot \in \{\wedge, \vee\}$;
- $\text{DEP}(Q^{\pm W}x\varphi, \iota) \stackrel{\text{def}}{=} \begin{cases} (\text{DEP}(\varphi, \iota') \setminus \{x\}) \cup \llbracket \pm W \rrbracket & \text{if } x \in \text{FREE}(\varphi, \iota'), \\ \text{DEP}(\varphi, \iota') & \text{otherwise} \end{cases}$
 where $\iota' \stackrel{\text{def}}{=} \iota \setminus \{x\}$, for $Q \in \{\exists, \forall\}$;
- $\text{DEP}(Q^{\pm W}x\varphi, \iota) \stackrel{\text{def}}{=} \text{DEP}(\varphi, \iota')$, where $\iota' \stackrel{\text{def}}{=} \iota[x \mapsto \llbracket \pm W \rrbracket]$, for $Q \in \{\Sigma, \Pi\}$.

Intuitively, a variable y belongs to the set $\text{DEP}(\varphi, \iota)$ if it appears in the dependence/independence constraint set $\llbracket \pm W \rrbracket$ of some first-order quantifier $Q^{\pm W}x$ of a free variable x and, at the same time, is not removed, i.e., is not under the scope of another first-order quantifier for y itself. Notice that the dependencies of the variables quantified by a meta-quantifier, which are maintained by the dependency context ι , are not taken into account here, as they are only used to determine which variables are free. At this point, the sets of

free variables $\text{FREE}(\varphi)$ and dependence variables $\text{DEP}(\varphi)$ of a Meta formula φ are defined as $\text{FREE}(\varphi, \emptyset)$ and $\text{DEP}(\varphi, \emptyset)$, respectively.

The semantics of Meta-ADIF simply extends the semantics of ADIF in Definition 3.2 in the same way Meta semantics of Definition 2.3 extends the hyperteam semantics Definition 2.1.

Definition 3.4 (Meta-ADIF Semantics). *The hyperteam semantic relation $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\alpha} \varphi$ for Meta-ADIF is inductively defined as follows, for all Meta-ADIF formulas φ , function assignments $\mathcal{F} \in \text{FunAsg}$, hyperteams $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi) \setminus \text{DOM}(\mathcal{F}))$, and alternation flags $\alpha \in \{\exists\forall, \forall\exists\}$:*

- 1,2,4-8) All ADIF cases, but those ones of the atomic relations, are defined by lifting, in the obvious way, the corresponding items of Definition 3.2 to function assignments, i.e., the latter play no role;
3. (a) $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\exists\forall} P(\vec{x})$ if there exists a team $X \in \text{EXT}(\mathfrak{X}, \mathcal{F})$ such that, for all assignments $\chi \in X$, it holds that $\chi(\vec{x}) \in P^{\mathcal{M}}$;
- (b) $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\forall\exists} P(\vec{x})$ if, for all teams $X \in \text{EXT}(\mathfrak{X}, \mathcal{F})$, there exists an assignment $\chi \in X$ such that $\chi(\vec{x}) \in P^{\mathcal{M}}$;
9. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\alpha} \Sigma^{+W} x. \phi$ if $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models_{\text{Meta}}^{\alpha} \phi$, for some $F \in \text{Rsp}_{+W}$;
10. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\alpha} \Pi^{+W} x. \phi$ if $\mathcal{M}, \mathcal{F}[x \mapsto F], \mathfrak{X} \models_{\text{Meta}}^{\alpha} \phi$, for all $F \in \text{Rsp}_{+W}$.

We can retrieve the results of Section 2.5, starting with the quantifier translation between ADIF quantifiers $\exists^{\pm W}$ and $\forall^{\pm W}$ to $\Sigma^{\pm W}$ and $\Pi^{\pm W}$ respectively. Remark that we still require the sub formula ϕ to be an FO formula.

Theorem 3.5 (ADIF Quantifier Interpretation). *The following equivalences hold true, for all FO formulas ϕ , variables $x \in V$, sets of variables $W \subseteq V$ with $x \notin W$, acyclic function assignments $\mathcal{F} \in \text{FunAsg}$ with $\text{DOM}(\mathcal{F}) \cap W = \emptyset$, and hyperteams $\mathfrak{X} \in \text{HypTeams}_{\supseteq}((\text{SUP}(\phi) \setminus \{x\}) \setminus \text{DOM}(\mathcal{F}))$ with $x \notin \text{VAR}(\mathfrak{X})$:*

1. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\alpha} \exists^{\pm W} x \phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\alpha} \Sigma^{+W} x \phi$;
2. $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\alpha} \forall^{\pm W} x \phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^{\alpha} \Pi^{+W} x \phi$.

We then proceed to extend the function HSP to constrained quantifiers in the following way, where \wp is a quantifier prefix, x a variable and W a set of variables.

1. $\text{HSP}(\varepsilon) \stackrel{\text{def}}{=} \varepsilon$;
2. $\text{HSP}(\wp. \exists^{\pm W} x) \stackrel{\text{def}}{=} \Sigma^{+W} x. \text{HSP}(\wp)$;

3. $\text{HSP}(\wp. \forall^{\pm W} x) \stackrel{\text{def}}{=} \Pi^{+W} x. \text{HSP}(\wp)$.

Remark that we do not need to specify any index formula for HSP as the translation keeps the constraints of the first-order quantifier. However, HSP is still a generalization of HSP_φ since the classic first-order quantifier \exists is defined as $\exists^{+\text{SUP}(\varphi)}$ with φ being the subformula of the quantifier.

We can now state the Herbrand-Skolem Theorem for ADIF.

Theorem 3.6 (Herbrand-Skolem Theorem). *Let $\wp_1 \wp_2 \phi$ be an ADIF formula in pnf with quantifier prefix $\wp_1 \wp_2 \in \mathbf{Q}$ and FO subformula ϕ . Then, $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \wp_1 \wp_2 \phi$ iff $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models_{\text{Meta}}^\alpha \text{HSP}(\wp_2) \wp_1 \phi$, for all acyclic function assignments $\mathcal{F} \in \text{FunAsg}$ with $\text{DOM}(\mathcal{F}) \cap \text{DEP}(\wp_1 \wp_2 \phi) = \emptyset$ and hyperteams $\mathfrak{X} \in \text{HypTeams}_\subseteq(\text{SUP}(\wp_1 \wp_2 \phi) \setminus \text{DOM}(\mathcal{F}))$ with $\text{VAR}(\mathfrak{X}) \cap \text{VAR}(\wp_1 \wp_2) = \emptyset$ and $\text{DOM}(\mathcal{F}) \cap \text{VAR}(\wp_1 \wp_2) = \emptyset$.*

Example 3.5. *Let us consider again the sentence from Example 3.4, i.e., $\phi_7 = \exists x \forall^{+\emptyset} y \exists^{+x} z (x = y) \wedge (y = z)$. We already saw that the sentence is true in the original binary structure \mathcal{M} of the same example. If we convert ϕ_7 into Meta-ADIF via the function HSP, we obtain $\Sigma^{+x} z \Pi^{+\emptyset} y \Sigma^{+\emptyset} x (x = y) \wedge (y = z)$. To show this sentence true in \mathcal{M} , it suffices to assign to z the identity function that copies the value assigned to x . Then, whatever value is chosen for y , the same value can be assigned to x . By the semantics of Meta-ADIF, the result immediately follows.*

Thanks to this Herbrand/Skolem-isation procedure, we can transform an ADIF sentence in pnf into a Meta-ADIF sentence in pnf, where only the meta-quantifiers $\Sigma^{+W} x$ and $\forall x$ occur. Since one needs only polynomial space in the size of the underlying structure to represent the quantified functions, the same approach used for FO model checking is also applicable here.

Theorem 3.7 (Model-Checking Problem). *Let \mathcal{M} be a finite structure and ϕ an ADIF sentence in pnf. Then, the model-checking problem $\mathcal{M} \models \phi$ can be decided in PSPACE w.r.t $|\mathcal{M}|$.*

As is the case of ADIF, we do not know whether Meta-ADIF enjoys a *prenex normal form*, even when we only take into consideration the two meta quantifiers Σ^{+W} and Π^{+W} .

Open problem 2 (Meta-ADIF Prenex Normal Form). *Is every Meta-ADIF formula equivalent to a Meta-ADIF formula in pnf?*

3.2 Comparisons to other logics

In this section, we compare ADIF with other well known logics: first we retrieve the adequacy theorem for FO from previous chapter; then we extend this adequacy for both DFL_{\exists} and DFL_{\forall} (and the IFL variants too); finally, we show that ADIF is expressively equivalent to the *Second-Order Logic* (SO for short).

3.2.1 First-order Adequacy

The adequacy of ADIF with FO comes from Theorem 2.5 of previous chapter.

Theorem 3.8 (FO Adequacy (ADIF)). *For all FO formulas φ and hyperteams $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi))$, it holds that:*

1. $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$ iff there exists a team $X \in \mathfrak{X}$ such that, for all assignments $\chi \in X$, it holds that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$;
2. $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi$ iff, for all teams $X \in \mathfrak{X}$, there exists an assignment $\chi \in X$ such that $\mathcal{M}, \chi \models_{\text{FO}} \varphi$.

To prove this theorem, we first restrict ourselves to hyperteams $\mathfrak{X} \in \text{HypTeams}(\text{FREE}(\varphi))$ thanks to Corollary 2.1. Then, we remark that for every hyperteam \mathfrak{X} and variable x , we have $\text{EXT}(\mathfrak{X}, x) = \text{EXT}_{\text{VAR}(\mathfrak{X})}(\mathfrak{X}, x)$ and the rest of the proof is the same as the one for Theorem 2.5.

3.2.2 Dependence/Independence-Friendly Logic Adequacy

As already mentioned, DIF and its fragments can be seen as syntactic fragments of ADIF. We show here that the semantics of those fragments coincide with the one presented in Section 1.2.

In order to show that ADIF is indeed a conservative extension of DIF, we need to be able to simulate the semantics on teams with hyperteams. As a first step, we lift the cylindrification operator to hyperteams in the obvious way, by defining $\text{CYL}(\mathfrak{X}, x) \stackrel{\text{def}}{=} \{\text{CYL}(X, x) \mid X \in \mathfrak{X}\}$. While the semantics of ADIF does not provide a primitive operator for cylindrification, this operation can easily be simulated by first dualising the hyperteam, then by applying the extension for x uniformly over all the variables in the domain of \mathfrak{X} , and, finally dualising the result again. The following lemma establishes the equivalence of these two different operations.

Lemma 3.2 (Cylindrical Extension). *Let $\mathfrak{X} \in \text{HypTeams}$ be a hyperteam. Then, $\text{CYL}(\mathfrak{X}, x) \equiv \overline{\text{EXT}_{\pm W}(\mathfrak{X}, x)}$, for all variables $x \in \text{Var}$ and sets of variables W , with $\text{VAR}(\mathfrak{X}) \subseteq W \subseteq \text{Var}$.*

A similar problem arises with the team partitioning operator that is not present in the semantics of ADIF. Once again, the dualisation operator, together with the hyperteam partitioning operator, allows us to simulate it. More specifically, we first apply the dualisation of the hyperteam \mathfrak{X} , then the partitioning to obtain $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$, and, finally, dualise the two resulting hyperteam and obtain $\overline{\mathfrak{X}_1}$ and $\overline{\mathfrak{X}_2}$, each of which happens to contain teams that would result from the team partitioning operation applied to the teams in \mathfrak{X} .

Lemma 3.3 (Team Partitioning). *Let $\mathfrak{X} \in \text{HypTeams}$ be a hyperteam. Then:*

1. *for all hyperteam bipartitions $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ and teams $X'_1 \in \overline{\mathfrak{X}_1}$ and $X'_2 \in \overline{\mathfrak{X}_2}$, there exists a team $X \in \mathfrak{X}$ such that $X \subseteq X'_1 \cup X'_2$;*
2. *for all teams $X \in \mathfrak{X}$ and team bipartitions $(X_1, X_2) \in \text{PART}(X)$, there exist a hyperteam bipartition $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ and two teams $X'_1 \in \overline{\mathfrak{X}_1}$ and $X'_2 \in \overline{\mathfrak{X}_2}$ such that $X'_1 \subseteq X_1$ and $X'_2 \subseteq X_2$.*

Based on these two lemmas, we can prove the following theorem, which establishes the required adequacy result.

Theorem 3.9 (DIF Adequacy). *For all DIF formulas φ and hyperteam $\mathfrak{X} \in \text{HypTeams}_{\subseteq}(\text{SUP}(\varphi))$, it holds that:*

1. *if φ is DIF_{\exists} then $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$ iff there is a team $X \in \mathfrak{X}$ such that $\mathcal{M}, X \models^{\forall}_{\text{DIF}} \varphi$;*
2. *if φ is DIF_{\forall} then $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi$ iff, for all teams $X \in \mathfrak{X}$, it holds that $\mathcal{M}, X \models^{\exists}_{\text{DIF}} \varphi$.*

From now on, for every DIF formula φ , we denote by φ_{\exists} and φ_{\forall} the DIF_{\exists} and DIF_{\forall} variants obtained from φ by removing the constraints on the universal and existential quantifiers, respectively, i.e., by substituting $-\emptyset$ for the variable restrictions of such quantifiers. Recall that, [Hod97b] (see also [MSS11]) defines an IFL sentence φ to be *true* in a structure \mathcal{M} , in symbols $\mathcal{M} \models_{\text{IFL}} \varphi$, if $\mathcal{M}, \{\emptyset\} \models^{\forall}_{\text{DIF}} \varphi$, and *false* in \mathcal{M} , namely $\mathcal{M} \not\models_{\text{IFL}} \varphi$, if $\mathcal{M}, \{\emptyset\} \models^{\exists}_{\text{DIF}} \varphi$. As observed above, this means that φ is *true* in \mathcal{M} , if $\mathcal{M}, \{\emptyset\} \models^{\forall}_{\text{DIF}} \varphi_{\exists}$, and *false* in \mathcal{M} , if $\mathcal{M}, \{\emptyset\} \not\models^{\exists}_{\text{DIF}} \varphi_{\forall}$. Therefore, thanks to Theorem 3.9, we can assert the following.

Remark 3.1. *For every DIF-sentence φ , we have that:*

- $\mathcal{M} \models_{IFL} \varphi$ iff $\mathcal{M}, \{\{\emptyset\}\} \models^{\exists\forall} \varphi_{\exists}$, i.e., $\mathcal{M} \models \varphi_{\exists}$, and
- $\mathcal{M} \not\models_{IFL} \varphi$ iff $\mathcal{M}, \{\{\emptyset\}\} \not\models^{\forall\exists} \varphi_{\forall}$, i.e., $\mathcal{M} \not\models \varphi_{\forall}$.

The following example illustrates the connection between ADIF and DIF.

Example 3.6. In Example 3.2, it has been observed that the two ADIF sentences $\varphi_3 = \forall x \exists^{+\emptyset} y (x = y)$ and $\varphi_4 = \exists x \forall^{+\emptyset} y \neg (x = y)$ evaluate to false and true, respectively, in the binary structure $\mathcal{M} = \langle \{0, 1\}, \{=\mathcal{M}\} \rangle$ against the trivial hyperteam $\{\{\emptyset\}\}$. We also claimed that they are the semantic negation of each other, something that now can be easily proved thanks to Corollary 2.5 and Theorem 2.3. Note that all these properties hold true for the two DIF $_{\exists}$ and DIF $_{\forall}$ sentences $\varphi_3' = \forall^{-\emptyset} x \exists^{+\emptyset} y (x = y)$ and $\varphi_4' = \exists^{-\emptyset} x \forall^{+\emptyset} y \neg (x = y)$ as well. At this point, we can show that the truth and falsity of φ_3' and φ_4' convey different meanings when evaluated in IFL (equivalently, DIF). Both φ_3' and φ_4' are IFL sentences. Moreover, as previously stated, φ_3' is an DIF $_{\exists}$ sentence, while φ_4' is a DIF $_{\forall}$ sentence. Thus, from Remark 3.1, we immediately obtain that, when evaluated in IFL, φ_3' is not true and φ_4' is not false. However, again by Remark 3.1, φ_3' is not false and φ_4' is not true either, since $\mathcal{M}, \{\{\emptyset\}\} \models \varphi_{3\forall}'$ and $\mathcal{M}, \{\{\emptyset\}\} \not\models \varphi_{4\exists}'$. Therefore, the two sentences are undetermined.

The considerations discussed above allows us to characterise elegantly in ADIF some meta-properties of IFL sentences, such as *indeterminacy* and *sensitivity to signalling phenomena*. These results witness the expressive advantages of ADIF over IFL and substantiate the intuition that ADIF can be thought of as a logic to reason *about* imperfect information, as opposed to IFL, which can be viewed more as a language to reason *with* imperfect information.

Let us start with indeterminacy of IFL sentences first. [Hod97b] defines an IFL sentence φ to be *undetermined* in a structure \mathcal{M} if it is neither true nor false in \mathcal{M} . Hence, an immediate application of Remark 3.1 gives us the following corollary.

Corollary 3.1 (Definability of IFL-Indeterminacy). *For every IFL sentence φ , let φ_u be the ADIF pseudo sentence $\neg\varphi_{\exists} \wedge \varphi_{\forall}$. Then, it holds that*

$$\mathcal{M} \models \varphi_u \text{ iff } \varphi \text{ is undetermined in } \mathcal{M}.$$

The second phenomenon is called *signalling* [Hod97b; MSS11]. In game-theoretic terms, the phenomenon arises in situations where, for instance, one of the existential (*resp.*, universal) players can store inside one of his variables, say variable z , the value of

some variable x of the opponent that another existential (*resp.*, universal) player is not allowed to see. However, by merely being able to access the value of z , this last player can infer the value of the forbidden variable x and choose a response accordingly.

The logical analogue of this phenomenon is captured in IFL by forms of *information leaks*, where information about the value of a variable may leak toward another variable by means of a third, possibly unused, one. The typical example of this phenomenon already emerges in the simple IFL sentence $\forall x \exists^{-x} y (x = y)$. Clearly, Eloise, who cannot see the value of x when choosing the value for y , does not have a uniform winning strategy to satisfy for equality. Since also Abelard does not have one to falsify it, the formula is undetermined in IFL. However, the sentence $\forall x \exists z \exists^{-x} y (x = y)$, where the dummy quantifier for z has been added, becomes determined, and specifically true. The reason is that now Eloise, who intuitively represents the team of existential players, does have a winning strategy. Indeed, when choosing z , she is allowed to see the value of x and can just copy that value onto z . This time, however, when choosing the value of y , while she still has no direct access to the value of x , she does have indirect access to its value through z , which she is allowed to see. The winning move here is then to copy whatever value is inside z onto y to satisfy the equality.

In general, then, we say that an IFL sentence φ is *sensitive to signalling w.r.t* some variables not in $\text{SUP}(\varphi)$, if the introduction of vacuous quantifiers over them in φ changes its truth value. For sentences in prenex normal form this means that, if we change the quantifier prefix \wp with one of its extensions $\hat{\wp}$, then the two sentences $\wp\psi$ and $\hat{\wp}\psi$ have different truth values. In IFL, this may only happen when φ is undetermined, while its extension $\hat{\varphi}$ is determined. In other words, either φ is not true, while $\hat{\varphi}$ is true, or φ is not false, while $\hat{\varphi}$ is false. Once again, by applying Remark 3.1, we obtain the following.

Corollary 3.2 (Definability of IFL-Signalling). *Let $\varphi = \wp\psi$ be an IFL sentence in pnf with quantifier prefix $\wp \in \mathbf{Q}$ and quantifier-free subformula ψ . Moreover, let $\hat{\wp} \in \mathbf{Q}$ be a quantifier prefix extending \wp and $\varphi_s^{\hat{\wp}}$ the ADIF pseudo sentence $(\neg\varphi_{\exists} \wedge \hat{\varphi}_{\exists}) \vee (\varphi_{\forall} \wedge \neg\hat{\varphi}_{\forall})$, with $\hat{\varphi} \stackrel{\text{def}}{=} \hat{\wp}\psi$. Then, it holds that*

$$\mathcal{M} \models \varphi_s^{\hat{\wp}} \text{ iff } \varphi \text{ is sensitive to signalling in } \mathcal{M} \text{ w.r.t } \hat{\wp}.$$

It is important to observe here that the ability of ADIF to restrict both the universal and existential quantifiers at the same time, that is to treat the two players in a completely symmetric way, is essential to characterise the above definability properties. Both φ_u and

$\varphi_s^{\hat{\phi}}$, on the other hand, are undetermined in IFL.

It is also worth remarking that the hyperteam semantics and pseudo sentences interact in quite a peculiar way, giving rise to a new form of information leak, separate from the one occurring in connection with signalling and dummy quantifiers. This is evidenced by the pseudo sentences φ_5 and φ_6 of Example 3.3. We showed there that $\mathcal{M}, \{\{\emptyset\}\} \not\models^{\exists\forall} \varphi_5$ and $\mathcal{M}, \{\{\emptyset\}\} \models^{\forall\exists} \varphi_6$, hence, $\mathcal{M}, \{\{\emptyset\}\} \not\models^{\forall\exists} \varphi_5$ and $\mathcal{M}, \{\{\emptyset\}\} \models^{\exists\forall} \varphi_6$. However, the example also shows that $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi_5$ and $\mathcal{M}, \mathfrak{X} \not\models^{\exists\forall} \varphi_6$, where $\mathfrak{X} = \{\{z:0, z:1\}\}$. Here, the information on z contained in \mathfrak{X} may leak into y through the hyperteam. Observe that hyperteam \mathfrak{X} can be obtained by means of a suitable dummy quantification of variable z and, therefore, we immediately obtain that $\mathcal{M}, \{\{\emptyset\}\} \models^{\exists\forall} \exists z \varphi_5$ and $\mathcal{M}, \{\{\emptyset\}\} \not\models^{\forall\exists} \forall z \varphi_6$. As a consequence, introducing a dummy quantifier for a variable that is free but not in the support of a pseudo sentence can change the truth value, even if such a variable cannot depend on any other variable. Note that this specific form of information leak does not actually reflect any signalling phenomenon in the classic game-theoretic sense and does not occur in IFL either.

3.2.3 Second-Order & Team Logics

We have previously shown that ADIF is a conservative extension of DIF. However, its game-theoretic determinacy gives us a considerably more expressive logic than DIF, with a full-fledged second-order flavour, even in the absence of a contradictory negation. Indeed, the meta-theory interpretation allows us to show that every SO and TL formula can be interpreted in the ADF fragment of ADIF. *Vice versa*, every ADF formula, over a restricted class of hyperteam, can be interpreted by corresponding SO sentences and TL formulas. This implies that, from a descriptive-complexity viewpoint, ADF formulas cover at least the entire polynomial hierarchy.

Every non-null hyperteam $\mathfrak{X} \in \text{HypTeams}(\vec{x})$ defined over a sequence of variables $\vec{x} \in V^*$, which is *at most equipotent* to the domain of the underlying structure \mathcal{M} , i.e., $|\mathfrak{X}| \leq |\mathcal{M}|$, can be encoded by a k -ary relation symbol $P_{\mathfrak{X}}$, with $k \stackrel{\text{def}}{=} |\vec{x}| + 1$, whose interpretation $P_{\mathfrak{X}}^{\mathcal{M}} \subseteq \mathbb{D}^k$ is defined (up to isomorphism) as follows: for every team $X \in \mathfrak{X}$, there is an element $v \in \mathbb{D}$ and, *vice versa*, for every element $v \in \mathbb{D}$, there is a team $X \in \mathfrak{X}$ such that

$$\chi \in X \text{ iff } \mathcal{M} \uplus \{P_{\mathfrak{X}}^{\mathcal{M}}\}, \chi[y \mapsto v] \models_{\text{FO}} P_{\mathfrak{X}}(\vec{x}y),$$

for all assignments $\chi \in \text{Asg}(\vec{x})$. Otherwise said, by $|\mathfrak{X}| \leq |\mathcal{M}|$, we can have a sur-

jection $f_{\text{surj}} : \mathbb{D} \rightarrow \mathfrak{X}$ and thanks to this surjection, we define $\mathbf{P}_{\mathfrak{X}}^{\mathcal{M}} \stackrel{\text{def}}{=} \{(v_1, \dots, v_k) \mid \chi_{v_1, \dots, v_{k-1}} \in f_{\text{surj}}(v_k)\}$ where $\chi_{v_1, \dots, v_{k-1}}$ maps x_i to v_i for $i \in \{1, \dots, k-1\}$.

Such an interpretation $\mathbf{P}_{\mathfrak{X}}^{\mathcal{M}}$ is later on called $\text{REL}(\mathfrak{X})$. It is not clear whether there exist other relational encodings of hypertteams with greater (possibly infinite) cardinality than the domain of the structure. Now, by Theorem 3.6, every ADF formula in pnf can be translated into an equivalent **Meta** formula, where the semantics of the meta quantifiers can be easily modelled via second-order quantifications. This leads to the result below, which implies that every ADF-definable hypertteam (under the above restriction) is SO-definable.

Theorem 3.10 (ADF-SO Interpretation). *For every ADF formula φ in pnf with quantifier prefix $\wp \in \mathbf{Q}$ over a signature \mathcal{S} , set of variables $\text{SUP}(\varphi) \subseteq V \subseteq \mathbf{Var}$ with $V \cap \text{VAR}(\varphi) = \emptyset$, and relation symbol $P_{\mathfrak{X}} \notin \mathcal{S}$ with $\text{AR}(P_{\mathfrak{X}}) = |V| + 1$, there exist two SO sentences $\psi_{\exists\forall}$ and $\psi_{\forall\exists}$ over signature $\mathcal{S} \uplus \{P_{\mathfrak{X}}\}$ such that, for all \mathcal{S} -structures \mathcal{M} and non-null hypertteams $\mathfrak{X} \in \text{HypTeams}(V)$ with $|\mathfrak{X}| \leq |\mathcal{M}|$, the following equivalence holds true: $\mathcal{M}, \mathfrak{X} \models^{\alpha} \varphi$ iff $\mathcal{M} \uplus \{\text{REL}(\mathfrak{X})\} \models_{\text{SO}} \psi_{\alpha}$.*

Using a similar approach, every non-empty non-null hypertteam $\mathfrak{X} \in \text{HypTeams}(V)$ defined over a set of variables $V \subseteq V$, with $|\mathfrak{X}| \leq |\mathcal{M}|$, can be encoded in a team $\text{TEAM}(\mathfrak{X}, y) \in \text{Teams}(V \cup y)$, with $y \notin V$, as follows: for every team $X \in \mathfrak{X}$, there is an element $v \in \mathbb{D}$ and, *vice versa*, for every element $v \in \mathbb{D}$, there is a team $X \in \mathfrak{X}$ such that

$$\chi \in X \text{ iff } \chi[y \mapsto v] \in \text{TEAM}(\mathfrak{X}, y),$$

for all assignments $\chi \in \text{Asg}(V)$. Since every SO-definable relation can be encoded in a TL-definable team [KV09; KN09], the next result easily follows from the previous one.

Corollary 3.3 (ADF-TL Interpretation). *For every ADF formula φ in pnf with quantifier prefix $\wp \in \mathbf{Q}$, set of variables $\text{SUP}(\varphi) \subseteq V \subseteq V$ with $V \cap \text{VAR}(\varphi) = \emptyset$, and variable $y \notin V \cup \text{VAR}(\varphi)$, there exist two TL formulas $\phi_{\exists\forall}$ and $\phi_{\forall\exists}$ with $\text{FREE}(\phi_{\exists\forall}) = \text{FREE}(\phi_{\forall\exists}) = V \cup y$ such that, for all structures \mathcal{M} and non-empty non-null hypertteams $\mathfrak{X} \in \text{HypTeams}(V)$ with $|\mathfrak{X}| \leq |\mathcal{M}|$, the following equivalence holds true: $\mathcal{M}, \mathfrak{X} \models^{\alpha} \varphi$ iff $\mathcal{M}, \text{TEAM}(\mathfrak{X}, y) \models_{\text{TL}} \phi_{\alpha}$.*

It is unknown whether the above two interpretation results still hold when the constraint $|\mathfrak{X}| \leq |\mathcal{M}|$ on the size of the hypertteam and the domain of the structure is violated.

Open problem 3 (ADF-SO/TL Interpretations). *Is it possible to obtain interpretation results in a similar vein to Theorem 3.10 and Corollary 3.3, when $|\mathfrak{X}| > |\mathcal{M}|$?*

In addition, it is not clear what the distinguishability power of ADF is *w.r.t* the cardinality of the hypertteams, especially in the infinite case.

Open problem 4 (Hyperteam Cardinality). *Is there an ADF satisfiable formula φ such that, if $\mathcal{M}, \mathfrak{X} \models^\alpha \varphi$, then $|\mathfrak{X}| > |\mathcal{M}| \geq \omega$, for some $\alpha \in \{\exists\forall, \forall\exists\}$?*

For the converse direction of the interpretation results, given an \mathcal{S} -structure \mathcal{M} , a relation symbol $P_{\mathfrak{X}} \in \mathcal{S}$, and a sequence of variables $\vec{x} \in V^{\text{AR}(P_{\mathfrak{X}})}$, we denote by $\text{TEAM}(P_{\mathfrak{X}}^{\mathcal{M}}, \vec{x}) \in \text{Teams}(\vec{x})$ the standard encoding in a team (up to isomorphism) of the interpretation $P_{\mathfrak{X}}^{\mathcal{M}}$ of $P_{\mathfrak{X}}$ defined as follows:

$$\chi \in \text{TEAM}(P_{\mathfrak{X}}^{\mathcal{M}}, \vec{x}) \text{ iff } \mathcal{M}, \chi \models_{\text{FO}} P_{\mathfrak{X}}(\vec{x}),$$

for all assignments $\chi \in \text{Asg}(\vec{x})$. Every SO sentence can be put in a canonical form, where every quantification over functions can be simulated by a meta quantifier that only depends on the variables to which the function is applied. Thus, by exploiting Theorem 3.6, the result below can be proved.

Theorem 3.11 (SO-ADF Interpretation). *For every SO sentence ψ over a signature \mathcal{S} , relation symbol $P_{\mathfrak{X}} \in \mathcal{S}$, and sequence of variables $\vec{x} \in V^{\text{AR}(P_{\mathfrak{X}})}$, with $\text{VAR}(\psi) \cap \vec{x} = \emptyset$, i.e., no variable in \vec{x} occurs in ψ , there exists an ADF formula φ in pnf over signature $\mathcal{S} \setminus P_{\mathfrak{X}}$ with $\text{SUP}(\varphi) = \text{FREE}(\varphi) = \vec{x}$ such that, for all \mathcal{S} -structures \mathcal{M} , the following equivalence holds true: $\mathcal{M} \models_{\text{SO}} \psi$ iff $\mathcal{M} \setminus P_{\mathfrak{X}}, \{\text{TEAM}(P_{\mathfrak{X}}^{\mathcal{M}}, \vec{x})\} \models^{\exists\forall} \varphi$.*

By using the translation from TL to SO (see [Vää07; Har79], for the sentences, and [KV09; KN09], for the formulas), we can show the following.

Corollary 3.4 (TL-ADF Interpretation). *For every TL formula ϕ , there exists an ADF formula φ in pnf with $\text{SUP}(\varphi) = \text{FREE}(\varphi) = \text{FREE}(\phi)$ such that, for all structures \mathcal{M} and teams $X \in \text{Teams}_{\supseteq}(\text{FREE}(\phi))$, the following equivalence holds true: $\mathcal{M}, X \models_{\text{TL}} \phi$ iff $\mathcal{M}, \{X\} \models^{\exists\forall} \varphi$.*

3.3 Game-Theoretic Semantics

As discussed in Section 3.1.3, the alternating Hodges semantic relation $\mathcal{M} \models \phi$ implies the existence of a *semantic game* $\mathcal{G}_{\phi}^{\mathcal{M}}$, played by Eloise and Abelard, with the property

that Eloise wins the game iff the ADIF sentence ϕ is indeed satisfied in the structure \mathcal{M} . In that game, basically, the two players battle each other in challenge-response trials, where each of them tries to win the quantifier-free subformula or force the other one to break the (in)dependence constraints. In this section, we formalise such a game, thus providing a *game-theoretic semantics* for ADIF and a proof of its adequacy *w.r.t* both the compositional semantics of Definition 3.2 and the Herbrand-Skolem semantics of Theorem 3.6. Thanks to Corollary 3.4, this result also provides an indirect game-theoretic semantics for TL, a result that, as far as we know, was still missing [Vää07]. Note that, unlike for DIF [HS97; MSS11], \mathcal{G}_ϕ^M needs to be a *zero-sum perfect-information* game in order to comply with the game-theoretic determinacy of the logic (see Corollary 2.4), which for sentences is reflected in the law of excluded middle (see Corollary 2.5).

A two-player turn-based *arena* $A = \langle P^\exists, P^\forall, p_I, M \rangle$ is a tuple where

1. P^\exists, P^\forall are the sets of positions owned by *Eloise* and *Abelard* respectively with $P^\exists \cap P^\forall = \emptyset$; we denote by P the set of all positions in the game, i.e. $P = P^\exists \uplus P^\forall$,
2. $p_I \in P$ is the *initial position*, and
3. $M \subseteq P \times P$ is the binary left-total relation describing all possible *moves*.

A *path* $\pi \in \mathbf{Paths} \subseteq P^\infty$ is a finite or infinite sequence of positions compatible with the move relation, i.e., $(\pi[i], \pi[i+1]) \in M$, for all $i \in \llbracket 0, \dots, |\pi| - 1 \rrbracket$; it is *initial* if $|\pi| > 0$ and $\pi[0] = p_I$. A *history* for player $\alpha \in \{\exists, \forall\}$ is a finite initial path $h \in \mathbf{Hist}_\alpha \subseteq \mathbf{Paths} \cap (P^* \cdot P^\alpha)$ terminating in an α -position. A *play* $\rho \in \mathbf{Plays} \subseteq \mathbf{Paths}$ is a maximal (i.e., infinite since M is left-total) initial path. A *strategy* for player $\alpha \in \{\exists, \forall\}$ is a function $\sigma_\alpha \in \mathbf{Strat}_\alpha \subseteq \mathbf{Hist}_\alpha \rightarrow P$ mapping each α -history $h \in \mathbf{Hist}_\alpha$ to a position $\sigma_\alpha(h) \in P$ compatible with the move relation, i.e., $(\text{lst}(h), \sigma_\alpha(h)) \in M$. The *induced play* of a pair of strategies $(\sigma_\exists, \sigma_\forall) \in \mathbf{Strat}_\exists \times \mathbf{Strat}_\forall$ is the unique play $\pi \in \mathbf{Plays}$ such that $\pi[i+1] = \sigma_\exists(\pi[:i])$, if $\pi[i] \in P^\exists$, and $\pi[i+1] = \sigma_\forall(\pi[:i])$, otherwise, for all $i \in \llbracket 0, \dots, |\pi| - 1 \rrbracket$. A *game* $\mathcal{G} = \langle A, \mathbf{Win} \rangle$ is a tuple, where A is an arena and $\mathbf{Win} \subseteq \mathbf{Plays}$ is the set of *winning* plays for Eloise; the complement $\mathbf{Plays} \setminus \mathbf{Win}$ is *winning* for Abelard. Eloise (*resp.*, Abelard) *wins* the game if she (*resp.*, he) has a strategy $\sigma_\exists \in \mathbf{Strat}_\exists$ (*resp.*, $\sigma_\forall \in \mathbf{Strat}_\forall$) such that, for all opponent strategies $\sigma_\forall \in \mathbf{Strat}_\forall$ (*resp.*, $\sigma_\exists \in \mathbf{Strat}_\exists$), the corresponding induced play does (*resp.*, does not) belong to \mathbf{Win} . A game is *determined* if one of the two players wins.

With the notation put in place, we can now describe the semantic game, called *independence game*, where not only the players perform the choices corresponding to the operators in the formula, but also check that the choices of the opponent conform to the

associated independence constraints. Although a specific move for each ADIF syntactic construct can be given, for the sake of a simpler presentation, we only define the moves for the quantifiers. Any quantifier-free FO formula ψ , indeed, can be interpreted as a monolithic atomic relation, whose truth can be immediately evaluated once an assignment on all its free variables is given. For this reason, we assume $\phi = \wp\psi$ to be in pnf, for some quantifier prefix $\wp \in \mathbf{Q}$, where no variable is quantified twice. Finally, as a standard assumption from a descriptive-complexity viewpoint [Imm99; Grä+05], we restrict to finite structures only. The general case, as well as the lift of the approach to formulas, will be the focus of future work.

The game for $\phi = \wp\psi$ consists of two recurrent stages/phases, called *decision* and *challenge*. The decision phase is almost identical to a classic Hintikka’s FO game [HS97], where the player associated with the current subformula $\varphi = Q^{\pm W}x.\varphi'$ of ϕ chooses a value for the bound variable x to be stored in the current assignment χ . Once all quantifiers are eliminated, however, instead of declaring the winner by simply evaluating the truth of $\mathcal{M}, \chi \models_{\text{FO}} \psi$, the game enters the challenge phase. Here the players, following again the order of quantification, are asked to confirm or change their choices. Making a change here is intended to allow for verifying that the independence constraints declared in \wp are satisfied; after all, if the opponent’s choice is indeed independent of the player’s one, such a change should not make any difference in the values of opponent’s variables. In more detail, the player associated with $Q^{\pm W}x.\varphi'$ can either

1. confirm her/his own choice for the value of x , maintaining both the assignment χ and phase unchanged or
2. challenge the adversary, by modifying the value assigned to the variable x in χ , deleting all values for the variables quantified in \wp after x , and reverting to the decision phase.

In both cases, the control is passed on to the player of the formula φ' in the scope of the quantifier $Q^{\pm W}x$, so as to allow her/him to reply to the challenge. As it should be evident from the alternation of phases, unlike the semantic game for FO, $\mathcal{G}_\phi^{\mathcal{M}}$ is an *infinite-duration* game that allows for both finite and infinite plays. The finite ones necessarily terminate in a position of the challenge phase with current subformula ψ , where the winner can be determined by evaluating the truth of $\mathcal{M}, \chi \models_{\text{FO}} \psi$. The infinite plays, instead, are won by the player able to force the adversary to change infinitely often the values of one of her/his own variables x in a way that violates the independence constraints, without being able, at the same time, to force the challenger to do the same on a variable

subsequent to x in \wp . We clarify this point later on.

The formalisation of the arena $A_\phi^{\mathcal{M}}$ underlying the independence game $\mathcal{G}_\phi^{\mathcal{M}}$ is reported below, where $\text{PSF}(\phi)$ denotes the smallest set of subformulas of ϕ , called *prefix subformulas*, such that

1. $\phi \in \text{PSF}(\phi)$ and
2. if $\varphi = Q^{\pm W}x. \varphi' \in \text{PSF}(\phi)$ then $\varphi' \in \text{PSF}(\phi)$.

As an example, for the sentence $\phi_4 = \exists x. \forall^{+\emptyset}y. \neg(x = y)$ reported in Example 3.2, we have $\text{PSF}(\phi_4) = \{\phi_4, \forall^{+\emptyset}y. \neg(x = y), \neg(x = y)\}$.

Construction 3.1 (Independence Arena). *For a finite structure \mathcal{M} and a pnf ADIF sentence $\phi = \wp\psi$, with $\psi \in \text{FO}$, the independence arena $A_\phi^{\mathcal{M}} = \langle P^\exists, P^\forall, p_I, M \rangle$ is defined as follows.*

1. the set of positions $P \subseteq \text{PSF}(\phi) \times \text{Asg} \times \{I, II\}$ contains those triples $(\varphi, \chi, \blacklozenge)$ of a prefix subformula $\varphi \in \text{PSF}(\phi)$ of ϕ , an assignment $\chi \in \text{Asg}$, and a phase flag $\blacklozenge \in \{I, II\}$ such that $\chi \in \text{Asg}(\text{FREE}(\varphi))$, if $\blacklozenge = I$, and $\chi \in \text{Asg}(\text{FREE}(\psi))$, otherwise;
2. the set P^\exists of Eloise's (resp., P^\forall of Abelard's) positions contains the triples of the form $(\exists^{\pm W}x. \varphi', \chi, \blacklozenge)$ or (ψ, χ, I) (resp., $(\forall^{\pm W}x. \varphi', \chi, \blacklozenge)$);
3. the initial position $p_I \stackrel{\text{def}}{=} (\phi, \emptyset, I)$ contains the original sentence ϕ associated with the empty assignment \emptyset and the phase flag I ;
4. the move relation $M \subseteq P \times P$ contains exactly those pairs of positions $(p_1, p_2) \in P \times P$ satisfying one of the conditions below:
 - (a) $p_1 = (Q^{\pm W}x. \varphi', \chi, \blacklozenge)$ and $p_2 = (\varphi', \chi, \blacklozenge)$, with $x \notin \text{FREE}(\varphi')$;
 - (b) $p_1 = (Q^{\pm W}x. \varphi', \chi, I)$ and $p_2 = (\varphi', \chi[x \mapsto v], I)$, for some $v \in \mathbb{D}$;
 - (c) $p_1 = (\psi, \chi, I)$ and $p_2 = (\phi, \chi, II)$;
 - (d) $p_1 = (Q^{\pm W}x. \varphi', \chi, II)$ and $p_2 = (\varphi', \chi, II)$;
 - (e) $p_1 = (Q^{\pm W}x. \varphi', \chi, II)$ and $p_2 = (\varphi', \chi'[x \mapsto v], I)$, for some $v \in \mathbb{D}$ with $v \neq \chi(x)$, where $\chi' \stackrel{\text{def}}{=} \chi \upharpoonright_{\text{FREE}(Q^{\pm W}x. \varphi')}$.

Intuitively, a position $(\varphi, \chi, \blacklozenge)$ maintains the information about the formula φ that still has to be played against, the assignment χ containing the variables whose values have already been chosen, and a flag \blacklozenge identifying the phase, either I or II . Item 4a forces the trivial move for the vacuous quantifications, Item 4b defines the moves for

the decision phase, Item 4c switches from the decision to the challenge phase, Item 4d defines the confirmation of the choice already made, and, finally, Item 4e describes the challenge to the adversary, where the phase is reverted to the decision one, the value for the variable involved in the challenge is changed, and all values for the subsequent variables are deleted.

The winning condition for the game is defined as follows. Since the winner of finite plays is easy to determine, as it only depends on whether the assignment in the last position satisfies ψ , we shall focus on the infinite ones. Let us consider an arbitrary prefix subformula $\varphi = Q^{\pm W}x.\varphi' \in \text{PSF}(\phi)$ with $x \in V \stackrel{\text{def}}{=} \text{FREE}(\varphi')$. By $\mathbf{F}_\varphi: \text{Asg}(V) \rightarrow 2^{\text{Rsp}_{\pm W}}$ we denote the map associating each assignment $\chi \in \text{Asg}(V)$ defined over the variables in V with the set $\mathbf{F}_\varphi(\chi) \stackrel{\text{def}}{=} \{F \in \text{Rsp}_{\pm W} \mid F(\chi) = \chi(x)\}$ of all the $\pm W$ -functions compatible with the value assigned to x in χ . In addition, by $\mathbf{R}_\varphi: \text{Hist} \rightarrow 2^{\text{Rsp}_{\pm W}}$, with $\text{Hist} \stackrel{\text{def}}{=} \text{Hist}_{\exists} \cup \text{Hist}_{\forall}$, we denote the map assigning to each history $h \in \text{Hist}$ the set $\mathbf{R}_\varphi(h)$ of all the $\pm W$ -functions compatible with the most recent assignments along h . \mathbf{R}_φ is referred to as a *bucket* of response functions. Formally:

- $\mathbf{R}_\varphi(p_I) \stackrel{\text{def}}{=} \text{Rsp}_{\pm W}$;
- $\mathbf{R}_\varphi(h \cdot (\varphi', \chi, I)) \stackrel{\text{def}}{=} \begin{cases} \mathbf{F}_\varphi(\chi), & \text{if } \mathbf{R}_\varphi(h) \cap \mathbf{F}_\varphi(\chi) = \emptyset; \\ \mathbf{R}_\varphi(h) \cap \mathbf{F}_\varphi(\chi), & \text{otherwise;} \end{cases}$
- $\mathbf{R}_\varphi(h \cdot p) \stackrel{\text{def}}{=} \mathbf{R}_\varphi(h)$, in all other cases, i.e., $p \neq (\varphi', _, I)$.

Essentially, the bucket $\mathbf{R}_\varphi(h)$ maintains the most updated set of Herbrand/Skolem functions for the variable x that the associated player can use to reply to all the variables which x depends upon. When a play starts, no choice has been made yet, so $\mathbf{R}_\varphi(p_I)$ is full. Once a position (φ', χ, I) is reached after a history h , a fresh value $\chi(x)$ for x has just been chosen to resolve the quantifier Q , so the bucket is updated by removing from $\mathbf{R}_\varphi(h)$ all the functions that are not compatible with this new value. If such resulting set becomes empty, the player is caught cheating and the bucket is replenished taking into account only the choice just made.

In general there are two reasons for a player to cheat. Either she/he is changing the value of the variable to challenge the adversary to prove that he/she is complying with the independence constraints (Item 4e), or she/he chooses a new value because is unable to both satisfy her/his goal and comply with the constraints on her/his variables (Item 4b). Obviously, the second type of cheating, called *defensive cheat*, can, in turn, induce one of the first type, called *challenge cheat*. Hence, complex chains of different types of cheating can occur. In order to identify which player is the last one who was forced to cheat,

we consider an arbitrary map $\text{PRT}: \text{PSF}(\phi) \rightarrow \mathbb{N}$ assigning to each prefix subformula $\varphi = Q^{\pm W}x. \varphi' \in \text{PSF}(\phi)$ a priority $\text{PRT}(\varphi)$ such that

1. $\text{PRT}(\varphi)$ is even iff $Q = \forall$ and
2. $\text{PRT}(\varphi) < \text{PRT}(\varphi')$.

To each history $h \in \text{Hist}$ we can then assign the sequence of cheats $\text{CHT}(h)$ occurring in it via the map $\text{CHT}: \text{Hist} \rightarrow \mathbb{N}^*$ as follows:

- $\text{CHT}(p_I) \stackrel{\text{def}}{=} 0$;
- $\text{CHT}(h \cdot (\varphi', \chi, I)) \stackrel{\text{def}}{=} \text{CHT}(h) \cdot \text{PRT}(\varphi)$, whenever $\mathbf{R}_\varphi(h) \cap \mathbf{F}_\varphi(\chi) = \emptyset$;
- $\text{CHT}(h \cdot p) \stackrel{\text{def}}{=} \text{CHT}(h) \cdot 0$, in all other cases.

This construction easily lifts to infinite plays $\pi \in \text{Plays}^\omega \stackrel{\text{def}}{=} \text{Plays} \cap P^\omega$ through the map $\text{CHT}: \text{Plays}^\omega \rightarrow \mathbb{N}^\omega$ such that $\text{CHT}(\pi)[i] = \text{CHT}(\pi[:i])$, for all $i \in \mathbb{N}$. Finally, $\text{PRT}(\pi)$ denotes the maximal priority seen infinitely often along $\text{CHT}(\pi)$. Note that every infinite play necessarily contains at least infinitely many challenge cheats (Item 4e). Thus, $\text{PRT}(\pi)$ uniquely identifies the right-most variable in \wp over which the corresponding player cheated, without being able, at the same time, to force the adversary to do the same. If $\text{PRT}(\pi)$ is even, Abelard is cheating infinitely often, so he loses the play π , which is, therefore, won by Eloise.

Construction 3.2 (Independence Game). *For a finite structure \mathcal{M} and a pnf ADIF sentence $\phi = \wp\psi$, with $\psi \in \text{FO}$, the independence game $\mathcal{G}_\phi^\mathcal{M} = \langle A, \text{Win} \rangle$ is defined as prescribed in the following:*

1. A is the independence arena $A_\phi^\mathcal{M}$ defined in Construction 3.1;
2. $\text{Win} \subseteq \text{Plays}$ is the set of all the plays π satisfying the following conditions:
 - (a) if π is infinite then $\text{PRT}(\pi)$ is even;
 - (b) if π is finite then $\text{lst}(\pi) = (\psi, \chi, II)$ and $\mathcal{M}, \chi \models_{\text{FO}} \psi$, for some assignment $\chi \in \text{Asg}(\text{FREE}(\psi))$.

Example 3.7. *Let us consider $\phi_7 = \exists x. \forall^{+\emptyset} y. \exists^{+x} z. (\psi_1(x, y) \wedge \psi_2(y, z))$, the sentence of Example 3.4 from Section 3.1.1, which is true in the binary structure \mathcal{M} of that example. Therefore, Eloise, who controls the values of the variables x and z , must have a strategy to win the independence game $\mathcal{G}_{\phi_7}^\mathcal{M}$. One possibility is to choose, during the decision phase, the constant function $f_x = 0$ for x and the identity function $f_z(x) = x$ for z . Clearly, she wins any finite play where Abelard chooses the constant function $f_y = 0$ for y , since the resulting assignment satisfies both $(x = y)$ and $(y = z)$. Let us assume, then, that he*

chooses $f_y = 1$, instead, in the decision phase. Since at the end of this phase Eloise knows she is losing, she will challenge Abelard by changing her function f_x for x to the constant 1. This raises the priority of the current play fragment to 1. Now, if Abelard sticks to function $f_y = 1$ for y , he loses, since $f_z(x) = x$ would now give z value 1 as well, leading to a finite play. So he needs to modify his choice to $f_y = 0$, this time raising the priority of the play fragment to 2 and generating a challenge for Eloise on z . Eloise, however, can stick to the identity function and make way to a new challenge phase. Now, since Eloise is losing with the current assignment, she will challenge once again, choosing $f_x = 0$ and raising priority 1. Abelard is then forced to change function and raise priority 2 and we are back to where we started. This cyclic process ends up forming an infinite play whose maximal priority is 2, since Eloise can force Abelard to defensively change bucket infinitely often, thus satisfying her winning condition.

It is worth noting that the game devised above bears some similarities with the *team-building game* proposed by [Bra13] for DL [Vää07]. Both ours and his are complete-information games extending Hintikka’s game for FO. In addition, Bradfield’s game also checks the uniformity of the choices made by Eloise by means of a challenge mechanism, where the sentence is played over repeatedly by the players. The similarities, however, end here as the two games differ significantly in nature. First, the repeated evaluations of a sentence ϕ in Bradfield’s game allow him to build teams during a play, one for each dependence atom occurring in ϕ . Each team is then used to check whether Eloise’s choices have been made in accordance to the dependency constraint encoded by the corresponding atom. All these teams are explicitly recorded in each state of his game, together with the partial assignment recording the choices made by the players so far in the current repetition. In this sense, then, Bradfield’s arena is intrinsically second order, as it records sets of assignments in each state and contains moves that update such sets. Second, Bradfield’s game on finite structures only admits finite plays and its winning condition, then, boils down to a simple reachability. On the contrary, our game is played in a purely first-order arena, whose states only keep track of players choices collected in the partial assignment. Moreover, it always admits infinite plays, where players can repeatedly challenge each other forever. The second-order power of our game, then, resides entirely in the winning condition, where the priority-based mechanism accounts for the alternation of the quantifiers along the, possibly infinite, repeated evaluations of the sentence.

To conclude, by exploiting Theorem 3.6, it is possible to prove the adequacy of the game-theoretic semantics *w.r.t* the model-theoretic one of Definition 3.4 and, in turn, *w.r.t*

the compositional one of Definition 3.2, where the Herbrand/Skolem functions obtained by the evaluation of the existential (*resp.*, universal) meta quantifiers of the **Meta** sentence $\text{HSP}_\varphi \wp \psi$ (*resp.*, $\neg \text{HSP}_\varphi \wp \psi$) induce a winning strategy for Eloise (*resp.*, Abelard) in $\mathcal{G}_\phi^{\mathcal{M}}$. This also implies the determinacy of the independence game, without the need to rely on topological determinacy theorems, as those of [Mar75; Mar85].

Theorem 3.12 (Game-Theoretic Semantics). *For a finite structure \mathcal{M} and an ADIF sentence ϕ in prenex form, the independence game $\mathcal{G}_\phi^{\mathcal{M}}$ defined in construction 3.2 is such that $\mathcal{M} \models \phi$ (*resp.*, $\mathcal{M} \not\models \phi$) iff $\mathcal{G}_\phi^{\mathcal{M}}$ is won by Eloise (*resp.*, Abelard).*

3.4 Conclusion

We have introduced Alternating Dependence/Independence-Friendly Logic (ADIF), a conservative extension of Independence-Friendly Logic (IFL), that incorporates negation in a very natural way and avoids the indeterminacy of the logic. This is achieved by means of a hyperteam semantics, which allows us to treat the two players symmetrically and force both of them to make their choices according to the (in)dependence constraints specified in the corresponding quantifiers. Thanks to the fully symmetric treatment of the (in)dependence constraints, the new semantics allows for restoring the law of excluded middle for sentences and enjoys the property of game-theoretic determinacy. Interestingly enough, this also grants ADIF the full expressive power of Second Order Logic (SO) and, as a consequence, also of Team Logic (TL), without the need of including additional connectives in the language. The expressive power gained with respect to IFL can be leveraged, for instance, to define directly in the logic notions such as indeterminacy and sensitivity to signalling, whose expressions require the restrictions of both players, that is the uniformity of their strategies, at the same time. This gives ADIF the flavour of a logic suitable to reason “about” imperfect information in a general sense. For the prenex fragment, a Herbrand-Skolem semantics is also provided that directly connects ADIF with SO, as well as a game-theoretic semantics on finite structures, given in terms of a determined turn-based infinite-duration perfect-information game played on a first-order arena.

Interesting questions that remain open concern whether a prenex normal form theorem holds for the language. Equally unsettled is the actual expressive power of ADIF. We do show that it is at least as expressive as SO and, thus, covers the full polynomial hierarchy. The proof for the other direction, however, relies upon the assumption of equipotency be-

tween the hyperteam \mathfrak{X} and the domain \mathbb{D} of the underlying structure \mathcal{M} , which allows us to encode hyperteam by means of a suitable relation $\text{REL}(\mathfrak{X})$. There seems to be no straightforward way to do the same for “big” hyperteam. Yet again, it is not clear whether such “big” hyperteam actually matter, in the sense of there being a formula that can distinguish between “big” and “small” hyperteam. Usually, similar questions have been addressed by defining suitable Ehrenfeucht-Fraïssé games to precisely characterise the expressive power of the logic. For this reason, one may think to do the same for ADIF as well. The main difficulty we foresee here is, however, the treatment of quantifications, for which no explicit commitment to a specific valued is made in the semantics (all choices are evenly encoded in the hyperteam). In a classic Ehrenfeucht-Fraïssé games game, instead, the moves corresponding to the choices of a value by a quantifier make explicit commitments. Currently, it is not clear to us how to circumvent this discrepancy.

3.A Proofs of Section 3.1

Most results can be proven thanks to a straight forward adaptation of what is presented in the Appendices of Chapter 2. However adapting the monotony results for the \sqsubseteq_W relation with respect to the $\text{EXT}_{\pm U}$ operator is a bit more intricate and is developed here.

Recall that, for a team $X \in \mathfrak{X}_{\upharpoonright W}$, we denote by $X \upharpoonright^W$ one (arbitrarily chosen) of the teams $Y \in \mathfrak{X}$ such that $Y_{\upharpoonright W} = X$.

Lemma 3.4 (ADIF Monotonicity I). *Let $\mathfrak{X}, \mathfrak{X}' \in \text{HypTeams}$ be two hypertteams with $\mathfrak{X} \sqsubseteq_W \mathfrak{X}'$, for some $W \subseteq \text{Var}$. Then, the following hold true:*

1. $\overline{\mathfrak{X}'} \sqsubseteq_W \overline{\mathfrak{X}}$;
2. (a) $\mathfrak{X} =_{=W} \text{EXT}_{+U}(\mathfrak{X}, x)$, if $x \notin W$, with $U \subseteq \text{Var}$;
 (b) $\text{EXT}_{+U}(\mathfrak{X}, x) \sqsubseteq_{W \cup \{x\}} \text{EXT}_{+U'}(\mathfrak{X}', x)$, with $x \in \text{Var}$, $U \subseteq U' \subseteq \text{Var}$, and $U \subseteq W$;
3. for every $(\mathfrak{X}'_1, \mathfrak{X}'_2) \in \text{PART}(\mathfrak{X}')$, there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ such that $\mathfrak{X}_1 \sqsubseteq_W \mathfrak{X}'_1$ and $\mathfrak{X}_2 \sqsubseteq_W \mathfrak{X}'_2$.

Proof. 1) By $\mathfrak{X} \sqsubseteq_W \mathfrak{X}'$, there is a function $f : \mathfrak{X}_{\upharpoonright W} \rightarrow \mathfrak{X}'_{\upharpoonright W}$ such that $f(X_{\upharpoonright W}) \subseteq X'_{\upharpoonright W}$ for all $X \in \mathfrak{X}$. Moreover, for all $X \in \mathfrak{X}$, since $f(X_{\upharpoonright W}) \subseteq X'_{\upharpoonright W}$, there is a function $g_X : \cup\{X' \in \mathfrak{X}' \mid X'_{\upharpoonright W} = f(X_{\upharpoonright W})\} \rightarrow X$ such that $\chi_{\upharpoonright W} = (g_X(\chi))_{\upharpoonright W}$ for all χ in $\cup\{X' \in \mathfrak{X}' \mid X'_{\upharpoonright W} = f(X_{\upharpoonright W})\}$. In order to prove the claim, consider a generic team $X' \in \overline{\mathfrak{X}'}_{\upharpoonright W}$. We have to show that there is $X \in \overline{\mathfrak{X}}$ such that $X_{\upharpoonright W} \subseteq X'$. By the definition of $\overline{\mathfrak{X}'}_{\upharpoonright W}$, we have that $X' = (\text{img}(\Gamma'))_{\upharpoonright W}$, for some $\Gamma' \in \text{Chc}(\mathfrak{X}')$. We define $\Gamma \in \text{Chc}(\mathfrak{X})$ as: $\Gamma(X) = g_X(\Gamma'((f(X_{\upharpoonright W})) \upharpoonright^W))$ for all $X \in \mathfrak{X}$. Clearly, $(\text{img}(\Gamma))_{\upharpoonright W} \subseteq (\text{img}(\Gamma'))_{\upharpoonright W} = X'$. Since $(\text{img}(\Gamma)) \in \overline{\mathfrak{X}}$, the thesis holds.

2a) The claim follows from the fact that for every $F \in \text{Rsp}$, $\chi \in \text{Asg}$, and $x \notin W$, it holds that $\chi[x \mapsto F(\chi)]_{\upharpoonright W} = \chi_{\upharpoonright W}$, which implies $\text{EXT}(X, F, x)_{\upharpoonright W} = X_{\upharpoonright W}$ for every $X \in \mathfrak{X}$ and $F \in \text{Rsp}_{+U}$, and the claim follows.

2b) By $\mathfrak{X} \sqsubseteq_W \mathfrak{X}'$, there is a function $f : \mathfrak{X}_{\upharpoonright W} \rightarrow \mathfrak{X}'_{\upharpoonright W}$ such that $f(X_{\upharpoonright W}) \subseteq X'_{\upharpoonright W}$ for all $X \in \mathfrak{X}$. In order to prove the claim, take a generic team $\hat{X} \in \text{EXT}_{+U}(\mathfrak{X}, x)$. Thus, $\hat{X} = \text{EXT}(X, F, x) = \{\chi[x \mapsto F(\chi)] \mid \chi \in X\}$, for some $X \in \mathfrak{X}$ and $F \in \text{Rsp}_{+U}$. Let $X' = (f(X_{\upharpoonright W})) \upharpoonright^W \in \mathfrak{X}'$. Clearly, $X'_{\upharpoonright W} = f(X_{\upharpoonright W}) \subseteq X_{\upharpoonright W}$. Moreover, $\text{EXT}(X', F, x) \in \text{EXT}_{+U'}(\mathfrak{X}', x)$, since $F \in \text{Rsp}_{+U} \subseteq \text{Rsp}_{+U'}$ (as $U \subseteq U'$). To complete the proof, it is enough to show that $\text{EXT}(X', F, x)_{\upharpoonright W \cup \{x\}} \subseteq \hat{X}_{\upharpoonright W \cup \{x\}}$. To this purpose, take $\chi'[x \mapsto F(\chi')]_{\upharpoonright W \cup \{x\}}$ for some $\chi' \in X'$. Observe that

- $\chi'_{|W} \in X'_{|W} = f(X_{|W}) \subseteq X_{|W}$, which means that there is $\chi \in X$ such that $\chi_{|W} = \chi'_{|W}$. Since $U \subseteq W$, it holds that $\chi_{|U} = \chi'_{|U}$, which implies $F(\chi) = F(\chi')$, as $F \in \text{Rsp}_{+U}$. Therefore, $\chi'[x \mapsto F(\chi')]_{|W \cup \{x\}} = \chi[x \mapsto F(\chi)]_{|W \cup \{x\}} \in \hat{X}_{|W \cup \{x\}}$.
- 3) By $\mathfrak{X} \sqsubseteq_W \mathfrak{X}'$, there is a function $f : \mathfrak{X}_{|W} \rightarrow \mathfrak{X}'_{|W}$ such that $f(X_{|W}) \subseteq X_{|W}$ for all $X \in \mathfrak{X}$. Let $(\mathfrak{X}'_1, \mathfrak{X}'_2) \in \text{PART}(\mathfrak{X}')$ and define $\mathfrak{X}_i = \{X \in \mathfrak{X} \mid (f(X_{|W})) \uparrow^W \in \mathfrak{X}'_i\}$ for $i \in \{1, 2\}$. We have to show that $\mathfrak{X}_i \sqsubseteq_W \mathfrak{X}'_i$ ($i \in \{1, 2\}$). To this end, let $X \in \mathfrak{X}_i$ and consider team $(f(X_{|W})) \uparrow^W \in \mathfrak{X}'_i$. Clearly, $((f(X_{|W})) \uparrow^W)_{|W} = f(X_{|W}) \subseteq X_{|W}$. The thesis follows as $((f(X_{|W})) \uparrow^W)_{|W} \in \mathfrak{X}'_i$. \square

Lemma 3.5 (Uniform Extension Monotonicity). *For all sets of variables $W \subseteq \text{Var}$, function assignments $\mathcal{F} \in \text{FunAsg}$, and hyperteams $\mathfrak{X}_1, \mathfrak{X}_2 \in \text{HypTeams}$, where $\mathfrak{X}_1 \sqsubseteq_W \mathfrak{X}_2$ and $\mathcal{F}(x) \in \text{Rsp}_W$, for all $x \in \text{DOM}(\mathcal{F}) \cap W$, it holds that $\text{EXT}(\mathfrak{X}_1, \mathcal{F}) \sqsubseteq_W \text{EXT}(\mathfrak{X}_2, \mathcal{F})$.*

Proof. The proof is almost the same as for Lemma 2.9. \square

Thanks to previous lemmas, Theorem 2.8 and thus, Theorem 2.9 can be immediately adapted to **Meta-ADIF**, which are the necessary results to prove Theorem 3.3.

Lemma 2.7 is generalised for ADIF as follows.

Lemma 3.6 (Extension Interpretation). *The following four equivalences hold true, for all hyperteams $\mathfrak{X} \in \text{HypTeams}(V)$ over $V \subseteq \text{Var}$, properties $\Psi \subseteq \text{Asg}(V \cup \{x\})$ over $V \cup \{x\}$ with $x \in \text{Var} \setminus V$, sets of variables $W \subseteq \text{Var}$, and quantifier symbols $Q \in \{\exists, \forall\}$.*

1. *Statements 1a and 1b are equivalent, whenever Q is α -coherent:*
 - (a) *there exists $X' \in \text{EXT}_\alpha(\mathfrak{X}, Q^{\pm W}x)$ such that $X' \subseteq \Psi$;*
 - (b) *there exist $F \in \text{Rsp}_{\pm W}$ and $X \in \mathfrak{X}$ such that $\text{EXT}(X, F, x) \subseteq \Psi$.*
2. *Statements 2a and 2b are equivalent, whenever Q is α -coherent:*
 - (a) *for all $X' \in \text{EXT}_\alpha(\mathfrak{X}, Q^{\pm W}x)$, it holds that $X' \cap \Psi \neq \emptyset$;*
 - (b) *for all $F \in \text{Rsp}_{\pm W}$ and $X \in \mathfrak{X}$, it holds that $\text{EXT}(X, F, x) \cap \Psi \neq \emptyset$.*
3. *Statements 3a and 3b are equivalent, whenever Q is $\bar{\alpha}$ -coherent:*
 - (a) *there exists $X' \in \text{EXT}_\alpha(\mathfrak{X}, Q^{\pm W}x)$ such that $X' \subseteq \Psi$;*
 - (b) *for all $F \in \text{Rsp}_{\pm W}$, it holds that $\text{EXT}(X, F, x) \subseteq \Psi$, for some $X \in \mathfrak{X}$.*
4. *Statements 4a and 4b are equivalent, whenever Q is $\bar{\alpha}$ -coherent:*
 - (a) *for all $X' \in \text{EXT}_\alpha(\mathfrak{X}, Q^{\pm W}x)$, it holds that $X' \cap \Psi \neq \emptyset$;*
 - (b) *there is $F \in \text{Rsp}_{\pm W}$ such that $\text{EXT}(X, F, x) \cap \Psi \neq \emptyset$, for all $X \in \mathfrak{X}$.*

The proof of this lemma is the same as the one for Lemma 2.7 modulo the addition of the dependencies. This lemma, with the addition of Theorem 3.8, allows to prove Theorem 3.5 as we proved Theorem 2.6. In turn, Theorem 3.6 follows.

3.B Proofs of Section 3.2

Lemma 3.2 (Cylindrical Extension). *Let $\mathfrak{X} \in \text{HypTeams}$ be a hyperteam. Then, $\text{CYL}(\mathfrak{X}, x) \equiv \overline{\text{EXT}_{\pm W}(\mathfrak{X}, x)}$, for all variables $x \in \text{Var}$ and sets of variables W , with $\text{VAR}(\mathfrak{X}) \subseteq W \subseteq \text{Var}$.*

Proof. The proof is done by showing the two directions of the equivalence.

First, we prove the following:

$$\text{CYL}(\mathfrak{X}, x) \sqsubseteq \overline{\text{EXT}_{\pm W}(\mathfrak{X}, x)}.$$

Let $X_u \in \text{CYL}(\mathfrak{X}, x)$. There is $X \in \mathfrak{X}$ such that $X_u = \text{CYL}(X, x)$. Remark that for every $X' \in \mathfrak{X}$ there is $\chi_{X'} \in X' \cap X$. Then, for every $F \in \text{Rsp}_W$, it holds that $\chi_{X'}[x \mapsto F(\chi_{X'})] \in X_u$. Now, observe that for every $\hat{X} \in \overline{\text{EXT}_{\pm W}(\mathfrak{X}, x)}$, there is $X' \in \mathfrak{X}$ and $F \in \text{Rsp}_W$ such that $\hat{X} = \text{EXT}(X', F, x)$. Consider $\Gamma \in \text{Chc}(\overline{\text{EXT}_{\pm W}(\mathfrak{X}, x)})$ defined as follows. For every $\hat{X} \in \overline{\text{EXT}_{\pm W}(\mathfrak{X}, x)}$, we define $\Gamma(\hat{X}) = \chi_{X'}[x \mapsto F(\chi_{X'})]$. We can deduce immediately that $\text{img}(\Gamma) \subseteq X_u$.

We turn now to showing that

$$\overline{\text{EXT}_{\pm W}(\mathfrak{X}, x)} \sqsubseteq \text{CYL}(\mathfrak{X}, x).$$

Let $\mathring{X} \in \overline{\text{EXT}_{\pm W}(\mathfrak{X}, x)}$. We have $\mathring{X} = \text{img}(\mathring{\Gamma})$ for some choice function $\mathring{\Gamma} \in \text{Chc}(\overline{\text{EXT}_{\pm W}(\mathfrak{X}, x)})$. Then,

$$\forall F \in \text{Rsp}_W, \forall X' \in \mathfrak{X}, \exists \chi' \in X' \text{ s.t. } \chi'[x \mapsto F(\chi')] \in \mathring{X}. \quad (3.1)$$

Toward contradiction, assume that $\text{CYL}(X, x) \not\subseteq \mathring{X}$ for all $X \in \mathfrak{X}$. Then for all $X \in \mathfrak{X}$, there is $\chi_X \in X$ and $v_X \in \mathbb{D}$ such that $\chi_X[x \mapsto v_X] \notin \mathring{X}$. We assume that $v_{X_1} = v_{X_2}$ if $\chi_{X_1} = \chi_{X_2}$ so that each χ is associated with only one $v \in \mathbb{D}$. Consider $\Gamma \in \text{Chc}(\mathfrak{X})$ such that $\Gamma(X) = \chi_X$ for all $X \in \mathfrak{X}$, and $F \in \text{Rsp}_W$ such that $F(\chi_X) = v_X$ for all $X \in \mathfrak{X}$. By construction, for all $\chi' \in \text{img}(\Gamma)$, it holds that $\chi'[x \mapsto F(\chi')] \notin \mathring{X}$ and, since $\text{img}(\Gamma) \in \mathfrak{X}$, we have a contradiction with (3.1). \square

Lemma 3.3 (Team Partitioning). *Let $\mathfrak{X} \in \text{HypTeams}$ be a hyperteam. Then:*

1. *for all hyperteam bipartitions $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ and teams $X'_1 \in \overline{\mathfrak{X}}_1$ and $X'_2 \in \overline{\mathfrak{X}}_2$, there exists a team $X \in \mathfrak{X}$ such that $X \subseteq X'_1 \cup X'_2$;*
2. *for all teams $X \in \mathfrak{X}$ and team bipartitions $(X_1, X_2) \in \text{PART}(X)$, there exist a hyperteam bipartition $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ and two teams $X'_1 \in \overline{\mathfrak{X}}_1$ and $X'_2 \in \overline{\mathfrak{X}}_2$ such that $X'_1 \subseteq X_1$ and $X'_2 \subseteq X_2$.*

Proof. In the following, we assume index i to range over $\{1, 2\}$.

1. Let $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ and $Y_i \in \overline{\mathfrak{X}}_i$. Then, there are $\Gamma_i \in \text{Chc}(\mathfrak{X}_i)$ such that $Y_i = \text{img}(\Gamma_i)$. Let $\Gamma \in \text{Chc}(\overline{\mathfrak{X}})$ be defined as: $\Gamma(X) = \Gamma_i(X)$ if $X \in \mathfrak{X}_i$, for all $X \in \overline{\mathfrak{X}}$. It clearly holds that $\text{img}(\Gamma) = \text{img}(\Gamma_1) \cup \text{img}(\Gamma_2)$ and $\text{img}(\Gamma) \in \overline{\mathfrak{X}}$. Finally, thanks to Lemma 2.1, there is $X^* \in \mathfrak{X}$ such that $X^* \subseteq \text{img}(\Gamma) = Y_1 \cup Y_2$.
2. Let $X \in \mathfrak{X}$ and $(X_1, X_2) \in \text{PART}X$. Consider \mathfrak{X}_1 and \mathfrak{X}_2 defined as follows: $\mathfrak{X}_1 = \{\text{img}(\Gamma) \mid \Gamma \in \text{Chc}(\mathfrak{X}) \text{ and } \Gamma(X) \in X_1\}$ and $\mathfrak{X}_2 = \overline{\mathfrak{X}} \setminus \mathfrak{X}_1$. Clearly, it holds that $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$. Moreover, for every $X'_i \in \mathfrak{X}_i$, it holds that $X'_i \cap X_i \neq \emptyset$. Let $\Gamma_i \in \text{Chc}(\mathfrak{X}_i)$ be such that $\Gamma_i(X'_i) \in Y_i \cap X_i$, for every $X'_i \in \mathfrak{X}_i$. Then, $\text{img}(\Gamma_i) \in \overline{\mathfrak{X}}_i$ is such that $\text{img}(\Gamma_i) \subseteq X_i$. □

The proof of the DIF adequacy property for ADIF uses the following monotonicity property known for IFL (and thus DIF).

Remark 3.2. *For all DIF formulas φ and teams $X, X' \subseteq \text{Asg}(V)$, with $\text{SUP}(\varphi) \subseteq V$ and $X \subseteq X'$, it holds that:*

1. *If $\mathcal{M}, X' \models_{\text{DIF}}^{\forall} \varphi$, then $\mathcal{M}, X \models_{\text{DIF}}^{\forall} \varphi$.*
2. *If $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \varphi$, then $\mathcal{M}, X' \models_{\text{DIF}}^{\exists} \varphi$;*

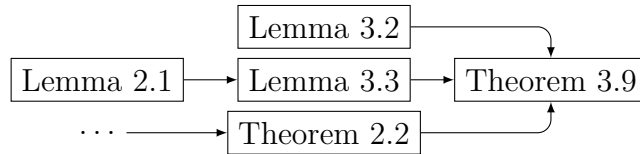


Figure 3.1 – Dependency graph of Theorem 3.9.

Theorem 3.9 (DIF Adequacy). *For all DIF formulas φ and hyperteam $\mathfrak{X} \in \text{HypTeams}_{\geq}(\text{SUP}(\varphi))$, it holds that:*

1. if φ is DIF_{\exists} then $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi$ iff there is a team $X \in \mathfrak{X}$ such that $\mathcal{M}, X \models_{DIF}^{\forall} \varphi$;
2. if φ is DIF_{\forall} then $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi$ iff, for all teams $X \in \mathfrak{X}$, it holds that $\mathcal{M}, X \models_{DIF}^{\exists} \varphi$.

Proof. In the following, we assume index i to range over $\{1, 2\}$.

To begin with, we prove Item 1. The proof is done by structural induction on the formula φ .

(base case) If $\varphi = P(\vec{x})$ or $\varphi = \neg P(\vec{x})$, then the property holds by the semantics rules.

(inductive cases) Suppose that the property holds for the subformulas of φ .

($\varphi = \varphi_1 \wedge \varphi_2$) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi_1 \wedge \varphi_2 \stackrel{sem.}{\Leftrightarrow}$ for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ it holds that $\mathcal{M}, \mathfrak{X}_1 \models^{\exists\forall} \varphi_1$ or $\mathcal{M}, \mathfrak{X}_2 \models^{\exists\forall} \varphi_2 \stackrel{ind.hp.}{\Leftrightarrow}$ for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\mathfrak{X})$ it holds that there is $X_1 \in \mathfrak{X}_1$ for which it holds $\mathcal{M}, X_1 \models_{DIF}^{\forall} \varphi_1$ or there is $X_2 \in \mathfrak{X}_2$ for which it holds $\mathcal{M}, X_2 \models_{DIF}^{\forall} \varphi_2 \Leftrightarrow$ there is $X \in \mathfrak{X}$ such that $\mathcal{M}, X \models_{DIF}^{\forall} \varphi_1$ and $\mathcal{M}, X \models_{DIF}^{\forall} \varphi_2 \stackrel{DIF-sem.}{\Leftrightarrow}$ there is $X \in \mathfrak{X}$ such that $\mathcal{M}, X \models_{DIF}^{\forall} \varphi_1 \wedge \varphi_2$.

($\varphi = \varphi_1 \vee \varphi_2$) If $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi_1 \vee \varphi_2$, then $\mathcal{M}, \overline{\mathfrak{X}} \models^{\forall\exists} \varphi_1 \vee \varphi_2$. By semantics, there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ such that $\mathcal{M}, \mathfrak{X}_1 \models^{\forall\exists} \varphi_1$ and $\mathcal{M}, \mathfrak{X}_2 \models^{\forall\exists} \varphi_2$, which amounts to say that there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ such that $\mathcal{M}, \overline{\mathfrak{X}}_1 \models^{\exists\forall} \varphi_1$ and $\mathcal{M}, \overline{\mathfrak{X}}_2 \models^{\exists\forall} \varphi_2$. By inductive hypothesis, there are $X_1 \in \overline{\mathfrak{X}}_1$ and $X_2 \in \overline{\mathfrak{X}}_2$ such that $\mathcal{M}, X_1 \models_{DIF}^{\forall} \varphi_1$ and $\mathcal{M}, X_2 \models_{DIF}^{\forall} \varphi_2$. By Item 1 of Lemma 3.3, there is $X \in \mathfrak{X}$ such that $X \subseteq X_1 \cup X_2$. By Item 1 of Remark 3.2, we have that $X'_1 \stackrel{def}{=} X_1 \cap X$ and $X'_2 \stackrel{def}{=} X \setminus X'_1$ are such that $\mathcal{M}, X'_1 \models_{DIF}^{\forall} \varphi_1$ and $\mathcal{M}, X'_2 \models_{DIF}^{\forall} \varphi_2$. Since, in addition, $(X'_1, X'_2) \in \text{PART}(X)$ holds, we conclude $\mathcal{M}, X \models_{DIF}^{\forall} \varphi_1 \vee \varphi_2$.

Conversely, if there is $X \in \mathfrak{X}$ such that $\mathcal{M}, X \models_{DIF}^{\forall} \varphi_1 \vee \varphi_2$, then there is $(X_1, X_2) \in \text{PART}(X)$ such that $\mathcal{M}, X_i \models_{DIF}^{\forall} \varphi_i$. By Item 2 of Lemma 3.3, there are $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ and $Y_i \in \overline{\mathfrak{X}}_i$ such that $Y_i \subseteq X_i$. Then, by Item 1 of Remark 3.2, it holds that $\mathcal{M}, Y_i \models_{DIF}^{\forall} \varphi_i$. By inductive hypothesis, we have $\mathcal{M}, \overline{\mathfrak{X}}_i \models^{\exists\forall} \varphi_i$, or, equivalently, $\mathcal{M}, \mathfrak{X}_i \models^{\forall\exists} \varphi_i$. Therefore, there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ such that $\mathcal{M}, \mathfrak{X}_i \models^{\forall\exists} \varphi_i$, which implies $\mathcal{M}, \overline{\mathfrak{X}} \models^{\forall\exists} \varphi_1 \vee \varphi_2$, and we can conclude $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \varphi_1 \vee \varphi_2$.

($\varphi = \exists^{\pm W} x \varphi$) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \exists^{\pm W} x \varphi \stackrel{sem.}{\Leftrightarrow} \mathcal{M}, \text{EXT}_{\pm W}(\mathfrak{X}, x) \models^{\exists\forall} \varphi \stackrel{ind.hp.}{\Leftrightarrow}$ there is $X \in \text{EXT}_{\pm W}(\mathfrak{X}, x)$ such that $\mathcal{M}, X \models_{DIF}^{\forall} \varphi \stackrel{def.}{\Leftrightarrow}$ there are $X \in \mathfrak{X}$ and $F \in \text{Rsp}_{\pm W}$ such that $\mathcal{M}, \text{EXT}(X, F, x) \models_{DIF}^{\forall} \varphi \stackrel{DIF-sem.}{\Leftrightarrow}$ there is $X \in \mathfrak{X}$ such that $\mathcal{M}, X \models_{DIF}^{\forall} \exists^{\pm W} x \varphi$.

($\varphi = \forall^{-\emptyset} x \varphi$) $\mathcal{M}, \mathfrak{X} \models^{\exists\forall} \forall^{-\emptyset} x \varphi \stackrel{sem.}{\Leftrightarrow} \mathcal{M}, \overline{\mathfrak{X}} \models^{\forall\exists} \forall^{-\emptyset} x \varphi \stackrel{sem.}{\Leftrightarrow} \mathcal{M}, \text{EXT}_{-\emptyset}(\overline{\mathfrak{X}}, x) \models^{\forall\exists} \varphi \stackrel{Thm. 2.2}{\Leftrightarrow} \mathcal{M}, \overline{\text{EXT}_{-\emptyset}(\mathfrak{X}, x)} \models^{\exists\forall} \varphi \stackrel{Lemma 3.2}{\Leftrightarrow} \mathcal{M}, \text{CYL}(\mathfrak{X}, x) \models^{\exists\forall} \varphi \stackrel{ind.hp.}{\Leftrightarrow}$ there

is $X \in \text{CYL}(\mathfrak{X}, x)$ such that $\mathcal{M}, X \models_{\text{DIF}}^{\forall} \varphi \stackrel{\text{def.}}{\Leftrightarrow}$ there is $X \in \mathfrak{X}$ such that $\mathcal{M}, \text{CYL}(X, x) \models_{\text{DIF}}^{\forall} \varphi \stackrel{\text{DIF-}^{\text{sem.}}}{\Leftrightarrow}$ there is $X \in \mathfrak{X}$ such that $\mathcal{M}, X \models_{\text{DIF}}^{\forall} \forall^{-\emptyset} x \varphi$.

We turn now to proving Item 2. We proceed by structural induction on the formula φ .

(base case) If $\varphi = \text{P}(\vec{x})$ or $\varphi = \neg\text{P}(\vec{x})$, then the property holds by the semantics rules.

(inductive cases) Suppose that the property holds for the subformulas of φ .

($\varphi = \varphi_1 \wedge \varphi_2$) We assume that $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi_1 \wedge \varphi_2$ and we show that for all teams $X \in \mathfrak{X}$, it holds that $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \varphi_1 \wedge \varphi_2$, which amount to showing that for all teams $X \in \mathfrak{X}$ and $(X_1, X_2) \in \text{PART}(X)$, it holds that $\mathcal{M}, X_1 \models_{\text{DIF}}^{\exists} \varphi_1$ or $\mathcal{M}, X_2 \models_{\text{DIF}}^{\exists} \varphi_2$. To this end, we let $X \in \mathfrak{X}$ and $(X_1, X_2) \in \text{PART}(X)$. By Item 2 of Lemma 3.3, there are $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$, $Y_1 \in \overline{\mathfrak{X}_1}$, and $Y_2 \in \overline{\mathfrak{X}_2}$, such that $Y_1 \subseteq X_1$ and $Y_2 \subseteq X_2$. From $\mathcal{M}, \mathfrak{X} \models^{\forall\exists} \varphi_1 \wedge \varphi_2$, it follows that $\mathcal{M}, \overline{\mathfrak{X}} \models^{\exists\forall} \varphi_1 \wedge \varphi_2$. By semantics, for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ it holds that $\mathcal{M}, \mathfrak{X}_1 \models^{\exists\forall} \varphi_1$ or $\mathcal{M}, \mathfrak{X}_2 \models^{\exists\forall} \varphi_2$, which, by Theorem 2.2, amounts to saying that for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ it holds that $\mathcal{M}, \overline{\mathfrak{X}_1} \models^{\forall\exists} \varphi_1$ or $\mathcal{M}, \overline{\mathfrak{X}_2} \models^{\forall\exists} \varphi_2$. By inductive hypothesis, for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ it holds that $\mathcal{M}, X_1 \models_{\text{DIF}}^{\exists} \varphi_1$ for all $X_1 \in \overline{\mathfrak{X}_1}$ or it holds that $\mathcal{M}, X_2 \models_{\text{DIF}}^{\exists} \varphi_2$ for all $X_2 \in \overline{\mathfrak{X}_2}$. Equivalently, for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$, $X_1 \in \overline{\mathfrak{X}_1}$, and $X_2 \in \overline{\mathfrak{X}_2}$, it holds that $\mathcal{M}, X_1 \models_{\text{DIF}}^{\exists} \varphi_1$ or $\mathcal{M}, X_2 \models_{\text{DIF}}^{\exists} \varphi_2$. Therefore, we have that $\mathcal{M}, Y_1 \models_{\text{DIF}}^{\exists} \varphi_1$ or $\mathcal{M}, Y_2 \models_{\text{DIF}}^{\exists} \varphi_2$, and, due to $Y_1 \subseteq X_1$ and $Y_2 \subseteq X_2$, and thanks to Item 2 of Remark 3.2, we conclude $\mathcal{M}, X_1 \models_{\text{DIF}}^{\exists} \varphi_1$ or $\mathcal{M}, X_2 \models_{\text{DIF}}^{\exists} \varphi_2$.

Conversely, assume that for all teams $X \in \mathfrak{X}$, it holds that $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \varphi_1 \wedge \varphi_2$, which amounts to saying that for all $X \in \mathfrak{X}$ and $(X_1, X_2) \in \text{PART}(X)$, it holds that $\mathcal{M}, X_1 \models_{\text{DIF}}^{\exists} \varphi_1$ or $\mathcal{M}, X_2 \models_{\text{DIF}}^{\exists} \varphi_2$. First, we show that for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$, $X_1 \in \overline{\mathfrak{X}_1}$, and $X_2 \in \overline{\mathfrak{X}_2}$, it holds that $\mathcal{M}, X_1 \models_{\text{DIF}}^{\exists} \varphi_1$ or $\mathcal{M}, X_2 \models_{\text{DIF}}^{\exists} \varphi_2$. To this end, let $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$, $X_1 \in \overline{\mathfrak{X}_1}$, and $X_2 \in \overline{\mathfrak{X}_2}$. By Item 1 of Lemma 3.3, there exists a team $X \in \mathfrak{X}$ such that $X \subseteq X_1 \cup X_2$. Let $X'_1 = X_1 \cap X$ and $X'_2 = X \setminus X'_1$. Clearly, $(X'_1, X'_2) \in \text{PART}(X)$, $X'_1 \subseteq X_1$, and $X'_2 \subseteq X_2$. By assumption, it holds that $\mathcal{M}, X'_1 \models_{\text{DIF}}^{\exists} \varphi_1$ or $\mathcal{M}, X'_2 \models_{\text{DIF}}^{\exists} \varphi_2$. From $X'_1 \subseteq X_1$ and $X'_2 \subseteq X_2$, and thanks to Item 2 of Remark 3.2, it follows $\mathcal{M}, X_1 \models_{\text{DIF}}^{\exists} \varphi_1$ or $\mathcal{M}, X_2 \models_{\text{DIF}}^{\exists} \varphi_2$. Therefore, we have showed that for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$, $X_1 \in \overline{\mathfrak{X}_1}$, and $X_2 \in \overline{\mathfrak{X}_2}$, it holds that $\mathcal{M}, X_1 \models_{\text{DIF}}^{\exists} \varphi_1$ or $\mathcal{M}, X_2 \models_{\text{DIF}}^{\exists} \varphi_2$. This amount to saying that for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$, it holds that $\mathcal{M}, X_1 \models_{\text{DIF}}^{\exists} \varphi_1$ for all $X_1 \in \overline{\mathfrak{X}_1}$ or it holds that $\mathcal{M}, X_2 \models_{\text{DIF}}^{\exists} \varphi_2$ for all $X_2 \in \overline{\mathfrak{X}_2}$. By inductive hypothesis, we have that for all $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$,

it holds that $\mathcal{M}, \overline{\mathfrak{X}}_1 \models^{\forall\exists} \varphi_1$ or $\mathcal{M}, \overline{\mathfrak{X}}_2 \models^{\forall\exists} \varphi_2$, which eventually amounts to saying $\mathcal{M}, \overline{\mathfrak{X}} \models^{\forall\exists} \varphi_1 \wedge \varphi_2$.

$(\varphi = \varphi_1 \vee \varphi_2)$ $\mathcal{M}, \overline{\mathfrak{X}} \models^{\forall\exists} \varphi_1 \vee \varphi_2 \stackrel{sem.}{\Leftrightarrow}$ there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ such that $\mathcal{M}, \overline{\mathfrak{X}}_1 \models^{\forall\exists} \varphi_1$ and $\mathcal{M}, \overline{\mathfrak{X}}_2 \models^{\forall\exists} \varphi_2 \stackrel{ind.hp.}{\Leftrightarrow}$ there is $(\mathfrak{X}_1, \mathfrak{X}_2) \in \text{PART}(\overline{\mathfrak{X}})$ such that for all $X_1 \in \mathfrak{X}_1$ it holds $\mathcal{M}, X_1 \models_{\text{DIF}}^{\exists} \varphi_1$ and for all $X_2 \in \mathfrak{X}_2$ it holds $\mathcal{M}, X_2 \models_{\text{DIF}}^{\exists} \varphi_2 \Leftrightarrow$ for all $X \in \overline{\mathfrak{X}}$ it holds that $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \varphi_1$ or $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \varphi_2 \stackrel{\text{DIF-}^{\exists} \text{sem.}}{\Leftrightarrow}$ for all $X \in \overline{\mathfrak{X}}$ it holds that $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \varphi_1 \vee \varphi_2$.

$(\varphi = \exists^{-\emptyset} x \varphi)$ $\mathcal{M}, \overline{\mathfrak{X}} \models^{\forall\exists} \exists^{-\emptyset} x \varphi \stackrel{sem.}{\Leftrightarrow} \mathcal{M}, \overline{\mathfrak{X}} \models^{\exists\forall} \exists^{-\emptyset} x \varphi \stackrel{sem.}{\Leftrightarrow} \mathcal{M}, \text{EXT}_{-\emptyset}(\overline{\mathfrak{X}}, x) \models^{\exists\forall} \varphi \stackrel{Thm. 2.2}{\Leftrightarrow} \mathcal{M}, \overline{\text{EXT}_{-\emptyset}(\overline{\mathfrak{X}}, x)} \models^{\forall\exists} \varphi \stackrel{\text{Lemma 3.2}}{\Leftrightarrow} \mathcal{M}, \text{CYL}(\overline{\mathfrak{X}}, x) \models^{\forall\exists} \varphi \stackrel{ind.hp.}{\Leftrightarrow}$ for all $X \in \text{CYL}(\overline{\mathfrak{X}}, x)$ it holds that $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \varphi \stackrel{def.}{\Leftrightarrow}$ for all $X \in \overline{\mathfrak{X}}$ it holds that $\mathcal{M}, \text{CYL}(X, x) \models_{\text{DIF}}^{\exists} \varphi \stackrel{\text{DIF-}^{\exists} \text{sem.}}{\Leftrightarrow}$ for all $X \in \overline{\mathfrak{X}}$ it holds that $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \exists^{-\emptyset} x \varphi$.

$(\varphi = \forall^{\pm W} x \varphi)$ $\mathcal{M}, \overline{\mathfrak{X}} \models^{\forall\exists} \forall^{\pm W} x \varphi \stackrel{sem.}{\Leftrightarrow} \mathcal{M}, \text{EXT}_{\pm W}(\overline{\mathfrak{X}}, x) \models^{\forall\exists} \varphi \stackrel{ind.hp.}{\Leftrightarrow}$ for all $X \in \text{EXT}_{\pm W}(\overline{\mathfrak{X}}, x)$ it holds that $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \varphi \stackrel{def.}{\Leftrightarrow}$ for all $X \in \overline{\mathfrak{X}}$ and $F \in \text{Rsp}_{\pm W}$ it holds that $\mathcal{M}, \text{EXT}(X, F, x) \models_{\text{DIF}}^{\exists} \varphi \stackrel{\text{DIF-}^{\exists} \text{sem.}}{\Leftrightarrow}$ for all $X \in \overline{\mathfrak{X}}$ it holds that $\mathcal{M}, X \models_{\text{DIF}}^{\exists} \forall^{\pm W} x \varphi$. \square

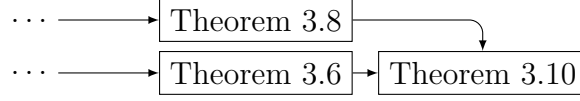


Figure 3.2 – Dependency graph of Theorem 3.10.

Theorem 3.10 (ADF-SO Interpretation). *For every ADF formula φ in pnf with quantifier prefix $\varphi \in \mathbf{Q}$ over a signature \mathcal{S} , set of variables $\text{SUP}(\varphi) \subseteq V \subseteq \text{Var}$ with $V \cap \text{VAR}(\varphi) = \emptyset$, and relation symbol $P_{\mathfrak{X}} \notin \mathcal{S}$ with $\text{AR}(P_{\mathfrak{X}}) = |V| + 1$, there exist two SO sentences $\psi_{\exists\forall}$ and $\psi_{\forall\exists}$ over signature $\mathcal{S} \uplus \{P_{\mathfrak{X}}\}$ such that, for all \mathcal{S} -structures \mathcal{M} and non-null hyperteam $\mathfrak{X} \in \text{HypTeams}(V)$ with $|\mathfrak{X}| \leq |\mathcal{M}|$, the following equivalence holds true: $\mathcal{M}, \overline{\mathfrak{X}} \models^{\alpha} \varphi$ iff $\mathcal{M} \uplus \{\text{REL}(\overline{\mathfrak{X}})\} \models_{\text{SO}} \psi_{\alpha}$.*

Proof. Let \vec{x} be a vector of all the variables in V . As first step, consider a formula $\varphi = \varphi\phi$ in pnf, where φ is a quantifier prefix and ϕ a quantifier-free subformula. Then, by Theorem 3.6, we transform φ into the equivalent Meta-ADF formula $\text{HSP}_{\varphi}(\varphi)\phi$. Naturally, $\text{HSP}_{\varphi}(\varphi) = (Q_i^{+W_i} x_i)_{i=1}^k$, for some $k \in \mathbb{N}$, where $W_i \subseteq \text{Var}$ and $Q_i \in \{\Sigma, \Pi\}$. Now, let $\widehat{\varphi} \stackrel{\text{def}}{=} (\widehat{Q}_i f_i)_{i=1}^k$ be the second-order function-quantifier prefix, where

1. the arity of each function symbol f_i equals the number of variables x_i depend on, i.e., $\text{AR}(f_i) = |W_i|$, and

2. each second-order quantifier symbol $\widehat{Q}_i \in \{\exists, \forall\}$ is existential iff the meta-quantifier symbol Q_i is existential.

At this point, the SO sentences $\psi_{\exists\forall}$ and $\psi_{\forall\exists}$ can be defined as follows, where $y \notin V \cup \text{VAR}(\varphi)$ and $\widehat{\phi}$ is obtained from the subformula ϕ by replacing each occurrence of a variable x_i with the corresponding term $f_i(\vec{w}_i)$, where \vec{w}_i is a vector of all the variables in W_i :

1. $\psi_{\exists\forall} \stackrel{\text{def}}{=} \widehat{\phi}. \exists y. (\exists \vec{x}. \mathbf{P}(\vec{x}y)) \wedge (\forall \vec{x}. \neg \mathbf{P}(\vec{x}y) \vee \widehat{\phi})$;
2. $\psi_{\forall\exists} \stackrel{\text{def}}{=} \widehat{\phi}. \forall y. \neg(\exists \vec{x}. \mathbf{P}(\vec{x}y)) \vee (\exists \vec{x}. \mathbf{P}(\vec{x}y) \wedge \widehat{\phi})$.

To conclude, the correctness of the translation can be proved by a simple induction on the length of the quantifier prefix φ , where, as base case, we exploit the extension of Theorem 3.8 to **Meta-ADIF** . \square

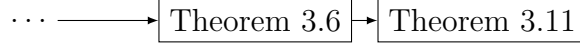


Figure 3.3 – Dependency graph of Theorem 3.11.

Theorem 3.11 (SO-ADF Interpretation). *For every SO sentence ψ over a signature \mathcal{S} , relation symbol $P_{\vec{x}} \in \mathcal{S}$, and sequence of variables $\vec{x} \in V^{\text{AR}(P_{\vec{x}})}$, with $\text{VAR}(\psi) \cap \vec{x} = \emptyset$, i.e., no variable in \vec{x} occurs in ψ , there exists an ADF formula φ in pnf over signature $\mathcal{S} \setminus P_{\vec{x}}$ with $\text{SUP}(\varphi) = \text{FREE}(\varphi) = \vec{x}$ such that, for all \mathcal{S} -structures \mathcal{M} , the following equivalence holds true: $\mathcal{M} \models_{\text{SO}} \psi$ iff $\mathcal{M} \setminus P_{\vec{x}}, \{ \text{TEAM}(P_{\vec{x}}^{\mathcal{M}}, \vec{x}) \} \models^{\exists\forall} \varphi$.*

Proof. To begin with, let us assume *without loss of generality* (see [KN09] for a proof) that the SO sentence ψ is of the form

$$(Q_i f_i)_{i=1}^k. \forall \vec{z}. (\mathbf{P}(\vec{y}) \leftrightarrow \tau_1 = \tau_2) \wedge \phi,$$

which in addition complies with the following constraints:

1. $\vec{y} \subseteq \vec{z}$, i.e., the vector of variables \vec{y} used in the atom $\mathbf{P}(\vec{y})$ is included in the vector of universally-quantified variables \vec{z} ;
2. every function f_i only appears in a single term $\tau_{f_i} = f_i(\vec{w}_i)$;
3. every term τ (including τ_1 and τ_2) is of the form $f_i(\vec{w})$, for some index $i \in \llbracket 0, \dots, k \rrbracket$ and vector of variables $\vec{w} \subseteq \vec{z}$;
4. the relation \mathbf{P} does not occur in the FO formula ϕ .

Now, let $\wp \stackrel{\text{def}}{=} (\widehat{Q_i^{+W_i} z_i})_{i=k}^1$ be the first-order quantifier prefix, where

1. the set of dependence variables W_i coincides with the vector of variables \vec{w}_i used in the term τ_{f_i} corresponding to the function f_i , and
2. each first-order quantifier symbol $\widehat{Q}_i \in \{\exists, \forall\}$ is existential iff the second-order quantifier symbol Q_i is existential.

Notice that the order of quantification is reversed *w.r.t* the one in $(Q_i f_i)_{i=1}^k$. At this point, the ADF formula φ can be defined as follows, where

1. $(\vec{y} = \vec{x})$ denotes a shortcut for a conjunction of equalities between corresponding variables in \vec{y} and \vec{x} ,
2. z_1' and z_2' are the variables corresponding to the functions used in the terms τ_1 and τ_2 , and
3. ϕ' is the FO formula obtained from ϕ by replacing each occurrence of a term τ_{f_i} with the corresponding variable z_i :

$$\varphi \stackrel{\text{def}}{=} \forall \vec{z}. \wp. ((\vec{y} = \vec{x}) \leftrightarrow z_1' = z_2') \wedge \phi'.$$

To conclude, the correctness of the translation can be shown by first applying Theorem 3.6 to φ , obtaining the **Meta-ADF** formula

$$\text{HSP}_\varphi(\wp). \forall \vec{z}. ((\vec{y} = \vec{x}) \leftrightarrow z_1' = z_2') \wedge \phi',$$

and then proceeding with a standard induction on the length of the quantifier prefix $(Q_i f_i)_{i=1}^k$. □

3.C Proofs of Section 3.3

In order to prove Theorem 3.12, we shall first prove two additional lemmas. The first one states a Skolemisation property for **Meta-ADIF**. A sentence of **Meta-ADIF** in prenex form that only has meta quantifiers Σ or Π can be viewed as an SO formula. Therefore, given a **Meta-ADIF** formula $\varphi \stackrel{\text{def}}{=} \Pi^{\pm \vec{U}} \vec{y} \Sigma^{\pm W} x \phi$, we can use classic Skolem results to define a function SKO_x for the first existentially quantified variable x such that if \mathcal{F} is a function assignment of variables (universally) quantified before x , then $\mathcal{F}[x \mapsto \text{SKO}_x(\mathcal{F})]$ satisfies the subformula ϕ that follows the quantification of x iff \mathcal{F} satisfies $\Sigma^{\pm W} x \phi$. First, we introduce some notations. For a quantifier prefix $\wp = Q_0^{+W_0} x_0 \dots Q_n^{+W_n} x_n$ and a

quantifier $Q \in \{\Sigma, \Pi\}$, the set $\text{VAR}_Q(\varphi) = \{x_i \mid Q_i = Q\}$ collects all the variables quantified in φ using the quantifier Q . A *Skolemisation* for φ is a sequence $(\text{SKO}_{x_i} : (\prod_{j < i} \text{Rsp}_{W_j}) \rightarrow \text{Rsp}_{W_i})_{x_i \in \text{VAR}_\Sigma(\varphi)}$ of functions, one for each existentially quantifier variable x_i of φ and each one intuitively mapping the interpretations of the variables preceding x_i in φ to some interpretation for x_i . A *Skolem extension* of \mathcal{F} w.r.t a Skolemisation $(\text{SKO}_{x_i})_{x_i \in \text{VAR}_\Sigma(\varphi)}$ for φ is a function assignment \mathcal{F}' such that: (i) $\text{DOM}(\mathcal{F}') = \text{DOM}(\mathcal{F}) \cup \text{VAR}(\varphi)$; (ii) $\mathcal{F}'(x) = \mathcal{F}(x)$, for $x \in \text{DOM}(\mathcal{F}) \setminus \text{VAR}(\varphi)$; and (iii) $\mathcal{F}'(x_i) = \text{SKO}_{x_i}((\mathcal{F}'(x_j))_{j < i})$, if $x_i \in \text{VAR}_\Sigma(\varphi)$. Observe that \mathcal{F} assigns a function to each variable in φ , using the Skolemisation for the existentially quantified variables and arbitrary functions for the universally quantified ones. We can now state the following lemma.

Lemma 3.7 (Meta-ADIF Skolemisation). *Let \mathfrak{X} be a hyperteam, \mathcal{F} a function assignment and $\varphi = \varphi\psi$ a Meta-ADIF formula in prenex form, where $\varphi = Q_0^{+W_0}x_0 \dots Q_n^{+W_n}x_n$ with $Q_i \in \{\Sigma, \Pi\}$ for $i \leq n$. The following holds: $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \varphi$ iff there exists a Skolemisation $(\text{SKO}_{x_i})_{x_i \in \text{VAR}_\Sigma(\varphi)}$ for φ such that $\mathcal{M}, \mathcal{F}', \mathfrak{X} \models^\alpha \psi$, for all Skolem extensions \mathcal{F}' of \mathcal{F} w.r.t $(\text{SKO}_{x_i})_{x_i \in \text{VAR}_\Sigma(\varphi)}$.*

Proof. We prove the result by induction on the size of $\text{VAR}_\Sigma(\varphi)$.

Base case $\text{var}_\Sigma(\varphi) = \emptyset$. The only Skolemisation for φ is the empty sequence of functions. A simple application of the semantic rules for the universal quantifiers, applied to Πx_i for each $i \leq n$, gives the result.

Inductive case. Suppose the property holds for all formulas with $|\text{VAR}_\Sigma(\varphi)| < n$. We construct SKO_x for each $x \in \text{VAR}_\Sigma(\varphi)$ with the desired properties. Let i_0 be the smallest integer such that $x_{i_0} \in \text{VAR}_\Sigma(\varphi)$, so that we can set $\varphi = \Pi^{+W_0}x_0 \dots \Pi^{+W_{i_0-1}}x_{i_0-1} \Sigma^{+W_{i_0}}x_{i_0} \varphi'$ and $\varphi' = Q_{i_0+1}^{+W_{i_0+1}}x_{i_0+1} \dots Q_n^{+W_n}x_n \psi = \varphi'\psi$. By application of the semantic rules for the $i_0 - 1$ universal quantifiers and the last existential one, we obtain that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \varphi$ iff for every sequence of functions $(F_{x_j})_{j < i_0}$, with $F_{x_j} \in \text{Rsp}_{W_j}$, there is function $F_{x_{i_0}} \in \text{Rsp}_{W_{i_0}}$ such that $\mathcal{M}, \mathcal{F}', \mathfrak{X} \models^\alpha \varphi'$, with $\mathcal{F}' = \mathcal{F}[x_0 \mapsto F_{x_0}, \dots, x_{i_0} \mapsto F_{x_{i_0}}]$. Now, since $\text{VAR}_\Sigma(\varphi') < n$, by inductive hypothesis $\mathcal{M}, \mathcal{F}', \mathfrak{X} \models^\alpha \varphi'$ iff there is Skolemisation $(\text{SKO}'_{x_i})_{x_i \in \text{VAR}_\Sigma(\varphi')}$ for φ' such that $\mathcal{M}, \mathcal{F}'', \mathfrak{X} \models^\alpha \psi$, for every Skolem extension \mathcal{F}'' of \mathcal{F}' w.r.t SKO' . We then have that $\mathcal{M}, \mathcal{F}, \mathfrak{X} \models^\alpha \varphi$ iff for all sequences of functions $(F_{x_j})_{j < i_0}$, there exist a function $F_{x_{i_0}}$ and a Skolemisation $(\text{SKO}'_{x_i})_{x_i \in \text{VAR}_\Sigma(\varphi')}$ for φ' such that $\mathcal{M}, \mathcal{F}'', \mathfrak{X} \models^\alpha \psi$, for every Skolem extension \mathcal{F}'' of $\mathcal{F}[x_0 \mapsto F_{x_0}, \dots, x_{i_0} \mapsto F_{x_{i_0}}]$ w.r.t $(\text{SKO}'_{x_i})_{x_i \in \text{VAR}_\Sigma(\varphi')}$. Since the choices of $F_{x_{i_0}}$

and of the Skolemisation $(\text{SKO}'_{x_i})_{x_i \in \text{VAR}_\Sigma(\wp')}$ depend on the sequence $(F_{x_j})_{j < i_0}$, obviously there exists a Skolemisation $(\text{SKO}_{x_i})_{x_i \in \text{VAR}_\Sigma(\wp)}$ for \wp such that $\mathcal{M}, \mathcal{F}'', \mathfrak{X} \models^\alpha \psi$, for all Skolem extension \mathcal{F}'' of \mathcal{F} w.r.t $(\text{SKO}_{x_i})_{x_i \in \text{VAR}_\Sigma(\wp)}$. Indeed, for all sequences $(F_{x_j})_{j < i_0}$, the function $F_{x_{i_0}}$ and the Skolemisation $(\text{SKO}'_{x_{i_k}})_{x_i \in \text{VAR}_\Sigma(\wp')}$ defined as follow satisfy the properties shown above:

- $F_{x_{i_0}} = \text{SKO}_{x_{i_0}}((F_{x_j})_{j < i_0})$;
- $\text{SKO}'_{x_{i_k}}((F_{x_j})_{i_0 < j < i_k}) = \text{SKO}_{x_{i_k}}((F_{x_j})_{j < i_k})$, for all $x_{i_k} \in \text{VAR}_\Sigma(\wp')$ and sequence of functions $(F_{x_j})_{i_0 < j < i_k}$.

□

The second lemma states a property of the independence game $\mathcal{G}_\varphi^{\mathcal{M}}$ defined in Construction 3.2 for an ADIF sentence φ and a structure \mathcal{M} : after a history h , no matter how the functions in each bucket are chosen, the only assignment that is coherent with the functions in the bucket is the one associated with the last position of h . In the following, we consider an ADIF sentence $\varphi = \wp\psi$ in prenex form, where $\wp = Q_0^{+W_0}x_0 \dots Q_n^{+W_n}x_n$ with $Q_i \in \{\forall, \exists\}$ for $i \in \llbracket 0, \dots, n \rrbracket$, and ψ quantifier free. For every subformula $\phi = Q_i^{+W_i}x_i\phi'$, we rename the buckets $\mathbf{R}_\phi(h)$ by $\mathbf{R}_{x_i}(h)$ and associate priorities with variables by setting $\text{PRT}(x_i) = \text{PRT}(\phi)$. Let $\mathbf{Buck} \stackrel{\text{def}}{=} 2^{\text{Rsp}}$ denote the set of all buckets. For convenience, we set $V_X = \{x_0, \dots, x_n\}$ and $V_{X_i} = \{x_0, \dots, x_i\}$, $V_{X\exists} = \{x_i \in V_X \mid Q_i = \exists\}$ and $V_{X\forall} = V_X \setminus V_{X\exists}$. We also introduce choice functions over buckets. Basically, a choice function over buckets chooses, for each variable x , a function F in the bucket of x . It takes both $\mathbf{R}_x(h)$ and x in input because there might be multiple variables with the same bucket (for instance, when they all depend exactly on the same variables and the same value have been played for all of them during the play).

$$\text{ChcBuck} = \{\mathbf{F} : (\mathbf{Buck} \times V_X) \rightarrow \text{Rsp} \mid \forall \mathbf{R} \in \mathbf{Buck}, \forall x \in V_X, \mathbf{F}(\mathbf{R}, x) \in \mathbf{R}\}$$

Given a function $F_j \in \text{Rsp}_{+W_j}$ for each variable $x_j \in V_{X_i}$ with $i \leq n$, we define $\chi_{(F_j)_{j \leq i}} \in \mathbf{Asg}(V_{X_i})$ as the unique assignment χ such that $\chi(x_j) = F_j(\chi_{\text{FREE}(Q_j^{+W_j}x_j\phi')})$ for every $j \leq i$. We say that χ is *coherent* with $(F_j)_{j \leq i}$.

Lemma 3.8 (Buckets soundness). *For every choice function $\mathbf{F} \in \text{ChcBuck}$ over buckets and every history $h = h'p$ with $p = (\phi, \chi, \clubsuit)$ where $\clubsuit \in \{I, II\}$, the following holds:*

- if $\phi = Q_i^{+W_i}x_i\phi'$ and $\clubsuit = I$, it holds $\chi = \chi_{(\mathbf{F}(\mathbf{R}_{x_j}(h), x_j))_{j \leq i}}$.
- if $\phi = \psi$ or $\clubsuit = II$ it holds $\chi = \chi_{(\mathbf{F}(\mathbf{R}_{x_j}(h), x_j))_{j \leq n}}$.

Proof. We prove this lemma by induction on the history h .

For the base case history, the property is trivial.

For the induction case, suppose the lemma holds for a history $h = h'p'$ with $p' = (\phi', \chi', \clubsuit)$. Consider a history of the form hp . There are two cases to consider: either $\clubsuit = I$, or $\clubsuit = II$.

($\clubsuit = I$) There are again two cases to look at:

1. if $\phi' = \psi$ then the only possible successor position p in the game is (φ, χ', II) . So, by definition of the bucket and direct application of the inductive hypothesis, the property holds for hp .
2. if $\phi' = Q_i^{+W_i} x_i \phi$, then p is of the form (ϕ, χ, I) . The only bucket that might change is \mathbf{R}_{x_i} . By definition, any function $F \in \mathbf{R}_{x_i}(hp)$ satisfies $F(\chi') = \chi'(x_i)$ and the property holds for hp . Indeed, by inductive hypothesis, $\chi(x_j) = \chi'(x_j)$, for every x_j with $j < i$.

($\clubsuit = II$) If $\phi' = \psi$, there is no reachable position. So, the only possibility is $\phi' = Q_i^{+W_i} x_i \phi$. There are again two possibilities:

1. p is of the form (ϕ, χ', II) . In this case, by the definition of bucket and a direct application of the inductive hypothesis, the property immediately follows for hp .
2. p is of the form $(\phi, \check{\chi}[x_i \mapsto a], I)$, for some $v \in \mathbb{D}$ with $v \neq \chi'(x_i)$, with $\check{\chi} \stackrel{\text{def}}{=} \chi'_{\upharpoonright_{\text{FREE}(Q_i^{+W_i} x_i \phi)}}$. By inductive hypothesis, for every $F \in \mathbf{R}_{x_i}(h)$, we have $F(\check{\chi}) = \chi'(x_i)$. Then, $\mathcal{F}_\phi(\check{\chi}) \cap \mathbf{R}_{x_i}(h) = \emptyset$. Thus, $\mathbf{R}_{x_i}(hp) = \mathcal{F}_\phi(\check{\chi})$ and, by definition, we have $F(\check{\chi}_{\upharpoonright W_i}) = \check{\chi}(x_i)$, for every $F \in \mathbf{R}_{x_i}(hp)$. Since the other buckets have not changed, the property holds for hp . □

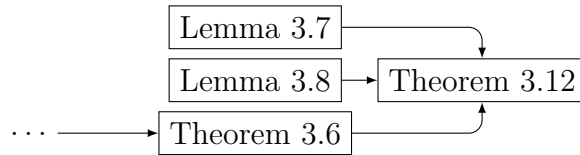


Figure 3.4 – Dependency graph of Theorem 3.12.

Theorem 3.12 (Game-Theoretic Semantics). *For a finite structure \mathcal{M} and an ADIF sentence ϕ in prenex form, the independence game $\mathcal{G}_\phi^{\mathcal{M}}$ defined in construction 3.2 is such that $\mathcal{M} \models \phi$ (resp., $\mathcal{M} \not\models \phi$) iff $\mathcal{G}_\phi^{\mathcal{M}}$ is won by Eloise (resp., Abelard).*

Proof. We prove that if the sentence is true in \mathcal{M} , then Eloise wins the game and if the sentence is false, then Abelard wins the game.

First, suppose that the sentence φ is true in \mathcal{M} . By Theorem 3.6, φ is equivalent to the Meta-ADIF sentence $\text{HSP}_\varphi(\wp)\psi$. So, by Lemma 3.7 and recalling that in $\text{HSP}_\varphi(\wp)$ the order of the quantifiers is reversed, we can conclude that there is a Skolemisation $(\text{SKO}_{x_i} : \prod_{j>i} \text{Rsp}_{W_j} \rightarrow \text{Rsp}_{W_i})_{x_i \in V_{X\exists}}$ such that $\mathcal{M}, \mathcal{F}, \{\{\emptyset\}\} \models \psi$, for every Skolem extension \mathcal{F} of the empty functional assignment *w.r.t* $(\text{SKO}_{x_i})_{x_i \in V_{X\exists}}$. We now define a strategy for Eloise and then prove that it is winning. Intuitively, the strategy consists in looking, by means of the buckets, at one possible function assignment of the variables controlled by Abelard and, then, applying what is prescribed by the Skolemisation $(\text{SKO}_{x_i})_{x_i \in V_{X\exists}}$ to select the values for the variables controlled by Eloise. Formally, let us fix a choice function $F \in \text{ChcBuck}$ on the buckets. Given a history h , we define F_i^h for $i \in \{0, \dots, n\}$ as follows. If $x_i \in V_{X\forall}$ then $F_i^h = F(\mathbf{R}_{x_i}(h), x_i)$, otherwise, $F_i^h = \text{SKO}_{x_i}((F_j^h)_{j>i})$. When Eloise has to make a move for the variable x_i at the history $h = h'p'$, with $p' = (\phi, \chi, _)$, she moves to the position $p = (\phi', \chi', _)$ with $\chi'(x_i) = F_i^h(\chi)$. Observe that this strategy does not depend on the current phase of the game but might trigger a challenge if $\chi \neq \chi'$.

Consider now a finite play $\pi = hp$, with $p = (\psi, \chi, II)$, compatible with the strategy. We define a choice function F^* as follows: for all $x_i \in X$

$$F^*(\mathbf{R}_{x_i}(\pi), x_i) = F_i^\pi$$

The function F^* is a choice function since, if $x_i \in V_{X\forall}$, by definition, $F_i^\pi \in \mathbf{R}_{x_i}(\pi)$ and if $x_i \in V_{X\exists}$, then because Eloise played according to F_i^π , this function is in the bucket of x_i . The Lemma 3.8 ensures that the assignment χ is coherent with $(F^*(\mathbf{R}_x(\pi), x))_{x \in V_X}$. By definition of $(\text{SKO}_{x_i})_{x_i \in V_{X\exists}}$, it holds that $\chi \models \psi$. Therefore, the play is won by Eloise.

Let us now consider an infinite play $\pi \in \text{Paths}_\omega$ compatible with the strategy. Toward a contradiction, suppose that the priority $\text{PRT}(\pi)$ of the play is odd. Then, there must be a variable $x_i \in V_{X\exists}$ such that (i) $\text{PRT}(x_i)$ appears infinitely often in $\text{PRT}\pi$ and (ii) for all $j > i$ the priority $\text{PRT}(x_j)$ appears only a finite number of time. Recall that if a variable x is not “caught cheating” after a finite prefix $\pi' = hp$ of π , then $\mathbf{R}_x(\pi') \subseteq \mathbf{R}_x(h)$. But then, since we assumed the domain to be finite, starting from some index N along the play π , the buckets for each x_j , with $j > i$, remain constant forever. Let us denote the constant bucket of x_j by \mathbf{R}_{x_j}' . Then, the strategy for x_i is for Eloise to chose the values of x_i by means of the function $F_i = \text{SKO}_{x_i}((F(\mathbf{R}_{x_j}'(\pi), x_j))_{j>i})$, which is constant since the

buckets of variables x_j do not change. By definition, for every prefix of π of size greater than N , the bucket of x_i contains F_i . Then the bucket of x_i is never emptied and x_i would never get “caught cheating”. This is a contradiction. We proved that if the sentence is true, then Eloise has a winning strategy in \mathcal{G}_φ^M .

The second part of the proof proceeds similarly, in that we can apply the same exact reasoning, with only the roles of Eloise and Abelard exchanged, to obtain a winning strategy for Abelard. \square

GOOD-FOR-GAME QUANTIFIED PROPOSITIONAL TEMPORAL LOGIC

ADIF works pretty well to reason about *single-round games*, a.k.a., *normal-form games* [NM44], or *finite-rounds games*, a.k.a., *extensive-form games* [Neu28; Kuh50; Kuh53]. Things, however, get much more complicated when *infinite-rounds games* come into play [GS53; Wol55]. For such a class of extensive-form games, indeed, plays are induced by infinite sequences of choices made by the players over time and a strategy dictates how a player at a given stage of a play responds to the choices made by the adversary up to that stage. Extending the quantification prefix to match the rounds would immediately lead to *infinitary logics*, such as the one proposed by [Kol85] and further studied by [HV94] (see also [HR76]). This technique has some interesting applications in logic [Hel89], computer science [Kai11], and even philosophy [FG17]. Besides its infinitary nature, however, this approach has also the drawback of heavily departing from the standard Tarskian viewpoint, as only non-compositional game-theoretic semantics have been provided.

A more viable route, instead, is to make the quantified variables x and y range over sequences of choices. For example, when the choices are simply *Boolean values*, *iterated Boolean games* are to be considered [GHW13; GHW15]. This approach makes us consider temporal logics such as *Linear-time Temporal Logic* (LTL for short) that are able to express properties over sequences of Boolean values. By adding quantifiers, we obtain *Quantified Propositional Temporal Logic* (QPTL for short): a quantified *propositional variable* defines a sequence of truth values along the timeline. However, when alternating quantifiers, say $\forall x \exists y$, all the Boolean values of the sequence quantified by $\exists y$ depends on the entire sequence quantified by $\forall x$, which is not desirable if we are to reason strategically. To retrieve strategies, we have to constrain this choice with temporal dependencies.

In this chapter we propose a novel semantics for QPTL, based on hyperteams. Similarly to semantics for ADIF, the semantics we propose here provides a compositional

formulation [SH01] and a game-theoretic interpretation of the quantifiers. The dependence of response functions on assignments allows us to impose various forms of temporal constraints among the variables. In particular, we investigate two specific forms, called *realizable* and *strongly-realizable*, that require response functions to choose the value of the variable at any given time instant based only upon the values dictated by the input assignment to the other variables up to that instant (possibly excluded). These are forms of independence constraints that make the choice of the value of a variable at a given time totally independent of the values that other variables assume in the future. The realizable restrictions are precisely what allows us to recover the correspondence between response functions and strategies and to reconcile the satisfiability and game solution problems, thus making the resulting version of QPTL, called *Good-for-Games QPTL* (GFG-QPTL), well suited to express game-theoretic concepts and a logical analogue of *Good-for-Games Automata* [HP06; BL19].

On the technical side, the hyperteam semantics under the realizable interpretation of the quantifiers leads to 2-EXPTIME decision procedures for both the satisfiability and model-checking problems. On the other hand, it does not give up expressiveness, as we show that the vanilla and realizable semantics turn out to be expressively equivalent. These results also show that the high complexity of the decision problems for vanilla QPTL stems from the fact that unrestricted dependencies among the quantified variables are allowed. The properties expressible by exploiting such unrestricted dependencies can, however, still be expressed under the realizable semantics via encoding of ω -regular automata, though with a non-elementary blowup.

The chapter is organized as follows. We first recall the definitions of LTL and QPTL in Section 4.1. Then we present the hyperteam semantics for GFG-QPTL in Section 4.2. Finally, we define its game-theoretic semantics in Section 4.3 that allows us to state some complexity results.

4.1 Recalls on temporal logics

For the whole chapter, as the considered logics are *propositional*, the first class citizen is now a *proposition* instead of a variable. The definitions of previous chapters are implicitly adapted accordingly.

4.1.1 Linear-time Temporal Logic

Linear-Time Temporal Logic (LTL for short) is a propositional logic introduced by Pnuelli [Pnu77] in order to formalize properties for system verification. The definitions of this section are inspired by those found in [DGL16a].

Definition 4.1. *An LTL formula is defined inductively as follows, where $p \in AP$ is an atomic propositions.*

$$\psi := p \mid \psi \wedge \psi \mid \neg\psi \mid X\psi \mid \psi U\psi$$

The language relies on two modal operators, X and U, read respectively “next” and “until”. Other Boolean connectives “ \vee ” or “ \rightarrow ” are defined as usual: $\psi \vee \psi' := \neg(\neg\psi \wedge \neg\psi')$ and $\psi \rightarrow \psi' := (\neg\psi) \vee \psi'$ and \top (true) can be defined as $p \vee \neg p$ with any proposition p .

Other temporal symbols may be expressed: \diamond (eventually) and \square (globally) and can be defined thanks to the U (until) and \top , where \top means true.

$$\diamond\psi = \top U\psi$$

$$\square\psi = \neg\diamond\neg\psi$$

Example 4.1. *Consider two atomic propositions $\{\mathbf{I_wear_my_coat}, \mathbf{cold}\}$, the formula $\psi = \square(\mathbf{I_wear_my_coat} U \neg\mathbf{cold})$ expresses “it is always the case that, $\mathbf{I_wear_my_coat}$ is true until \mathbf{cold} is false”. Remark that the formula allows $\mathbf{I_wear_my_coat}$ to be true even if \mathbf{cold} is not. Intuitively, if $\mathbf{I_wear_my_coat}$ represent whether I wear a coat and \mathbf{cold} whether the temperature is low, the formula express the fact that I am safe from cold (which can be true or false depending on when I wear the coat and when the temperature is low).*

We denote by $\text{PROP}(\psi)$ the set of atomic propositions that occur in formula ψ . For instance, consider the formula $\psi = p \wedge (qU\neg r)$ the atomic propositions of ψ is the set $\text{PROP}(\psi) = \{p, q, r\}$.

An LTL assignment (also called *trace* in the literature) specifies which atomic propositions are true at each time step. Formally, an LTL assignment $\chi \in \text{Asg}$ is an assignment over the domain $\{\top, \perp\}^\omega$.

For any natural number $t \in \mathbb{N}$, we denote by χ_{+t} the assignment corresponding to χ without the t first letters of each proposition’s word. Formally, χ_{+t} is defined by $\chi_{+t}(p)[n] = \chi(p)[n + t]$.

Example 4.2. Here is the definition of an example χ over $\{\mathbf{I_wear_my_coat}, \mathbf{cold}\}$.

$$\begin{aligned}\chi(\mathbf{I_wear_my_coat}) &= \top \perp \top \top \perp \dots \\ \chi(\mathbf{cold}) &= \top \perp \top \top \top \dots\end{aligned}$$

Definition 4.2 (LTL Tarski Semantics). *The statement saying that an LTL $\text{PROP}(\psi)$ -assignment χ satisfies an LTL formula ψ is written $\chi \models_{LTL} \psi$. The definition of the semantics is given inductively on the structure of the formula.*

$$\begin{aligned}\chi \models_{LTL} p & \quad \text{iff} \quad \chi(p)[0] = \top \\ \chi \models_{LTL} \neg\psi & \quad \text{iff} \quad \chi \not\models_{LTL} \psi \text{ does not hold} \\ \chi \models_{LTL} \psi_1 \wedge \psi_2 & \quad \text{iff} \quad \chi \models_{LTL} \psi_1 \text{ and } \chi \models_{LTL} \psi_2 \\ \chi \models_{LTL} X\psi & \quad \text{iff} \quad \chi_{+1} \models_{LTL} \psi \\ \chi \models_{LTL} \psi_1 U\psi_2 & \quad \text{iff} \quad \text{there exist a natural number } t \text{ such that } \chi_{+t} \models_{LTL} \psi_2 \\ & \quad \text{and for every integer } t' < t, \chi_{+t'} \models_{LTL} \psi_1.\end{aligned}$$

Example 4.3. Consider the formula $\psi = \Box(\mathbf{I_wear_my_coat}U\neg\mathbf{cold})$ from the Example 4.1 and the assignment χ defined in Example 4.2. The assignment χ does not satisfy the formula ψ because $\chi_{\mathbf{I_wear_my_coat}}(4) = \perp$ and $\chi_{\mathbf{cold}}(4) = \top$. Intuitively, at time 4, the considered person does not wear a coat even if the temperature is low.

Remark that formula ψ would be satisfied in an assignment where $\mathbf{I_wear_my_coat}$ always holds. Indeed, always wearing a coat protects from cold.

4.1.2 Quantified Propositional Temporal Logic

We can extend LTL with propositional quantifiers and obtain Quantified Propositional Temporal Logic (QPTL for short).

For convenience, we provide a syntax for QPTL where quantifications do not occur within temporal operators. This is equivalent to the original logic, thanks to the *prenex normal form* (pnf, for short) property enjoyed by QPTL [Sis83], which allows to move quantifiers outside temporal operators.

Definition 4.3 (QPTL Syntax). *The Quantified Propositional Temporal Logic is the set of formulas built accordingly to the following context-free grammar, where $\psi \in LTL$ and $p \in AP$:*

$$\varphi := \psi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists p. \varphi \mid \forall p. \varphi.$$

The classic semantics is also given in terms of LTL assignments.

The *satisfaction relation* \models_{QPTL} between an assignment χ and a QPTL formula φ is defined below. As usual, by $\text{FREE}(\varphi)$ we denote the set of propositions free in φ .

Definition 4.4 (QPTL Tarski Semantics). *The Tarski-semantics relation $\chi \models_{\text{QPTL}} \varphi$ is inductively defined as follows, for all QPTL formulas φ and assignments $\chi \in \text{Asg}(P)$ with $\text{FREE}(\varphi) \subseteq P$.*

1. $\chi \models_{\text{QPTL}} \psi$, if $\chi \models_{\text{LTL}} \psi$, whenever ψ is an LTL formula;
2. the semantics of Boolean connectives is defined as usual;
3. for all atomic propositions $p \in AP$:
 - (a) $\chi \models_{\text{QPTL}} \exists p. \phi$ if $\chi[p \mapsto t] \models_{\text{QPTL}} \phi$, for some $t \in \{\top, \perp\}^\omega$;
 - (b) $\chi \models_{\text{QPTL}} \forall p. \phi$ if $\chi[p \mapsto t] \models_{\text{QPTL}} \phi$, for all $t \in \{\top, \perp\}^\omega$.

4.2 Dynamic Reasoning

In QPTL, a Skolem function cannot be interpreted as a strategy in the game-theoretic sense anymore, since its value at a given stage depends on the entire evaluation of its argument p , namely the entire sequence of choices made by the adversary, including all the future ones. By contrast, a strategy for a player can only dictate, step by step, what its responses should be, depending on the choices made so far by its opponent. What that means is that, in principle, the satisfiability and the game solution problems do not coincide anymore.

4.2.1 Realizable responses

Given a set of assignments $\text{Asg}(P)$ over some $P \subseteq AP$, a *realizable quantification* *w.r.t* a proposition $p \in P$ should choose, for each assignment $\chi \in \text{Asg}(P)$, a temporal valuation $t: \mathbb{N} \rightarrow \{\top, \perp\}$ in such a way that, intuitively, at each instant of time $k \in \mathbb{N}$, the value $t[k]$ of t at k only depends on the values $\chi(p)[t]$ of the temporal valuation $\chi(p)$ at the instants of time $t \leq k$; this means that $t[k]$ is independent of the values $\chi(p)[t]$ at any future instant $t > k$. To be more precise, consider two assignments $\chi_1, \chi_2 \in \text{Asg}(P)$ that may differ only on p strictly after k . Then, the response $F \in \text{Rsp}$ interpreting a quantification realizable *w.r.t* p must return the same value at k as a reply to both χ_1 and χ_2 , i.e., $F(\chi_1)[k] = F(\chi_2)[k]$; in other words, $F(\chi)[k]$ cannot exploit the knowledge

of the values $\chi(p)[t]$, for each $p \in \text{DOM}(\chi)$ with $t > k$. An analogous concept has been introduced in SL [MMPV14] and will be discussed in the next chapter. A stronger notion of realizability, similar to one reported by [GBM18], requires the response F to satisfy the above equality when χ_1 and χ_2 only (possibly) differ on p for $t \geq k$ and leads to the concept of *strongly realizable quantification*. In game-theoretic terms, the interpretation of a *realizable quantifier w.r.t p* requires the corresponding player to choose the value of a proposition at each round only based on the choices for p made by the adversary up to that round. For a *strongly realizable quantifier*, instead, the adversary keeps its choice for p at the current round hidden and the player can only access the choices made for p at previous rounds. Definitions 4.5 and 4.6 formalize these fundamental concepts.

Definition 4.5 (Assignment Distinguishability). *Let $\chi_1, \chi_2 \in \text{Asg}(P)$ be two assignments over some set $P \subseteq AP$ of propositions, $p \in P$ one of these propositions, and $k \in \mathbb{N}$ a number. Then, χ_1 and χ_2 are (p, k) -strict distinguishable (resp., (p, k) -distinguishable), in symbols $\chi_1 \approx_p^{>k} \chi_2$ (resp., $\chi_1 \approx_p^{\geq k} \chi_2$), if the following properties hold:*

1. $\chi_1(q) = \chi_2(q)$, for all atomic propositions $q \in P$ with $q \neq p$;
2. $\chi_1(p)[t] = \chi_2(p)[t]$, for all time instants $t \leq k$ (resp., $t < k$).

The notion of (p, k) -strict distinguishability (resp., (p, k) -distinguishability) allows us to identify all the assignments that can only differ on the proposition p at some time instant $t > k$ (resp., $t \geq k$). Indeed, $\approx_p^{>k}$ (resp., $\approx_p^{\geq k}$) is an equivalence relation on $\text{Asg}(P)$, whose equivalence classes identify those assignments precisely. A *realizable* (resp., *strongly-realizable*) response must reply at time k uniformly to all $\approx_p^{>k}$ -equivalent (resp., $\approx_p^{\geq k}$ -equivalent) assignments.

Definition 4.6 (Realizable Response). *Let $F \in \text{Rsp}$ be a response function and $p \in P$ a proposition. Then, F is realizable (resp., strongly realizable) w.r.t p if $F(\chi_1)[k] = F(\chi_2)[k]$, for all numbers $k \in \mathbb{N}$ and pairs of $\approx_p^{>k}$ -equivalent (resp., $\approx_p^{\geq k}$ -equivalent) assignments $\chi_1, \chi_2 \in \text{Asg}$.*

Example 4.4. *Let χ_1 and χ_2 be two assignments over the singleton $\{p\}$ defined as reported in Figure 4.1. It is clear that $\chi_1 \approx_p^{>3} \chi_2$, but $\chi_1 \not\approx_p^{>4} \chi_2$, and so $\chi_1 \approx_p^{\geq 4} \chi_2$, but $\chi_1 \not\approx_p^{\geq 5} \chi_2$. Also, consider the three responses $F_A, F_R, F_S \in \text{Rsp}(\{p\})$ defined as follows, for all hypertteams $\mathfrak{X} \in \text{Asg}(\{p\})$ and time instants $t \in \mathbb{N}$: $F_A(\chi)[t] \stackrel{\text{def}}{=} \chi(p)[t + 1]$; $F_R(\chi)[t] \stackrel{\text{def}}{=} \overline{\chi(p)[t]}$; $F_S(\chi)[t] \stackrel{\text{def}}{=} \top$, if $t = 0$, and $F_S(\chi)[t] \stackrel{\text{def}}{=} \chi(p)[t - 1]$, otherwise. It is immediate*

	0	1	2	3	4	5	
$\chi_1 = \{ p: $	⊤	⊥	⊥	⊤	⊥	⊤	⋯
$\chi_2 = \{ p: $	⊤	⊥	⊥	⊤	⊤	⊥	⋯
$F_A(\chi_1) = $	⊥	⊥	⊤	⊥	⊤	⊥	⋯
$F_A(\chi_2) = $	⊥	⊥	⊤	⊤	⊥	⊤	⋯
$F_R(\chi_1) = $	⊥	⊤	⊤	⊥	⊤	⊥	⋯
$F_R(\chi_2) = $	⊥	⊤	⊤	⊥	⊥	⊤	⋯
$F_S(\chi_1) = $	⊤	⊤	⊥	⊥	⊤	⊥	⋯
$F_S(\chi_2) = $	⊤	⊤	⊥	⊥	⊤	⊤	⋯

Figure 4.1 – Two $\approx_p^{>3}$ (resp., $\approx_p^{\geq 4}$) -equivalent assignments with one non-realizable (F_A), one realizable (F_R) and one strongly-realizable (F_S) response.

to see that F_R is realizable, while F_S is strongly realizable. However, F_A does not enjoy any realizable property, being defined as a future-dependent response. Indeed, $F_A(\chi_1)(3) \neq F_A(\chi_2)(3)$, even though $\chi_1 \approx_p^{>3} \chi_2$.

To capture in the logic the realizable constraints on the responses, we extend QPTL with additional decorations for the quantifiers that express realizable dependencies among the propositions involved. The result is a new logic, called *Good-for-Games QPTL*, able to express in a natural way game-theoretic concepts of Boolean games.

Definition 4.7 (GFG-QPTL Syntax). *Good-for-Games QPTL (GFG-QPTL) is the set of formulas built according to the following context-free grammar, where $\psi \in LTL$, $p \in AP$, and $P_R, P_S \subseteq AP$:*

$$\varphi := \psi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists^{\Theta} p \varphi \mid \forall^{\Theta} p \varphi; \quad \Theta := \langle P_R, P_S \rangle.$$

A propositional quantifier of the form $Q^{\langle P_R, P_S \rangle} p$, with $Q \in \{\exists, \forall\}$, explicitly expresses a Q -quantification over p , i.e., a choice of a response to interpret p that is also realizable *w.r.t* all the propositions in P_R and strongly-realizable *w.r.t* those in P_S .

To ease the notation, we may write $R : P_R$ and $S : P_S$ for $\langle P_R, \emptyset \rangle$ and $\langle \emptyset, P_S \rangle$, respectively, and R and S instead of $R : AP$ and $S : AP$. We also omit the quantifier specification $\langle \emptyset, \emptyset \rangle$, using $Qp \varphi$ to denote $Q^{\langle \emptyset, \emptyset \rangle} p \varphi$. Observe that the quantifier Qp , which is not restricted, is equivalent to the corresponding QPTL quantifier. Finally, we may drop the curly brackets for the sets P_R and P_S and write $R : p, q$ instead of $R : \{p, q\}$.

We say that a GFG-QPTL formula is *realizable* (resp., *strongly-realizable*) if it is

in prenex form, its quantifier prefix does not contain duplicated variables (i.e., every variable is quantified over at most once), and all its quantifier specifications are equal to R (resp., S). We call *realizable GFG-QPTL* the syntactic fragment of GFG-QPTL that considers realizable GFG-QPTL formulas only. We denote by \mathbf{Q} (resp., \mathbf{Q}_R) the set of (resp., realizable) quantifier prefixes and by \mathbf{O} the set of quantifier specifications.

Given assignments $\chi_1, \chi_2 \in \mathbf{Asg}(P)$, we write $\chi_1 \sim_{\Theta}^k \chi_2$, for some $\Theta = \langle P_R, P_S \rangle \in \mathbf{O}$ and $k \in \mathbb{N}$, if one of the following conditions holds:

1. $\chi_1 = \chi_2$;
2. $\chi_1 \approx_p^{>k} \chi_2$, for some $p \in P_R$;
3. $\chi_1 \approx_p^{\geq k} \chi_2$, for some $p \in P_S$.

We use \approx_{Θ}^k to denote the transitive closure of the reflexive and symmetric relation \sim_{Θ}^k .

Proposition 4.1. *Let $P \subseteq AP$ be a set of atomic propositions, $\chi_1, \chi_2 \in \mathbf{Asg}(P)$ two assignments, $\Theta \in \mathbf{O}$ a quantifier specification, and $k \in \mathbb{N}$ a time instant. Then, $\chi_1 \approx_{\Theta}^k \chi_2$ iff the following hold true:*

1. $\chi_1(q) = \chi_2(q)$, for all $q \in P \setminus (P_R \cup P_S)$;
2. $\chi_1(p)[t] = \chi_2(p)[t]$, for all $t \leq k$ and $p \in (P_R \cap P) \setminus P_S$;
3. $\chi_1(p)[t] = \chi_2(p)[t]$, for all $t < k$ and $p \in P_S \cap P$.

		0	1	2	3	4	5		
$\chi_1 =$	{	$p:$	⊤	⊥	⊥	⊤	⊥	⊤	⋯
		$q:$	⊥	⊥	⊤	⊤	⊥	⊤	⋯
$\chi_2 =$	{	$p:$	⊤	⊥	⊥	⊤	⊤	⊥	⋯
		$q:$	⊥	⊥	⊤	⊤	⊥	⊤	⋯
$\chi_3 =$	{	$p:$	⊤	⊥	⊥	⊤	⊤	⊥	⋯
		$q:$	⊥	⊥	⊤	⊥	⊤	⊤	⋯

Figure 4.2 – Three \approx_{Θ}^3 -equivalent assignments, with $\Theta \stackrel{\text{def}}{=} \langle p, q \rangle$.

Example 4.5. *Consider the three assignments χ_1 , χ_2 , and χ_3 over the doubleton $\{p, q\}$ depicted in Figure 4.2. It is easy to see that $\chi_1 \approx_p^{>3} \chi_2$, as $\chi_1(q) = \chi_2(q)$ and the first position at which the two assignments differ on p is 4; in addition, $\chi_2 \approx_q^{\geq 3} \chi_3$, since $\chi_2(p) = \chi_3(p)$ and the first position at which the two assignments differ on q is 3. Therefore, taking $\Theta \stackrel{\text{def}}{=} \langle p, q \rangle$, we have $\chi_1 \sim_{\Theta}^3 \chi_2 \sim_{\Theta}^3 \chi_3$, which implies $\chi_1 \approx_{\Theta}^3 \chi_3$.*

Given a set of propositions $P \subseteq AP$ and a quantifier specification $\Theta \stackrel{\text{def}}{=} \langle P_R, P_S \rangle \in \mathcal{O}$, we introduce the set of Θ -responses $\text{Rsp}_\Theta(P) \subseteq \text{Rsp}(P)$ containing exactly those $F \in \text{Rsp}(P)$ that are realizable *w.r.t* all the propositions in $P_R \cap P$ and strongly realizable *w.r.t* those in $P_S \cap P$.

Example 4.6. Any $\langle p, q \rangle$ -response F replies to all assignments of Figure 4.2 uniformly, for all time instants between 0 and 3 included. Indeed, $F(\chi_1)[3] = F(\chi_1)[3]$, since $\chi_1 \approx_p^{>3} \chi_2$, being F realizable *w.r.t* p . Similarly, $F(\chi_2)[3] = F(\chi_3)[3]$, since $\chi_2 \approx_q^{\geq 3} \chi_3$, being F strongly-realizable *w.r.t* q . Hence, $F(\chi_1)[3] = F(\chi_3)[3]$.

The following proposition ensures that the above example highlights a general phenomenon.

Proposition 4.2. If $\chi_1 \approx_\Theta^k \chi_2$ then $F(\chi_1)[k] = F(\chi_2)[k]$, for all assignments $\chi_1, \chi_2 \in \text{Asg}(P)$, quantifier specifications $\Theta \in \mathcal{O}$, time instants $k \in \mathbb{N}$, and Θ -responses $F \in \text{Rsp}_\Theta(P)$.

A compositional semantics for GFG-QPTL can be obtained by extending the hyperteam semantics of QPTL reported in Definition 4.8 to account for the possible dependency constraints associated with the quantifiers. To this end, we simply need to parameterise the extension operation for hyperteams with the corresponding specification of the realizable dependencies:

$$\text{EXT}_\Theta(\mathfrak{X}, p) \stackrel{\text{def}}{=} \{\text{EXT}(X, F, p) \mid X \in \mathfrak{X}, F \in \text{Rsp}_\Theta\}.$$

Definition 4.8 (Hyperteam Semantics for GFG-QPTL). *The hyperteam semantics relation $\mathfrak{X} \models^\alpha \varphi$ is inductively defined as in Definition 2.1, for all but Items 3, 7a and 8b that are modified, respectively, as follows, for all propositions $p \in AP$ and quantifier specifications $\Theta \in \mathcal{O}$:*

- 3a') $\mathfrak{X} \models^{\exists^\forall} \psi$ if there is a team $X \in \mathfrak{X}$ such that, for each assignment $\chi \in X$, it holds that $\chi \models_{LTL} \psi$;
- 3b') $\mathfrak{X} \models^{\forall^\exists} \psi$ if, for all teams $X \in \mathfrak{X}$, there is an assignment $\chi \in X$ such that $\chi \models_{LTL} \psi$;
- 7a') $\mathfrak{X} \models^{\exists^\forall} \exists^\Theta p \varphi$ if $\text{EXT}_\Theta(\mathfrak{X}, p) \models^{\exists^\forall} \varphi$;
- 8b') $\mathfrak{X} \models^{\forall^\exists} \forall^\Theta p \varphi$ if $\text{EXT}_\Theta(\mathfrak{X}, p) \models^{\forall^\exists} \varphi$.

Note that one could easily extend both the syntax and semantics of the quantifier specification $\langle P_R, P_S \rangle$ of GFG-QPTL in order to accommodate other types of (in)dependence

constraints, like the ones already studied in first-order logic of incomplete information [HS89; Hod97a; Vää07; MSS11; GV13]. It would suffice to introduce suitable classes of response functions and corresponding constructs, such as the *dependence atoms* of dependence logic, whose semantics can be easily defined via hyperteams.

Of course, we can retrieve results from Chapter 2, such as with the adequacy theorem that instantiates as follows.

Theorem 4.1 (Semantics Adequacy). *For all QPTL formulas φ and hyperteams $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi))$:*

1. $\mathfrak{X} \models^{\exists\forall} \varphi$ iff there exists a set of assignments $X \in \mathfrak{X}$ such that $\chi \models \varphi$, for all assignments $\chi \in X$;
2. $\mathfrak{X} \models^{\forall\exists} \varphi$ iff, for all sets of assignments $X \in \mathfrak{X}$, it holds that $\chi \models \varphi$, for some assignment $\chi \in X$.

4.2.2 Examples

At this point, let us consider some examples to provide insights on the expressive power of the new logic.

Example 4.7. *Let us consider the QPTL sentence $\Phi \stackrel{\text{def}}{=} \forall p(\psi_p \rightarrow \exists q(\psi_q \wedge (q \leftrightarrow Xp)))$, with $\psi_p \stackrel{\text{def}}{=} \neg p \wedge X(\Box p \vee \Box \neg p)$ and $\psi_q \stackrel{\text{def}}{=} \Box q \vee \Box \neg q$. The sentence Φ can be viewed as the description of a very simple game with two players, Abelard and Eloise. Abelard can only choose a truth value for p that will hold constant at any time instant except for time 0, where it is false regardless of his choice, in accordance with ψ_p . Eloise, instead, chooses a truth value for q that will hold constant from time 0 onward, as dictated by ψ_q . The LTL formula $q \leftrightarrow Xp$ encodes the game objective, requiring that the truth value of p at time 1 matches that of q at time 0. Sentence Φ , then, asks whether Eloise can respond with one of her legal moves to every legal move by Abelard so that the objective is always met. If we apply the semantics rule for hyperteams without realizable constraints, so, as in Definition 2.1, we may obtain the following chain of semantic conditions:*

1. $\{\{\emptyset\}\} \models^{\forall\exists} \Phi$;
2. $\{\{\chi_p\}, \{\chi_{\bar{p}}\}, \dots\} \models^{\forall\exists} \psi_p \rightarrow \exists q(\psi_q \wedge (q \leftrightarrow Xp))$;
3. $\{\dots\} \models^{\forall\exists} \neg\psi_p$ and $\{\{\chi_p\}, \{\chi_{\bar{p}}\}\} \models^{\forall\exists} \exists q(\psi_q \wedge (q \leftrightarrow Xp))$;
4. $\{\{\chi_p, \chi_{\bar{p}}\}\} \models^{\exists\forall} \exists q(\psi_q \wedge (q \leftrightarrow Xp))$;

$$5. \{\{\chi_{pq}, \chi_{\bar{p}q}\}, \{\chi_{p\bar{q}}, \chi_{\bar{p}\bar{q}}\}, \dots\} \models^{\exists\forall} \psi_q \wedge (q \leftrightarrow Xp).$$

where Step 3, according to the semantics of disjunction, derives from one of the possible, existentially quantified, partitioning of the hyperteam in Step 2. The steps above go as follows. Being Φ a sentence, it is satisfiable iff Step 1 holds true. By Rule 8b of Definition 2.1 on universal quantifications, we derive Step 2, where $\chi_p \stackrel{\text{def}}{=} \{p \mapsto \perp \top^\omega\}$ and $\chi_{\bar{p}} \stackrel{\text{def}}{=} \{p \mapsto \perp^\omega\}$ are the only two assignments satisfying the precondition ψ_p . The first assignment is obtained by extending \emptyset by means of the constant response function $F_{\perp\top}$ which returns false at time 0 and true at every future instant, i.e., $\chi_p = \text{EXT}(\emptyset, F_{\perp\top}, p)$. Similarly, the second one is obtained by the constant response function F_{\perp} returning false at any time. The assignments obtained by the uncountably many remaining response functions are summarised by the ellipsis. Applying Rule 6b, one can choose to split the hyperteam into the following two parts: $\{\dots\}$ containing all the singleton sets of those assignments violating ψ_p and its complement $\{\{\chi_p\}, \{\chi_{\bar{p}}\}\}$. On the first hyperteam we need to check $\neg\psi_p$, while on the second one the remaining part of the formula, as stated in Step 3. Since $\{\dots\} \models^{\forall\exists} \neg\psi_p$ holds by construction, Rule 7b applied to the second part leads to Step 4, where we use the equality $\{\{\chi_p, \chi_{\bar{p}}\}\} = \overline{\{\{\chi_p\}, \{\chi_{\bar{p}}\}\}}$. Rule 7a on existential quantifications allows, then, to derive Step 5, where $\chi_{\flat q} \stackrel{\text{def}}{=} \chi_{\flat}[q \mapsto \top^\omega]$ and $\chi_{\flat\bar{q}} \stackrel{\text{def}}{=} \chi_{\flat}[q \mapsto \perp^\omega]$, with $\flat \in \{p, \bar{p}\}$. The relevant sets of assignments in the hyperteam at Step 5 are obtained as follows:

1. $\{\chi_{pq}, \chi_{\bar{p}q}\} = \text{EXT}(\{\chi_p, \chi_{\bar{p}}\}, F_{\top}, q)$, where F_{\top} is the constant response function returning true at every time;
2. $\{\chi_{p\bar{q}}, \chi_{\bar{p}\bar{q}}\} = \text{EXT}(\{\chi_p, \chi_{\bar{p}}\}, F_p, q)$, where $F_p(\chi)$ returns at time i the value of p in χ at $i + 1$;
3. $\{\chi_{p\bar{q}}, \chi_{\bar{p}q}\} = \text{EXT}(\{\chi_p, \chi_{\bar{p}}\}, F_{\bar{p}}, q)$, where $F_{\bar{p}}(\chi)$ returns at time i the dual value of p in χ at $i + 1$;
4. $\{\chi_{p\bar{q}}, \chi_{\bar{p}\bar{q}}\} = \text{EXT}(\{\chi_p, \chi_{\bar{p}}\}, F_{\perp}, q)$, where F_{\perp} is the constant response function returning false at every time.

At this point, since $\psi_q \wedge (q \leftrightarrow Xp)$ is an LTL formula, Rule 3 of Definition 2.1 can be applied (by interpreting it as in Definition 4.8), thus asking for a set of assignments containing only assignments that make $\psi_q \wedge (q \leftrightarrow Xp)$ true. Both assignments in the doubleton $\{\chi_{pq}, \chi_{\bar{p}q}\}$ satisfy the LTL formula $\psi_q \wedge (q \leftrightarrow Xp)$, which implies that Φ is satisfiable, witnessing Eloise's win.

Example 4.8. *The simple game in the previous example can equivalently be expressed by the following prenex-form sentence $\Phi' \stackrel{\text{def}}{=} \forall p \exists q (\psi_p \rightarrow (\psi_q \wedge (q \leftrightarrow Xp)))$, where an LTL formula is preceded by a quantifier prefix. The semantic steps here are slightly different and somewhat simpler, since we assume the classic semantics for temporal and Boolean operators within a pure LTL formula. In this case, by applying the semantics, one would obtain the following chain of conditions:*

1. $\{\{\emptyset\}\} \models^{\forall \exists} \Phi'$;
2. $\{\{\chi_p\}, \{\chi_{\bar{p}}\}, \dots\} \models^{\forall \exists} \exists q (\psi_p \rightarrow (\psi_q \wedge (q \leftrightarrow Xp)))$;
3. $\{\{\chi_p, \chi_{\bar{p}}\}, \dots\} \models^{\exists \forall} \exists q (\psi_p \rightarrow (\psi_q \wedge (q \leftrightarrow Xp)))$;
4. $\{\{\chi_{pq}, \chi_{\bar{p}q}, \dots\}, \{\chi_{pq}, \chi_{\bar{p}q}, \dots\}, \{\chi_{p\bar{q}}, \chi_{\bar{p}q}, \dots\}, \{\chi_{p\bar{q}}, \chi_{\bar{p}q}, \dots\}, \dots\} \models^{\exists \forall} \psi_p \rightarrow (\psi_q \wedge (q \leftrightarrow Xp))$.

As in Example 4.7, Φ' is satisfiable iff Step 1 holds true and Step 2 is obtained by applying Rule 8b of Definition 2.1 on universal quantifications, where the ellipsis in the hyperteam is in place of all those singletons of assignments not satisfying ψ_p . Steps 3 and 4 are due to Rules 7b and 7a on existential quantifications. In particular, the innermost ellipses in the hyperteam at Step 4 are again in place of assignments not satisfying ψ_p , while the outermost ellipsis stands for all those sets of assignments not satisfying ψ_q . Finally, it is clear that $\{\chi_{pq}, \chi_{\bar{p}q}, \dots\}$ is the only set of assignments universally satisfying the LTL formula $\psi_p \rightarrow (\psi_q \wedge (q \leftrightarrow Xp))$, as all the other sets have at least one assignment satisfying ψ_p , but falsifying ψ_q or $q \leftrightarrow Xp$.

Example 4.9. *Continuing with the QPTL pnf sentence $\Phi' \stackrel{\text{def}}{=} \forall p \exists q (\psi_p \rightarrow (\psi_q \wedge (q \leftrightarrow Xp)))$ of Example 4.8. Obviously, Eloise cannot win the game described by that sentence following a realizable strategy, as she would need to know at round 0 the opponent's choice for p at round 1. This is clearly reflected in the compositional semantics. Indeed, the response F_p required to obtain the two satisfying assignments χ_{pq} and $\chi_{\bar{p}q}$ is clearly non-realizable. Therefore, Φ' is not realizable (in the sense of behavioral in [PR89]), since the response F_p cannot be implemented by any concrete synchronous transducer. This is confirmed by observing that if we replace the two quantifiers with their realizable counterparts, the resulting GFG-QPTL formula $\Phi_R' \stackrel{\text{def}}{=} \forall^R p \exists^R q (\psi_p \rightarrow (\psi_q \wedge (q \leftrightarrow Xp)))$, is no longer satisfiable. Indeed, the only realizable responses allowing for the satisfaction of ψ_q are F_{\top} and F_{\perp} (that answer always \top and \perp respectively). Therefore, semantic steps analogous to the ones shown in Example 4.8 applied to Φ_R' would lead to*

$\{\{\chi_{pq}, \chi_{\bar{p}q}, \dots\}, \{\chi_{p\bar{q}}, \chi_{\bar{p}\bar{q}}, \dots\}, \dots\} \models^{\exists\forall} \psi_p \rightarrow (\psi_q \wedge (q \leftrightarrow Xp))$, where the sets of assignments are obtained as follows:

- $\{\chi_{pq}, \chi_{\bar{p}q}, \dots\} = \text{EXT}(\{\chi_p, \chi_{\bar{p}}, \dots\}, F_{\top}, q)$;
- $\{\chi_{p\bar{q}}, \chi_{\bar{p}\bar{q}}, \dots\} = \text{EXT}(\{\chi_p, \chi_{\bar{p}}, \dots\}, F_{\perp}, q)$;
- the outer ellipsis $\dots = \{\text{EXT}(\{\chi_p, \chi_{\bar{p}}, \dots\}, F, q) \mid F \in \text{Rsp}_R \setminus \{F_{\top}, F_{\perp}\}\}$ contains all the extensions of $\{\chi_p, \chi_{\bar{p}}, \dots\}$ w.r.t the remaining realizable responses.

Clearly, each set of assignments in the outer ellipsis contains no assignment satisfying ψ_q . Each such set also contains at least one assignment that does not satisfy ψ_p . As a consequence, no set in the outer ellipsis universally satisfies $\psi_p \rightarrow (\psi_q \wedge (q \leftrightarrow Xp))$. Moreover, in the first set of assignments $\{\chi_{pq}, \chi_{\bar{p}q}, \dots\}$, the assignment $\chi_{\bar{p}q}$ satisfies both ψ_p and ψ_q , but not $q \leftrightarrow Xp$. In the second set $\{\chi_{p\bar{q}}, \chi_{\bar{p}\bar{q}}, \dots\}$, instead, the unsatisfying assignment is $\chi_{p\bar{q}}$, for the same reason. This shows that Φ_R' is unsatisfiable.

The previous example shows a satisfiable QPTL sentence whose realizable counterpart becomes unsatisfiable. The opposite may also occur, as the following example illustrates.

$$\begin{array}{l} \chi'_1 = \left\{ \begin{array}{l} q: \begin{array}{cc} 0 & 1 \\ a & b \end{array} \dots \end{array} \right\} \\ \chi'_2 = \left\{ \begin{array}{l} q: \begin{array}{cc} a & \bar{b} \end{array} \dots \end{array} \right\} \end{array} \quad \begin{array}{l} \chi_1 = \left\{ \begin{array}{l} p: \begin{array}{ccc} 0 & 1 & \\ c & * & \dots \end{array} \\ q: \begin{array}{ccc} a & b & \dots \end{array} \end{array} \right\} \\ \chi_2 = \left\{ \begin{array}{l} p: \begin{array}{ccc} c & * & \dots \end{array} \\ q: \begin{array}{ccc} a & \bar{b} & \dots \end{array} \end{array} \right\} \end{array}$$

Figure 4.3 – Two schema assignments, with $a, b, c \in \{\top, \perp\}$, where $\chi'_1, \chi'_2 \in Y$, $\chi_1, \chi_2 \in X \in \text{EXT}_R(\{Y\}, p)$, \bar{b} denotes the dual of b , and $*$ denotes a don't-care value.

Example 4.10. Consider the QPTL sentence $\exists q \forall p \psi$, with $\psi \stackrel{\text{def}}{=} p \leftrightarrow Xq$, which allows for non-realizable responses/strategies. According to the classic Tarskian semantics of QPTL, the sentence is unsatisfiable. In game-theoretic terms, indeed, Abelard can falsify ψ by looking at the value of q one instant in the future and choosing the opposite value as the present value for p . By Theorem 4.1, the sentence is unsatisfiable also under the realizable hyperteam semantics. On the other hand, if we require that the two players only use realizable strategies, things may change. In particular, the two GFG-QPTL sentences $\forall^R p \exists^S q \psi$ and $\exists^R q \forall^R p \psi$ are both satisfiable w.r.t the trivial hyperteam $\{\{\emptyset\}\}$ regardless of the alternation flag, being self-dual. For the first one, it is enough to observe that the strongly-realizable response F_S of Example 4.4 allows to mimic any temporal valuation assigned to the proposition p one instant in the past, as required by the LTL property

ψ . For the second one, we need to show that, $\text{EXT}_R(\{Y\}, p) \models^{\forall\exists} \psi$, with $Y = \text{Asg}(\{q\})$. Now, let $X \in \text{EXT}_R(\{Y\}, p)$ be an arbitrary team obtained by extending those in Y as prescribed by the realizable restriction R associated with the universal quantifier. Also, consider $\chi_1, \chi_2 \in X$ as two of those assignments that differ on q at time 1, but are equal at time 0, i.e., $\chi_1(q)[0] = \chi_2(q)[0]$, but $\chi_1(q)[1] \neq \chi_2(q)[1]$ (see Figure 4.3). Due to the required realizability w.r.t q of the responses used in the extension of Y , we necessarily have that $\chi_1(p)[0] = \chi_2(p)[0]$. As a consequence, either χ_1 or χ_2 satisfies ψ , and thus $\text{EXT}_R(\{Y\}, p) \models^{\forall\exists} \psi$, as required by Item 3') of the semantics. In other words, Abelard cannot realizably falsify the sentence, since he can no longer look at the value of q in the future.

Example 4.11. Information leaks via quantification of unused variables is a well-known phenomenon in IFL [Vää07]. The same occurs in GFG-QPTL. Consider a formula φ where $p, q \in \text{FREE}(\varphi)$, but $s \notin \text{FREE}(\varphi)$. Then, both the equivalences $\forall p \exists s \exists^R q \varphi \equiv \forall p \exists q \varphi$ and $\forall p \exists^R q \varphi \equiv \forall p \exists^R s \exists^R q \varphi$ do hold. However, the equivalence $\forall p \exists q \varphi \equiv \forall p \exists^R q \varphi$ may fail in general. Indeed, an arbitrary response G_q for q in $\forall p \exists q \varphi$ can be simulated in $\forall p \exists s \exists^R q \varphi$ by the responses $F_s = G_q$, for s , and $F_q(\chi) = \chi(s)$, for q . Clearly, F_q , being the identity on s , is realizable. Intuitively, the unused non-realizably-quantified proposition s leaks information about the future of p to q even if the latter is realizably quantified, as it can see the future of p through the value of s at the present time instant.

The following example expands on the connection between GFG-QPTL and GFG-Automata briefly mentioned in the introduction and shows that GFG-QPTL can express the property of being good-for-game for an automaton.

Example 4.12. It is well known that QPTL is able to express any ω -regular language [Sis83]. This can be proved by encoding the existence of an accepting run of an arbitrary nondeterministic Büchi word automaton \mathcal{A} into a formula $\varphi \stackrel{\text{def}}{=} \exists p q_1 \dots \exists q_k \psi$, where $\text{FREE}(\varphi) = \{p_1, \dots, p_n\}$ is the set of propositions needed to encode the alphabet Σ of the recognised language $L(\mathcal{A})$, the k mutually exclusive fresh atomic propositions q_1, \dots, q_k encode the set of states $Q = \{q_1, \dots, q_k\}$ of the automaton, and ψ is the LTL formula encoding the transition function and the Büchi acceptance condition. Formally, $\{X\} \models^{\exists\forall} \varphi$ iff $\mathcal{L}_X \subseteq L(\mathcal{A})$, where \mathcal{L}_X is the set of ω -words whose encodings over p_1, \dots, p_n are the assignments in X . The realizable GFG-QPTL formula $\varphi_R \stackrel{\text{def}}{=} \exists^R q_1 \dots \exists^R q_k \psi$ identifies precisely the sublanguages recognised by \mathcal{A} when the nondeterminism is resolved in a good-for-game manner [HP06], i.e., $\{X\} \models^{\exists\forall} \varphi$ iff

1. $\mathcal{L}_X \subseteq L(\mathcal{A})$,
2. there exists a function $\sigma: \Sigma^* \times Q \rightarrow Q$, choosing a successor state $\sigma(u, q)$ of a state $q \in Q$ based on the prefix $u \in \Sigma^*$ of the input words read up to that moment, and
3. for every ω -word $u \in \mathcal{L}_X$, there exists an accepting run $r \in Q^\omega$ of \mathcal{A} such that $r[i+1] = \sigma(u[:i], r[i])$, for every time instant $i \in \mathbb{N}$.

Intuitively, the function σ is a uniform strategy to resolve the nondeterminism of the automaton and can be clearly modelled by means of realizable responses. As a consequence, the GFG-QPTL sentence $\forall p_1 \dots \forall p_n (\varphi \leftrightarrow \varphi_R)$ is satisfiable iff \mathcal{A} is a good-for-game automaton.

4.3 Game-Theoretic Semantics

The satisfiability problem for the realizable fragment of GFG-QPTL can be solved by showing the existence of a game, played by Eloise and Abelard, with the property that Eloise wins the game iff the corresponding formula is indeed satisfiable. We provide here a general result, showing that, for any realizable quantifier prefix φ and Borelian property Ψ^1 , there exists a game, called *quantification game*, such that Eloise wins the game iff the hyperteam obtained by evaluating the prefix, namely $\text{EXT}_{\exists\forall}(\emptyset, \varphi)$, contains a team completely included in Ψ . The correctness of this result depends, in turn, on the existence of canonical forms for the quantifier prefixes that allow one to reduce the alternations to at most one.

4.3.1 Quantification Game for Sentences

To define the quantification game, we first recall few preliminary notions.

A two-player turn-based *arena* $A = \langle P^\exists, P^\forall, p, M \rangle$ is a tuple where

1. P^\exists, P^\forall are the sets of positions owned by *Eloise* and *Abelard* respectively with $P^\exists \cap P^\forall = \emptyset$; we denote by P the set of all positions in the game, i.e. $P = P^\exists \uplus P^\forall$,
2. $p_I \in P$ is the *initial position*, and

1. By Borelian property we mean an arbitrary team (possibly, but non-necessarily, induced by an LTL formula ψ) corresponding to a set in the Borel hierarchy built upon a suitable Cantor topological space [PP04]; we recall that, starting from the open sets in the space (e.g., eventuality properties, such as, those induced by LTL formulas of the form $\Diamond p$), the hierarchy is uniquely built by applying the operations of countable union, countable intersection, and complementation.

3. $M \subseteq P \times P$ is the binary left-total relation describing all possible *moves*.

A *path* $\pi \in \mathbf{Paths} \subseteq P^\infty$ is a finite or infinite sequence of positions compatible with the move relation, i.e., $(\pi[i], \pi[i + 1]) \in M$, for all $i \in \llbracket 0, \dots, |\pi| - 1 \rrbracket$; it is *initial* if $|\pi| > 0$ and $\pi[0] = p_I$. A *history* for player $\alpha \in \{\exists, \forall\}$ is a finite initial path $h \in \mathbf{Hist}_\alpha \subseteq \mathbf{Paths} \cap (P^* \cdot P^\alpha)$ terminating in an α -position. A *play* $\rho \in \mathbf{Plays} \subseteq \mathbf{Paths}$ is a maximal (i.e., infinite since M is left-total) initial path. A *strategy* for player $\alpha \in \{\exists, \forall\}$ is a function $\sigma_\alpha \in \mathbf{Strat}_\alpha \subseteq \mathbf{Hist}_\alpha \rightarrow P$ mapping each α -history $h \in \mathbf{Hist}_\alpha$ to a position $\sigma_\alpha(h) \in P$ compatible with the move relation, i.e., $(\text{lst}(h), \sigma_\alpha(h)) \in M$. A path $\pi \in \mathbf{Paths}$ is *compatible with a pair of strategies* $(\sigma_\exists, \sigma_\forall) \in \mathbf{Strat}_\exists \times \mathbf{Strat}_\forall$ if, for all $i \in \llbracket 0, \dots, |\pi| - 1 \rrbracket$, it holds that $\pi[i + 1] = \sigma_\exists(\pi[: i])$, if $\pi[i] \in P^\exists$, and $\pi[i + 1] = \sigma_\forall(\pi[: i])$, otherwise. As one may expect, we say that a path is *compatible with a strategy* $\sigma_\exists \in \mathbf{Strat}_\exists$ if it is compatible with the pair of strategies $(\sigma_\exists, \sigma_\forall) \in \mathbf{Strat}_\exists \times \mathbf{Strat}_\forall$, for some strategy $\sigma_\forall \in \mathbf{Strat}_\forall$. The *induced play* of a pair of strategies $(\sigma_\exists, \sigma_\forall) \in \mathbf{Strat}_\exists \times \mathbf{Strat}_\forall$ is the unique play $\pi \in \mathbf{Plays}$ compatible with both σ_\exists and σ_\forall . The *play function* $\text{PLAY}: \mathbf{Strat}_\exists \times \mathbf{Strat}_\forall \rightarrow \mathbf{Plays}$ returns, for each pair of strategies $(\sigma_\exists, \sigma_\forall) \in \mathbf{Strat}_\exists \times \mathbf{Strat}_\forall$, the unique play $\text{PLAY}(\sigma_\exists, \sigma_\forall) \in \mathbf{Plays}$ compatible with them.

A *game* $\mathcal{G} = \langle A, \text{Win} \rangle$ is a tuple, where A is an arena and $\text{Win} \subseteq \mathbf{Plays}$ is the set of *winning* plays for Eloise; the complement $\mathbf{Plays} \setminus \text{Win}$ is *winning* for Abelard. The *observation function* $\text{OBS}: \mathbf{Paths} \rightarrow O^\infty$, with $O \subseteq P$, associates with each path $\pi \in \mathbf{Paths}$ the ordered sequence $w \stackrel{\text{def}}{=} \text{OBS}(\pi) \in O^\infty$ of all observable positions occurring in it. In other words, w is the maximal subsequence of π that contains only positions in O . Formally, there exists a monotone bijection $f: \llbracket 0, \dots, |w| \rrbracket \rightarrow \{j \in \llbracket 0, \dots, |\pi| \rrbracket \mid \pi[j] \in O\}$ satisfying the equality $w[i] = \pi[f(i)]$, for all $i \in \llbracket 0, \dots, |w| \rrbracket$. Eloise (*resp.*, Abelard) *wins* the game if she (*resp.*, he) has a strategy $\sigma_\exists \in \mathbf{Strat}_\exists$ (*resp.*, $\sigma_\forall \in \mathbf{Strat}_\forall$) such that, for all adversary strategies $\sigma_\forall \in \mathbf{Strat}_\forall$ (*resp.*, $\sigma_\exists \in \mathbf{Strat}_\exists$), the corresponding play $\text{PLAY}(\sigma_\exists, \sigma_\forall)$ induces an observation sequence $\text{OBS}(\text{PLAY}(\sigma_\exists, \sigma_\forall))$ belonging (*resp.*, not belonging) to Win . Notice that, even if the winning conditions are defined on a subset of observable positions, here we only consider perfect-information games, since strategies have, instead, full knowledge of the entire set of histories.

Martin's determinacy theorem [Mar75; Mar85] states that all games whose winning condition is a Borel set in the Cantor topological space of infinite words [PP04] are determined, i.e., one of the two players necessarily wins the game. To ensure that the quantification game we are about to define is indeed determined, we require a form of Borelian condition that can be applied to sets of assignments. This determinacy require-

ment is crucial here, since it is tightly connected with the fact that GFG-QPTL does not allow for undetermined formulas. To this end, let $\mathbf{Val} \stackrel{\text{def}}{=} \mathbf{AP} \rightarrow \{\top, \perp\}$ denote the set of Boolean valuations for sets of propositions and $\mathbf{Val}(P) \stackrel{\text{def}}{=} \{\nu \in \mathbf{Val} \mid \text{DOM}(\nu) = P\}$ the set of valuations for propositions in $P \subseteq \mathbf{AP}$. Also, $\#(\nu) \stackrel{\text{def}}{=} |\text{DOM}(\nu)|$. We can now define a bijection between teams over P and languages of infinite words over the alphabet $\mathbf{Val}(P)$. Let $\text{WRD}: \mathbf{Asg}(P) \rightarrow \mathbf{Val}(P)^\omega$ be the *word function* mapping each assignment $\chi \in \mathbf{Asg}(P)$ to the word $u \stackrel{\text{def}}{=} \text{WRD}(\chi) \in \mathbf{Val}(P)^\omega$ defined by $u[t](p) := \chi(p)[t]$, for all $p \in P$ and $t \in \mathbb{N}$. Clearly WRD is a bijection. Now, every property $\Psi \subseteq \mathbf{Asg}(P)$, i.e., every set of assignments, uniquely induces the language of infinite words $\text{WRD}(\Psi) \stackrel{\text{def}}{=} \{\text{WRD}(\chi) \mid \chi \in \Psi\} \subseteq \mathbf{Val}(P)^\omega$ over the alphabet $\mathbf{Val}(P)$. Thus, Ψ is said to be *Borelian* (*resp.*, *regular*) if the language $\text{WRD}(\Psi)$ is a Borel (*resp.*, *regular*) set.

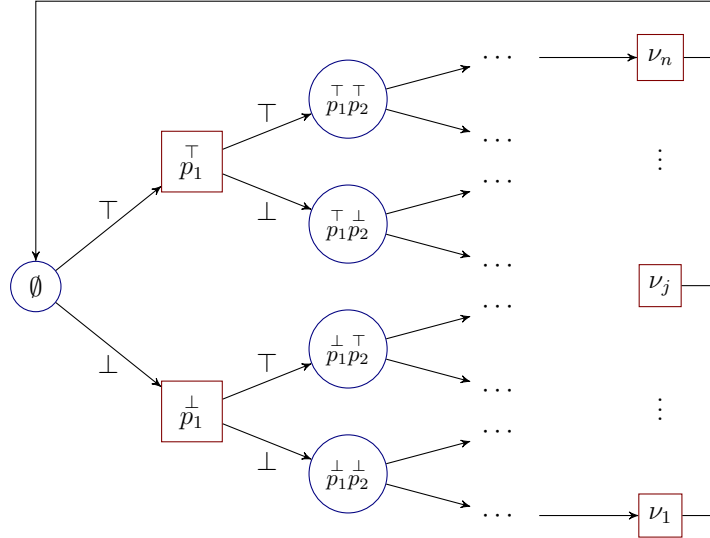


Figure 4.4 – Quantification game for the sentence $\varphi = \exists^R p_1 \forall^R p_2 \exists^R p_3 \dots$. Eloise owns the circled positions, while Abelard the squared ones. From the total-valuation positions ν_1, \dots, ν_n , with $n = 2^{|\mathbf{AP}(\varphi)|}$, Abelard moves to the initial position with empty evaluation.

Given a realizable sentence $\varphi\psi$, let $\text{MOD}(\psi) \subseteq \mathbf{Asg}(\mathbf{AP}(\varphi))$ denote the team satisfying the LTL formula ψ . The quantification game $\mathcal{G}_\varphi^\psi \stackrel{\text{def}}{=} \mathcal{G}_\varphi^{\text{MOD}(\psi)}$ is defined in Construction 4.1 and exemplified in Figure 4.4. Recall that we assume that the prefix φ does not contain duplicates, namely every variable is quantified over at most once in the prefix. The positions of the game are (partial) valuations of the propositions in φ and each position belongs to the player corresponding to the first quantifier in the prefix whose proposition is not defined at that position. The initial position of the game contains the empty val-

uation and in the example of Figure 4.4 belongs to Eloise, since she is the first to play in \wp . Obviously, the game features an infinite number of rounds. Each round begins with the empty valuation and ends in a total valuation, after the players have chosen (jointly) a value for all the propositions. A move in the round corresponds to a player choosing a value for the next proposition in the prefix \wp . Take, for instance, position $\nu \stackrel{\text{def}}{=} \{p_1 \mapsto \perp\}$ in the figure, where the first proposition p_1 has been already assigned value \perp by Eloise. From that position, Abelard first chooses a Boolean value, say \top , for the next proposition p_2 in the prefix. Then he moves to the position $\nu' \stackrel{\text{def}}{=} \{p_1 \mapsto \perp, p_2 \mapsto \top\}$, corresponding to the valuation $\nu[p_2 \mapsto \top]$, obtained by extending ν with the value chosen for p_2 . Position ν' belongs to Eloise, since the next quantifier $\exists p_3$ in the prefix is existential. The last positions belong to Abelard and, from there, he can only move back to the starting position for the next turn. By sampling any infinite sequence of rounds of the games at the positions with total valuations, namely the observable positions, we obtain an infinite word u corresponding to some assignment $\chi \stackrel{\text{def}}{=} \text{WRD}^{-1}(u)$. Then, u is winning for Eloise iff χ belongs to $\Psi \stackrel{\text{def}}{=} \text{MOD}(\psi)$ (i.e., $\chi \models \psi$), while it is winning for Abelard otherwise. This intuition is formalised by the following construction.

Construction 4.1 (Quantification Game I). *For every realizable quantifier prefix $\wp \in \mathbf{Q}_R$ and property $\Psi \subseteq \text{Asg}(AP(\wp))$, the game $\mathcal{G}_\wp^\Psi \stackrel{\text{def}}{=}} \langle A_\wp, O, \text{Win} \rangle$ with arena $A_\wp \stackrel{\text{def}}{=}} \langle P^\exists, P^\forall, p_I, M \rangle$ is defined as prescribed in the following:*

- the set of positions $P \subseteq \mathbf{Val}$ contains exactly those valuations $\nu \in \mathbf{Val}$ of the propositions in $AP(\wp)$ that are quantified in the prefix $(\wp)_{<\#(\nu)}$ of \wp having length $\#(\nu)$ i.e., $\text{DOM}(\nu) = AP((\wp)_{<\#(\nu)})$;
- the set of Eloise’s positions $P^\exists \subseteq P$ only contains the valuations $\nu \in P$ for which the proposition quantified in \wp at index $\#(\nu)$ is existentially quantified, i.e., $(\wp)_{\#(\nu)} = \exists^R p$, for some $p \in AP(\wp)$;
- the initial position $p_I \stackrel{\text{def}}{=} \emptyset$ is just the empty valuation;
- the move relation $M \subseteq P \times P$ contains exactly those pairs of valuations $(\nu_1, \nu_2) \in P \times P$ such that:
 - $\nu_1 \subseteq \nu_2^2$ and $\#(\nu_2) = \#(\nu_1) + 1$, or
 - $\nu_1 \in \mathbf{Val}(AP(\wp))$ and $\nu_2 = \emptyset$;
- the set of observable positions $O \stackrel{\text{def}}{=} \mathbf{Val}(AP(\wp))$ precisely contains the valuations of all the propositions in \wp ;

2. As usual, $\nu_1 \subseteq \nu_2$ denotes the inclusion between functions, i.e., $\text{DOM}(\nu_1) \subseteq \text{DOM}(\nu_2)$ and $\nu_1(p) = \nu_2(p)$, for all $p \in \text{DOM}(\nu_1)$.

- the winning condition induced by the property Ψ is the language of infinite words $\text{Win} \stackrel{\text{def}}{=} \text{WRD}(\Psi)$ over $\text{Val}(\text{AP}(\wp))$.

The game \mathcal{G}_\wp^ψ above essentially provides a game-theoretic version of the semantics of realizable quantifications. The correctness of the game is established by the following theorem.

Theorem 4.2 (Game-Theoretic Semantics (GFG-QPTL) I). *A realizable GFG-QPTL sentence $\wp\psi$, with $\psi \in \text{LTL}$, is satisfiable (resp., unsatisfiable) iff Eloise (resp., Abelard) wins \mathcal{G}_\wp^ψ .*

The proof of this result is split into the following three steps. First, for an arbitrary realizable quantifier prefix \wp , we provide two syntactic transformations, $\vec{\exists}\vec{\forall}(\wp)$ and $\vec{\forall}\vec{\exists}(\wp)$, called *canonicalisations*, which allow one to reduce a realizable GFG-QPTL sentence $\Phi = \wp\psi$ to the sentences $\vec{\exists}\vec{\forall}(\wp)\psi$ and $\vec{\forall}\vec{\exists}(\wp)\psi$ featuring at most a single alternation of quantifiers. Second, in Theorem 4.3, we connect the winner of the game \mathcal{G}_\wp^ψ with the satisfiability of one of the normal forms $\vec{\exists}\vec{\forall}(\wp)\psi$ and $\vec{\forall}\vec{\exists}(\wp)\psi$, showing also that $\vec{\exists}\vec{\forall}(\wp)\psi$ implies $\vec{\forall}\vec{\exists}(\wp)\psi$. Finally, in Theorem 4.4, we prove that the original sentence Φ is equisatisfiable with the two normal forms.

Let us start with the definition of the two prefix canonicalisations based on the following syntactic quantifier-swap operations. Consider, *e.g.*, the formula $\forall^R p \exists^R q \psi$. A naive quantifier-swap operator would simply swap the two quantifiers that, in game-theoretic terms, corresponds to swapping the choices of the two players, which allows Abelard to see Eloise’s move at the current round. To balance this additional power, we restrict the universal quantifier to be strictly realizable, thus preventing Abelard from reading Eloise’s choice. This leads to the formula $\exists^R q \forall^{(\text{AP},q)} p \psi$. A symmetric swap operation would transform the formula $\exists^R q \forall^R p \psi$ into $\forall^R p \exists^{(\text{AP},p)} q \psi$. Essentially, the swap operation exchanges the positions of two adjacent dual realizable quantifiers and restricts the inner one to be strongly realizable *w.r.t* the proposition of the outer one. By iteratively swapping adjacent quantifiers and adjusting the quantifier specification accordingly, we can reduce the quantifier alternation to at most one, still preserving the dependencies in the quantifications at each instant of time.

For technical convenience we use a vector notation for the quantifier prefixes:

$$Q^{(\vec{\theta})}(\vec{p}) \psi \stackrel{\text{def}}{=} Q^{\theta_0} p_0 \cdots Q^{\theta_k} p_k \psi,$$

where $|\vec{p}| = |\vec{\theta}| = k + 1$. We omit the vector symbol in $(\vec{\theta})$ if this is just a sequence of R or S specifications and consider (\vec{p}) as sets of propositions when convenient. We also define in a natural way the union of two quantifier specifications as follows:

$$\langle P_{R1}, P_{S1} \rangle \cup \langle P_{R2}, P_{S2} \rangle \stackrel{\text{def}}{=} \langle P_{R1} \cup P_{R2}, P_{S1} \cup P_{S2} \rangle.$$

Given a realizable quantifier prefix $\wp \in \mathbf{Q}_R$, the two syntactic transformations $\vec{\exists}\vec{\forall}(\cdot)$ and $\vec{\forall}\vec{\exists}(\cdot)$ yield the single-alternation prefixes $\vec{\exists}\vec{\forall}(\wp)$ and $\vec{\forall}\vec{\exists}(\wp)$, by applying all the quantifier swap operations at once. More specifically, the function $\vec{\exists}\vec{\forall}(\cdot)$ provides an $\exists\forall$ -prefix, where all existential quantifiers precede the universal ones, while $\vec{\forall}\vec{\exists}(\cdot)$ gives us the dual $\forall\exists$ -prefix.

For the definition of $\vec{\exists}\vec{\forall}(\cdot)$, we observe that every realizable quantifier prefix \wp can be written in the following form:

$$\wp = \exists^R(\vec{q})_0 (\forall^R(\vec{p})_i \exists^R(\vec{q})_{i=1}^k \forall^R(\vec{p})_{k+1},$$

for some $k \in \mathbb{N}$ and vectors $(\vec{q})_i$, with $i \in \llbracket 0, \dots, k \rrbracket$, and $(\vec{p})_i$, with $i \in \llbracket 1, \dots, k + 1 \rrbracket$, where $|(\vec{q})_i|, |(\vec{p})_i| \geq 1$, for all $i \in \llbracket 1, \dots, k \rrbracket$. For a quantifier prefix \wp we then define

$$\vec{\exists}\vec{\forall}(\wp) \stackrel{\text{def}}{=} (\exists^R(\vec{q})_{i=0}^k (\forall^{\vec{\theta})_i}(\vec{p})_{i=1}^{k+1},$$

where $(\vec{\theta})_i$ is a vector, for every $i \in \llbracket 1, k + 1 \rrbracket$, whose components are defined as $((\vec{\theta})_i)_j \stackrel{\text{def}}{=} \langle \text{AP}, \{(\vec{q})_i \cup \dots \cup (\vec{q})_k\} \rangle$, for all $j \in \llbracket 0, \dots, |(\vec{p})_i| \rrbracket$.

The definition of $\vec{\forall}\vec{\exists}(\cdot)$ is analogous. First, we write a prefix \wp in the form:

$$\wp = \forall^R(\vec{p})_0 (\exists^R(\vec{q})_i \forall^R(\vec{p})_{i=1}^k \exists^R(\vec{q})_{k+1},$$

for some $k \in \mathbb{N}$ and vectors $(\vec{p})_i$, with $i \in \llbracket 0, \dots, k \rrbracket$, and $(\vec{q})_i$, with $i \in \llbracket 1, \dots, k + 1 \rrbracket$, where $|(\vec{p})_i|, |(\vec{q})_i| \geq 1$, for all $i \in \llbracket 1, \dots, k \rrbracket$. Then, we define

$$\vec{\forall}\vec{\exists}(\wp) \stackrel{\text{def}}{=} (\forall^R(\vec{p})_{i=0}^k (\exists^{\vec{\theta})_i}(\vec{q})_{i=1}^{k+1},$$

where $(\vec{\theta})_i$ is a vector, for every $i \in \llbracket 1, k + 1 \rrbracket$, whose components are defined as $((\vec{\theta})_i)_j \stackrel{\text{def}}{=} \langle \text{AP}, \{(\vec{p})_i \cup \dots \cup (\vec{p})_k\} \rangle$, for all $j \in \llbracket 0, \dots, |(\vec{q})_i| \rrbracket$.

Example 4.13. Consider the realizable quantifier prefix $\wp = \forall^R p \exists^R q \exists^R r \forall^R s \exists^R t$. The corresponding $\exists\forall$ canonical-form is $\vec{\exists}\vec{\forall}(\wp) = \exists^R q \exists^R r \exists^R t \forall^{\theta^p} p \forall^{\theta^s} s$, where $\theta^p \stackrel{\text{def}}{=} \langle \text{AP}, \{p\} \rangle$ and $\theta^s \stackrel{\text{def}}{=} \langle \text{AP}, \{s\} \rangle$.

$\langle AP, \{q r t\} \rangle$ and $\theta^s \stackrel{\text{def}}{=} \langle AP, t \rangle$. The $\forall\exists$ canonical-form prefix is, instead, $\vec{\forall\exists}(\varphi) = \forall^R p \forall^R s \exists^\theta q \exists^\theta r \exists^R t$, where $\theta \stackrel{\text{def}}{=} \langle AP, s \rangle$.

For the second part of the proof of Theorem 4.2, we need to connect the winner of \mathcal{G}_φ^ψ with the satisfiability of (one among) $\vec{\exists\forall}(\varphi)\psi$ and $\vec{\forall\exists}(\varphi)\psi$. This also corresponds to showing that $\vec{\exists\forall}(\varphi)\psi \implies \vec{\forall\exists}(\varphi)\psi$. To this end, we exploit the ω -regularity of LTL languages, which ensures that the game is Borelian.

Theorem 4.3 (Quantification Game I). *For each realizable quantification prefix $\varphi \in \mathbf{Q}_R$ and Borelian property $\Psi \subseteq \mathbf{Asg}(AP(\varphi))$, the game \mathcal{G}_φ^Ψ satisfies the following two properties:*

1. *if Eloise wins then $E \subseteq \Psi$, for some $E \in \text{EXT}_{\exists\forall}(\vec{\forall\exists}(\varphi))$;*
2. *if Abelard wins then $E \not\subseteq \Psi$, for all $E \in \text{EXT}_{\exists\forall}(\vec{\exists\forall}(\varphi))$.*

The idea of the proof is to extract from a winning strategy of Eloise (*resp.*, Abelard) a vector \vec{f} of response functions, one for each proposition associated with that player, witnessing the existence (*resp.*, non-existence) of a set E of assignments that satisfies the property Ψ . More precisely, assume Eloise has a strategy σ to win the game and let $\forall^R(\vec{p}) \exists^{\vec{\theta}}(\vec{q}) = \vec{\forall\exists}(\varphi)$ be the $\forall\exists$ canonical-form of φ . Then, thanks to the bijection between plays ρ and assignments χ , we can operate as follows, for every round k and existential proposition q_i in (\vec{q}) : given Abelard's choices up to round k in ρ , we can extract, from Eloise's response for q_i in σ , the response to χ at time k of the response function f_i in \vec{f} . As a consequence, for all $\chi \in \mathbf{Asg}((\vec{p}))$ chosen by Abelard, Eloise's response corresponding to the extension of χ with \vec{f} on (\vec{q}) satisfies, i.e., belongs to, the property Ψ . The witness E is precisely the set of all those extensions. An analogous argument applies to Abelard for the $\exists\forall$ canonical-form. Notice that \vec{f} meets the specification $(\vec{\theta})$ thanks to the alternation of the players prescribed by φ in each round of \mathcal{G}_φ^Ψ . A detailed proof is provided in Section 4.B. The following result is now immediate.

Corollary 4.1 (Quantification Game I). *For every realizable GFG-QPTL sentence $\varphi\psi$, with $\psi \in \text{LTL}$, the game \mathcal{G}_φ^ψ satisfies the following two properties:*

1. *if Eloise (*resp.*, Abelard) wins then $\vec{\forall\exists}(\varphi)\psi$ is satisfiable (*resp.*, $\vec{\exists\forall}(\varphi)\psi$ is unsatisfiable);*
2. *if $\vec{\exists\forall}(\varphi)\psi$ is satisfiable (*resp.*, $\vec{\forall\exists}(\varphi)\psi$ is unsatisfiable) then Eloise (*resp.*, Abelard) wins.*

Proof. Item 1 immediately follows from Item 3') of Definition 4.8, Theorem 2.4 and the two items of Theorem 4.3. For Item 2, instead, let us assume that $\vec{\exists}\forall(\wp)\psi$ is satisfiable (*resp.*, $\vec{\forall}\exists(\wp)\psi$ is unsatisfiable). Thanks to Item 3') of Definition 4.8 and Theorem 2.4, if $\emptyset \models^{\exists\forall} \vec{\exists}\forall(\wp)\psi$ (*resp.*, $\emptyset \not\models^{\exists\forall} \vec{\forall}\exists(\wp)\psi$), then $E \subseteq \Psi \stackrel{\text{def}}{=} \text{MOD}(\psi)$, for some $E \in \text{EXT}_{\exists\forall}(\vec{\exists}\forall(\wp)) = \text{EXT}_{\exists\forall}(\emptyset, \vec{\exists}\forall(\wp))$ (*resp.*, $E \not\subseteq \Psi$, for all $E \in \text{EXT}_{\exists\forall}(\vec{\forall}\exists(\wp)) = \text{EXT}_{\exists\forall}(\emptyset, \vec{\forall}\exists(\wp))$). Thus, by Item 2 (*resp.*, Item 1) of Theorem 4.3, it follows that Abelard (*resp.*, Eloise) loses the game \mathcal{G}_{\wp}^{ψ} , which means, by determinacy, that Eloise (*resp.*, Abelard) wins. Recall that \mathcal{G}_{\wp}^{ψ} is determined, since its winning condition is Borelian [Mar75]. \square

The final step establishes the equisatisfiability of a realizable GFG-QPTL sentence $\wp\psi$ with its two canonical forms $\vec{\exists}\forall(\wp)\psi$ and $\vec{\forall}\exists(\wp)\psi$.

Theorem 4.4 (Sentence Canonical Forms). *For every realizable GFG-QPTL sentence $\wp\psi$, with $\psi \in \text{LTL}$, it holds that $\wp\psi$, $\vec{\exists}\forall(\wp)\psi$, and $\vec{\forall}\exists(\wp)\psi$ are equisatisfiable.*

Towards the proof, we can derive the chain of implications $\vec{\forall}\exists(\wp)\psi \implies \wp\psi \implies \vec{\exists}\forall(\wp)\psi$ by exploiting the following property of the generalized extension function. Specifically, this asserts a total ordering *w.r.t* the preorder \sqsubseteq between a realizable quantifier prefix \wp and its two canonical forms $\vec{\exists}\forall(\wp)$ and $\vec{\forall}\exists(\wp)$, which can be proved by induction on the structure of \wp .

Proposition 4.3. $\text{EXT}_{\alpha}(\mathfrak{X}, \vec{\alpha}(\wp)) \sqsubseteq \text{EXT}_{\alpha}(\mathfrak{X}, \wp) \sqsubseteq \text{EXT}_{\alpha}(\mathfrak{X}, \vec{\alpha}'(\wp))$, for all hyperteams $\mathfrak{X} \in \text{HypTeams}$ and realizable quantifier prefixes $\wp \in \mathbf{Q}_R$, with $\text{AP}(\wp) \cap \text{AP}(\mathfrak{X}) = \emptyset$.

Proof of Theorem 4.4. From Proposition 4.3, Theorem 3.4, and Theorem 2.1, the chain of implications $\vec{\forall}\exists(\wp)\psi \implies \wp\psi \implies \vec{\exists}\forall(\wp)\psi$ easily follows. Indeed, by Theorem 3.4, we have that

1. $\wp\psi$ is satisfiable iff $\text{EXT}_{\exists\forall}(\wp) \models^{\exists\forall} \psi$,
2. $\vec{\exists}\forall(\wp)\psi$ is satisfiable iff $\text{EXT}_{\exists\forall}(\vec{\exists}\forall(\wp)) \models^{\exists\forall} \psi$, and
3. $\vec{\forall}\exists(\wp)\psi$ is satisfiable iff $\text{EXT}_{\exists\forall}(\vec{\forall}\exists(\wp)) \models^{\exists\forall} \psi$.

Now, by Proposition 4.3, it holds that $\text{EXT}_{\exists\forall}(\vec{\forall}\exists(\wp)) \sqsubseteq \text{EXT}_{\exists\forall}(\wp) \sqsubseteq \text{EXT}_{\exists\forall}(\vec{\exists}\forall(\wp))$. Therefore, by Theorem 2.1, we have that if $\vec{\forall}\exists(\wp)\psi$ is satisfiable then $\wp\psi$ is satisfiable too, which, in turn, implies that $\vec{\exists}\forall(\wp)\psi$ is satisfiable as well. To complete the proof, we need to show that, if $\vec{\exists}\forall(\wp)\psi$ is satisfiable, then also $\vec{\forall}\exists(\wp)\psi$ is satisfiable. This fact is, however, a direct consequence of Corollary 4.1. \square

We can finally prove of the main result of this subsection, namely Theorem 4.2.

Proof of Theorem 4.2. We want to prove that $\wp\psi$ is satisfiable (*resp.*, unsatisfiable) iff Eloise (*resp.*, Abelard) wins \mathcal{G}_\wp^ψ . For the *if-direction*, by Item 1 of Corollary 4.1, if Eloise (*resp.*, Abelard) wins the game then $\vec{\forall}\exists(\wp)\psi$ is satisfiable (*resp.*, $\vec{\exists}\forall(\wp)\psi$ is unsatisfiable). However, this implies that $\wp\psi$ is satisfiable (*resp.*, unsatisfiable), thanks to Theorem 4.4. For the *only-if-direction*, if $\wp\psi$ is satisfiable (*resp.*, unsatisfiable) then $\vec{\exists}\forall(\wp)\psi$ is satisfiable (*resp.*, $\vec{\forall}\exists(\wp)\psi$ is unsatisfiable), again due to Theorem 4.4. However, this implies, in turn, that Eloise (*resp.*, Abelard) wins the game, thanks to Item 2 of Corollary 4.1. \square

4.3.2 Quantification Game for Formulas

The game defined in the previous section can easily be adapted to deal with the satisfiability problem for realizable GFG-QPTL as shown in the next section. Solving the model-checking problem requires, however, a generalization of Theorem 4.2, connecting a suitable game with the satisfaction of an arbitrary realizable formula *w.r.t* a hyperteam \mathfrak{X} . We can prove such a property under the assumption that \mathfrak{X} is *well-behaved*, i.e., \mathfrak{X} is the generalized extension of a singleton, composed of a Borelian team X , *w.r.t* some realizable prefix $\tilde{\wp}$. The Borelian requirement is again connected to determinacy of the underlying game. The realizable requirement, instead, allows for a simple proof that leverages the quantification game for sentences directly. At this stage, it is not clear whether the property actually holds for arbitrary Borelian hyperteam. In the model-checking procedure provided later on, however, both properties are satisfied.

To formalize the two assumptions above, we introduce the notion of *generator* for a hyperteam $\mathfrak{X} \in \text{HypTeams}$ as a pair $\langle \tilde{\wp}, X \rangle$ consisting of

1. a realizable quantification prefix $\tilde{\wp} \in \mathbf{Q}_R$ and
2. a Borelian team $\emptyset \neq X \subseteq \text{Asg}(\text{AP}(\mathfrak{X}) \setminus \text{AP}(\tilde{\wp}))$

such that $\mathfrak{X} = \text{EXT}_{\exists\forall}(\{X\}, \tilde{\wp})$. A hyperteam $\mathfrak{X} \in \text{HypTeams}$ is *Borelian realizable* if there is a generator for it.

The new quantification game is defined *w.r.t* a *quantification-game schema* that comprises the input hyperteam \mathfrak{X} , the quantification prefix \wp describing how the players alternate in the game, and the Borelian property Ψ corresponding to the desired goal.

Definition 4.9 (Quantification-Game Schema). *A quantification-game schema is a tuple $\mathcal{S} \stackrel{\text{def}}{=} \langle \mathfrak{X}, \wp, \Psi \rangle$, where*

1. $\mathfrak{X} \in \text{HypTeams}$ is *Borelian realizable*,

2. $\wp \in \mathbf{Q}_R$ is a realizable quantification prefix,
3. $\Psi \subseteq \mathbf{Asg}(AP(\wp) \cup AP(\mathfrak{X}))$ is Borelian, and
4. $AP(\wp) \cap AP(\mathfrak{X}) = \emptyset$.

The idea behind the game-theoretic construction reported below is quite simple. Given a generator $\langle \tilde{\wp}, X \rangle$ for a well-behaved hyperteam \mathfrak{X} , we force the two players to simulate the given \mathfrak{X} by playing according to the prefix $\tilde{\wp}$, once Abelard has arbitrarily chosen the values of the atomic propositions (\vec{p}) over which the team X is defined. Since $\text{EXT}_{\exists\forall}(\forall(\vec{p})) = \{\mathbf{Asg}((\vec{p}))\}$ and $X \subseteq \mathbf{Asg}((\vec{p}))$, it is clear that $\text{EXT}_{\exists\forall}(\forall(\vec{p})) \sqsubseteq \{X\}$ and, by the monotonicity property stated in Lemma 3.4, that extends naturally to GFG-QPTL, we have that $\text{EXT}_{\exists\forall}(\forall(\vec{p}) \tilde{\wp}) \sqsubseteq \mathfrak{X} = \text{EXT}_{\exists\forall}(\{X\}, \tilde{\wp})$. Thus, if Eloise wins the game, she can ensure a given temporal property, i.e., $\mathfrak{X} \models^{\exists\forall} \wp\psi$. Notice, however, that we gave Abelard the freedom to cheat and choose arbitrary values for (\vec{p}). Thus, in principle, Eloise could be able to satisfy the property while losing the game, since Abelard can choose assignments over (\vec{p}) that do not belong to X . To remedy this, we add all those assignments to Eloise's winning set, thus deterring Abelard from cheating.

Construction 4.2 (Quantification Game II). *For a quantification-game schema $\mathcal{S} \stackrel{\text{def}}{=} \langle \mathfrak{X}, \wp, \Psi \rangle$, we say that \mathcal{G} is a \mathcal{S} -game if there is a generator $\langle \tilde{\wp}, X \rangle$ for \mathfrak{X} such that $\mathcal{G} \stackrel{\text{def}}{=} \mathcal{G}_{\tilde{\wp}}^{\hat{\Psi}}$, where*

- $\hat{\wp} \stackrel{\text{def}}{=} \forall(\vec{p}) \tilde{\wp} \wp$ and
- $\hat{\Psi} \stackrel{\text{def}}{=} \Psi \cup \{\chi \in \mathbf{Asg}(P) \mid \chi|_{(\vec{p})} \notin X\}$,

with $(\vec{p}) \stackrel{\text{def}}{=} AP(\mathfrak{X}) \setminus AP(\tilde{\wp})$ and $P \stackrel{\text{def}}{=} AP(\wp) \cup AP(\mathfrak{X})$.

The quantification-game schema for a formula $\wp\psi$, with $\psi \in \text{LTL}$, and a hyperteam \mathfrak{X} is the tuple $\mathcal{S}_{\wp\psi}^{\mathfrak{X}} \stackrel{\text{def}}{=} \langle \mathfrak{X}, \wp, \text{MOD}(\psi) \rangle$. We can now generalize Theorem 4.2 to formulas.

Theorem 4.5 (Game-Theoretic Semantics II). *$\mathfrak{X} \models^{\exists\forall} \wp\psi$ (resp., $\mathfrak{X} \not\models^{\exists\forall} \wp\psi$) iff Eloise (resp., Abelard) wins every $\mathcal{S}_{\wp\psi}^{\mathfrak{X}}$ -game, for all realizable GFG-QPTL formulas $\wp\psi$, with $\psi \in \text{LTL}$, and Borelian realizable hyperteam $\mathfrak{X} \in \text{HypTeams}(\text{FREE}(\wp\psi))$.*

The proof of the above result follows an approach similar to the one described in the previous subsection for Theorem 4.2 and uses the following result, proven in Section 4.B, which generalizes Theorem 4.3 to formulas.

Theorem 4.6 (Quantification Game II). *Every \mathcal{S} -game \mathcal{G} , for some quantification-game schema $\mathcal{S} \stackrel{\text{def}}{=} \langle \mathfrak{X}, \wp, \Psi \rangle$, satisfies the following two properties:*

1. if Eloise wins then $E \subseteq \Psi$, for some $E \in \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\forall\exists}(\varphi))$;
2. if Abelard wins then $E \not\subseteq \Psi$, for all $E \in \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\exists\forall}(\varphi))$.

The connection between the quantification game and the satisfaction problem *w.r.t* a hyperteam is stated by the following result.

Corollary 4.2 (Quantification Game II). *For every realizable GFG-QPTL formula $\varphi\psi$, with $\psi \in \text{LTL}$, and Borelian realizable hyperteam $\mathfrak{X} \in \text{HypTeams}(\text{FREE}(\varphi\psi))$, every $\mathcal{S}_{\varphi\psi}^{\mathfrak{X}}$ -game satisfies the following two properties:*

1. if Eloise (resp., Abelard) wins then $\mathfrak{X} \models^{\exists\forall} \vec{\forall\exists}(\varphi)\psi$ (resp., $\mathfrak{X} \not\models^{\exists\forall} \vec{\exists\forall}(\varphi)\psi$);
2. if $\mathfrak{X} \models^{\exists\forall} \vec{\exists\forall}(\varphi)\psi$ (resp., $\mathfrak{X} \not\models^{\exists\forall} \vec{\forall\exists}(\varphi)\psi$) then Eloise (resp., Abelard) wins.

Proof. Let \mathcal{G} be an arbitrary $\mathcal{S}_{\varphi\psi}^{\mathfrak{X}}$ -game. Item 1 immediately follows from Item 3') of Definition 4.8, Theorem 3.4 and the two items of Theorem 4.6. For Item 2, instead, let us assume that $\mathfrak{X} \models^{\exists\forall} \vec{\exists\forall}(\varphi)\psi$ (resp., $\mathfrak{X} \not\models^{\exists\forall} \vec{\forall\exists}(\varphi)\psi$). Thanks to Item 3') of Definition 4.8 and Theorem 3.4, it holds that $E \subseteq \Psi \stackrel{\text{def}}{=} \text{MOD}(\psi)$, for some $E \in \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\exists\forall}(\varphi))$ (resp., $E \not\subseteq \Psi$, for all $E \in \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\forall\exists}(\varphi))$). Thus, by Item 2 (resp., Item 1) of Theorem 4.6, it follows that Abelard (resp., Eloise) loses the game \mathcal{G} , which means, by determinacy, that Eloise (resp., Abelard) wins. \square

Corollary 4.2, together with Proposition 4.3, lifts Theorem 4.4 to formulas as follows.

Theorem 4.7 (Formula Canonical Forms). *For every realizable GFG-QPTL formula $\varphi\psi$, with $\psi \in \text{LTL}$, it holds that $\mathfrak{X} \models^{\alpha} \varphi\psi$ iff $\mathfrak{X} \models^{\alpha} \vec{\exists\forall}(\varphi)\psi$ iff $\mathfrak{X} \models^{\alpha} \vec{\forall\exists}(\varphi)\psi$, for all Borelian realizable hyperteam $\mathfrak{X} \in \text{HypTeams}(\text{FREE}(\varphi\psi))$.*

Proof. We focus on the statement for $\alpha = \exists\forall$, as the case $\alpha = \forall\exists$ can be easily derived from the previous one by observing that, thanks to the Boolean laws of Theorem 2.3,

1. $\mathfrak{X} \models^{\forall\exists} \varphi\psi$ iff $\mathfrak{X} \not\models^{\exists\forall} \overline{\varphi}\neg\psi$,
2. $\mathfrak{X} \models^{\exists\forall} \vec{\exists\forall}(\varphi)\psi$ iff $\mathfrak{X} \not\models^{\forall\exists} \vec{\forall\exists}(\overline{\varphi})\neg\psi$, and
3. $\mathfrak{X} \models^{\forall\exists} \vec{\forall\exists}(\varphi)\psi$ iff $\mathfrak{X} \not\models^{\exists\forall} \vec{\exists\forall}(\overline{\varphi})\neg\psi$.

As done in the proof of Theorem 4.4, one chain of implication – if $\mathfrak{X} \models^{\exists\forall} \vec{\forall\exists}(\varphi)\psi$ then $\mathfrak{X} \models^{\exists\forall} \varphi\psi$ and if $\mathfrak{X} \models^{\exists\forall} \varphi\psi$ then $\mathfrak{X} \models^{\exists\forall} \vec{\exists\forall}(\varphi)\psi$ – is an immediate consequence of Proposition 4.3, Theorem 3.4, and Theorem 2.1. Indeed, by Theorem 3.4, we have that

1. $\mathfrak{X} \models^{\exists\forall} \varphi\psi$ iff $\text{EXT}_{\exists\forall}(\mathfrak{X}, \varphi) \models^{\exists\forall} \psi$,

2. $\mathfrak{X} \models^{\exists\forall} \overrightarrow{\exists\forall}(\varphi)\psi$ iff $\text{EXT}_{\exists\forall}(\mathfrak{X}, \overrightarrow{\exists\forall}(\varphi)) \models^{\exists\forall} \psi$, and
3. $\mathfrak{X} \models^{\exists\forall} \overrightarrow{\forall\exists}(\varphi)\psi$ iff $\text{EXT}_{\exists\forall}(\mathfrak{X}, \overrightarrow{\forall\exists}(\varphi)) \models^{\exists\forall} \psi$.

Now, by Proposition 4.3, it holds that $\text{EXT}_{\exists\forall}(\mathfrak{X}, \overrightarrow{\forall\exists}(\varphi)) \sqsubseteq \text{EXT}_{\exists\forall}(\mathfrak{X}, \varphi) \sqsubseteq \text{EXT}_{\exists\forall}(\mathfrak{X}, \overrightarrow{\exists\forall}(\varphi))$. Therefore, by Theorem 2.1, we have that $\mathfrak{X} \models^{\exists\forall} \overrightarrow{\forall\exists}(\varphi)\psi$ implies $\mathfrak{X} \models^{\exists\forall} \varphi\psi$, which, in turn, implies $\mathfrak{X} \models^{\exists\forall} \overrightarrow{\exists\forall}(\varphi)\psi$. The converse implication – if $\mathfrak{X} \models^{\exists\forall} \overrightarrow{\exists\forall}(\varphi)\psi$ then $\mathfrak{X} \models^{\exists\forall} \overrightarrow{\forall\exists}(\varphi)\psi$ – is a direct consequence of Corollary 4.2. \square

The previous theorem allows us to obtain a proof for Theorem 4.5.

Proof of Theorem 4.5. Given an arbitrary $\mathcal{S}_{\varphi\psi}^{\mathfrak{X}}$ -game \mathcal{G} , we want to prove that $\mathfrak{X} \models^{\exists\forall} \varphi\psi$ (*resp.*, $\mathfrak{X} \not\models^{\exists\forall} \varphi\psi$) holds true iff Eloise (*resp.*, Abelard) wins \mathcal{G} . For the *if-direction*, by Item 1 of Corollary 4.2, if Eloise (*resp.*, Abelard) wins \mathcal{G} then $\mathfrak{X} \models^{\exists\forall} \overrightarrow{\forall\exists}(\varphi)\psi$ (*resp.*, $\mathfrak{X} \not\models^{\exists\forall} \overrightarrow{\forall\exists}(\varphi)\psi$). However, this implies that $\mathfrak{X} \models^{\exists\forall} \varphi\psi$ (*resp.*, $\mathfrak{X} \not\models^{\exists\forall} \varphi\psi$), thanks to Theorem 4.7. For the *only-if-direction*, if $\mathfrak{X} \models^{\exists\forall} \varphi\psi$ (*resp.*, $\mathfrak{X} \not\models^{\exists\forall} \varphi\psi$) then $\mathfrak{X} \models^{\exists\forall} \overrightarrow{\exists\forall}(\varphi)\psi$ (*resp.*, $\mathfrak{X} \not\models^{\exists\forall} \overrightarrow{\exists\forall}(\varphi)\psi$) holds true, again due to Theorem 4.7. This implies, in turn, that Eloise (*resp.*, Abelard) wins \mathcal{G} , thanks to Item 2 of Corollary 4.2. \square

4.3.3 Complexity Results

The first step in deciding the satisfiability problem is to derive from a realizable sentence $\Phi = Q\psi$ a *parity game* [Mos91; EJ91] that is won by Eloise iff Φ is satisfiable. To do that, we first construct a deterministic parity automaton \mathcal{A}_ψ for the LTL formula ψ , by combining the Vardi-Wolper theorem [VW86a] with the Safra-like translation from Büchi to parity acceptance condition [Pit06]. We then compute the synchronous product of the arena A_Q of Construction 4.1 with \mathcal{A}_ψ , where the automaton component changes state only when Abelard takes a move starting from an observable position containing full valuation of the propositions. Such valuation is read by the transition function of \mathcal{A}_ψ to determine its successor state. The resulting game simulates both the quantification game and the automaton, so that Eloise wins iff the play satisfies ψ . This result, formally stated below, is proven in Section 4.B.

Theorem 4.8 (Satisfiability Game). *For every realizable GFG-QPTL sentence Φ there is a parity game, with $2^{2^{|\Phi|}}$ positions and $2^{|\Phi|}$ priorities, won by Eloise iff Φ is satisfiable.*

We can then obtain an upper bound on the complexity of the problem from the fact that parity games can be solved in time polynomial in the number of positions and

exponential in that of the priorities [EJ88; EJS93; Zie98]. For the lower bound, instead, we observe that the *reactive synthesis problem* [PR89] of an LTL formula ψ can be reduced to the satisfiability of a sentence of the form $\forall^R(\vec{p}) \exists^R(\vec{q}) \psi$, where (\vec{p}) and (\vec{q}) denote, respectively, the input and output signals of the desired system.

Theorem 4.9 (Satisfiability Complexity). *The satisfiability problem for realizable GFG-QPTL sentences is 2-EXPTIME-complete.*

As to the (*universal*) *model-checking problem*, given a Kripke structure \mathbf{K} , we ask whether $\mathbf{K} \models \Phi$, in the sense that $\mathfrak{X}_{\mathbf{K}} \models^{\exists\forall} \Phi$ holds, where $\mathfrak{X}_{\mathbf{K}} \stackrel{\text{def}}{=} \{\{\chi \in \text{Asg}(\text{AP}(\mathbf{K})) \mid \text{WRD}(\chi) \in \text{MOD}(\mathbf{K})\}\}$ is the hyperteam obtained by collecting all the assignments $\chi \in \text{Asg}(\text{AP}(\mathbf{K}))$ over the propositions of \mathbf{K} for which the infinite word $\text{WRD}(\chi)$ belongs to the ω -language $L(\mathbf{K})$ generated by \mathbf{K} . Since $L(\mathbf{K})$ is an ω -regular language, $\mathfrak{X}_{\mathbf{K}}$ is clearly a Borelian realizable hyperteam. As a consequence, Construction 4.2 applies. Thus, we can adopt the same synchronous product described above between the arena of the game and the union of the two automata \mathcal{A}_{ψ} and $\mathcal{A}_{\overline{\mathbf{K}}}$, where \mathcal{A}_{ψ} is obtained from the formula ψ , while $\mathcal{A}_{\overline{\mathbf{K}}}$ is a co-safety automaton of size linear in $|\mathbf{K}|$, recognising the complement of $L(\mathbf{K})$. Observe that one may also consider the dual notion of *existential model-checking*, which asks whether $\mathbf{K} \models \Phi$ in the sense of $\mathfrak{X}_{\mathbf{K}} \models^{\forall\exists} \Phi$, which can be solved analogously.

Theorem 4.10 (Model-Checking Game). *For every Kripke structure \mathbf{K} and realizable GFG-QPTL formula Φ , with $\text{FREE}(\Phi) \subseteq \text{AP}(\mathbf{K})$, there is a parity game, with $2^{2^{O|\Phi|}} \cdot |\mathbf{K}|$ positions and $2^{O|\Phi|}$ priorities, won by Eloise iff $\mathbf{K} \models \Phi$.*

Upper bounds *w.r.t* both formula and model complexity, and the lower bound *w.r.t* formula complexity, are proved as in the case of the satisfiability problem. As far as the model complexity is concerned, the lower bound can be naturally derived by reducing from reachability games [Imm81].

Theorem 4.11 (Model-Checking Complexity). *The model-checking problem for realizable GFG-QPTL has a 2-EXPTIME-complete formula complexity and a PTIME-complete model complexity.*

4.4 Conclusion

We have introduced a novel semantics for QPTL extending in a non-trivial way Hodges' team semantics for Hintikka and Sandu's logic of imperfect information IFL.

On the one hand, the new semantic setting can express games with both symmetric and asymmetric restrictions on the players. On the other hand, it allows for encoding realizable constraints on the quantified propositions, connecting the underlying logic with the game-theoretic notion of realizable strategies. Based on this semantics, the extension of QPTL with constraints on the functional dependencies among propositions, called GFG-QPTL, has surprisingly interesting properties. For one, its realizable fragment enables reducing the solution of two-player zero-sum games to the decision problems for the logic. Indeed, the deep connection with realizable strategies ensures that satisfiable formulae of the logic express linear time properties that can always be realised by means of actual strategies. This fragment also enjoys good computational properties, being 2-EXPTIME-*complete* for both satisfiability and model-checking. It is also very expressive, being equivalent to, though less succinct than, QPTL, hence able to describe all ω -regular properties. Second, the realizable semantics also bears a connection to good-for-game automata, allowing to naturally express the property of being a GFG-QPTL automata, the significance of which is probably worth investigating further.

To the best of our knowledge, this is the first attempt to provide a compositional account of realizable constraints. We believe the generality and flexibility of the semantic settings opens up the possibility of a systematic investigation of the impact of this type of constraints in quantified temporal logics, such as QCTL [Fre01; LM14], Substructure Temporal Logic [BMM13; BMM15], HyperLTL/CTL* [CFKMRS14; FRS15; FH16; FZ16; CFHH19], Coordination Logic [FS10].

In the next chapter, we investigate how to adapt hyperteam semantics for strategic reasoning in a fashion comparable to Strategy Logic [CHP10; MMPV14; MMPV17].

4.A Proofs of Section 4.2

Proposition 4.1. *Let $P \subseteq AP$ be a set of atomic propositions, $\chi_1, \chi_2 \in \text{Asg}(P)$ two assignments, $\Theta \in \mathcal{O}$ a quantifier specification, and $k \in \mathbb{N}$ a time instant. Then, $\chi_1 \approx_{\Theta}^k \chi_2$ iff the following hold true:*

1. $\chi_1(q) = \chi_2(q)$, for all $q \in P \setminus (P_R \cup P_S)$;
2. $\chi_1(p)[t] = \chi_2(p)[t]$, for all $t \leq k$ and $p \in (P_R \cap P) \setminus P_S$;
3. $\chi_1(p)[t] = \chi_2(p)[t]$, for all $t < k$ and $p \in P_S \cap P$.

Proof. Assume $\chi_1 \approx_{\Theta}^k \chi_2$. Because \approx_{Θ}^k is the transitive closure of \sim_{Θ}^k , we have $\chi_1 =$

$\chi^{(1)} \sim_{\Theta}^k \chi^{(2)} \sim_{\Theta}^k \dots \sim_{\Theta}^k \chi^{(r)} = \chi_2$, for some $\chi^{(1)}, \dots, \chi^{(r)}$, with $r \in \mathbb{N} \setminus \{0\}$ (observe that $\chi_1 = \chi_2$ if $r = 1$).

We prove, by induction on r , that items 1–3 hold. If $r = 1$, then the claim follows trivially. Let $r > 1$. Since $\chi^{(1)} \sim_{\Theta}^k \chi^{(2)}$, we have that 1–3 hold when instantiated with $\chi^{(1)}$ and $\chi^{(2)}$, by Definition 4.5. Moreover, by inductive hypothesis, 1–3 hold when instantiated with $\chi^{(2)}$ and $\chi^{(r)}$. The claim follows by transitivity of 1–3.

Now, in order to prove the converse direction, assume that items 1–3 hold. Let $\{p_1, \dots, p_r\}$ be an enumeration of $P_R \cup P_S$ and define $\chi^{(1)} \stackrel{\text{def}}{=} \chi_1$ and $\chi^{(i+1)} \stackrel{\text{def}}{=} \chi^{(i)}[p_i \mapsto \chi_2(p_i)]$ for $i \in [1, \dots, r]$. It is not difficult to convince oneself that $\chi_1 = \chi^{(1)} \sim_{\Theta}^k \chi^{(2)} \sim_{\Theta}^k \dots \sim_{\Theta}^k \chi^{(r+1)} = \chi_2$ holds, hence $\chi_1 \approx_{\Theta}^k \chi_2$. \square

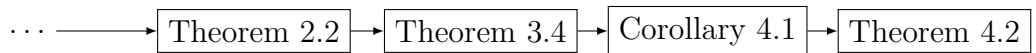
Proposition 4.2. *If $\chi_1 \approx_{\Theta}^k \chi_2$ then $F(\chi_1)[k] = F(\chi_2)[k]$, for all assignments $\chi_1, \chi_2 \in \text{Asg}(P)$, quantifier specifications $\Theta \in \mathcal{O}$, time instants $k \in \mathbb{N}$, and Θ -responses $F \in \text{Rsp}_{\Theta}(P)$.*

Proof. Assume $\chi_1 \approx_{\Theta}^k \chi_2$, i.e., $\chi_1 = \chi^{(1)} \sim_{\Theta}^k \chi^{(2)} \sim_{\Theta}^k \dots \sim_{\Theta}^k \chi^{(r)} = \chi_2$, for some $\chi^{(1)}, \dots, \chi^{(r)}$, with $r \in \mathbb{N} \setminus \{0\}$ (observe that $\chi_1 = \chi_2$ if $r = 1$).

We prove, by induction on r , that $F(\chi_1)(k) = F(\chi_2)(k)$. If $r = 1$, then the claim follows trivially. Let $r > 1$. Since $\chi^{(1)} \sim_{\Theta}^k \chi^{(2)}$ and $F \in \text{Rsp}_{\Theta}(P)$, we have that $F(\chi^{(1)})(k) = F(\chi^{(2)})(k)$. Moreover, by inductive hypothesis, $F(\chi^{(2)})(k) = F(\chi^{(r)})(k)$. The claim follows by transitivity. \square

4.B Proofs of Section 4.3

Now, we showcase the graph of dependency for Theorem 4.2, presenting the lemma, corollary and theorem used for the proof in the main paper.



In order to provide the missing proofs of Theorems 4.3 and 4.6 and Proposition 4.3, in this appendix we shall also need to prove the auxiliary Propositions 4.4 to 4.9 and 4.12, and 4.13 and to introduce, later on, the notion of *normal evolution function* and a refinement of the order between hyperteams.

Proposition 4.4. *Let $\mathfrak{X} \in \text{HypTeams}(P)$ be a hyperteam over $P \subseteq s$, $\Theta \in \mathcal{O}$ a quantifier specification, $p \in s \setminus P$ an atomic proposition, and $\Psi \subseteq \text{Asg}(P \cup \{p\})$ a set of assignments.*

There exists a set of assignments $W \in \text{EXT}_\alpha(\mathfrak{X}, Q^\ominus p)$ such that $W \subseteq \Psi$ iff the following conditions hold true:

1. there exist $F \in \text{Rsp}_\ominus(P)$ and $X \in \mathfrak{X}$ such that $\text{EXT}(X, F, p) \subseteq \Psi$, whenever α and Q are coherent;
2. for all $F \in \text{Rsp}_\ominus(P)$, there is $X \in \mathfrak{X}$ such that $\text{EXT}(X, F, p) \subseteq \Psi$, whenever α and Q are not coherent.

Proof. We consider the two conditions separately.

- [1] If α and Q are coherent, by definition of evolution function, we have

$$\text{EXT}_\alpha(\mathfrak{X}, Q^\ominus p) = \text{EXT}_\ominus(\mathfrak{X}, p) = \{\text{EXT}(X, F, p) \mid X \in \mathfrak{X}, F \in \text{Rsp}_\ominus(\text{AP}(\mathfrak{X}))\}.$$

Thus, for every set of assignments $W \subseteq \text{Asg}(P \cup \{p\})$, it holds that $W \in \text{EXT}_\alpha(\mathfrak{X}, Q^\ominus p)$ iff there exists a \ominus -response $F \in \text{Rsp}_\ominus(P)$ and a set of assignments $X \in \mathfrak{X}$ such that $W = \text{EXT}(X, F, p)$. Hence, Condition 1 immediately follows.

- [2] If α and Q are not coherent, by definition of evolution function, we have

$$\text{EXT}_\alpha(\mathfrak{X}, Q^\ominus p) = \overline{\text{EXT}_\ominus(\overline{\mathfrak{X}}, p)} = \{\text{img}(\Gamma) \mid \Gamma \in \text{Chc}(\text{EXT}_\ominus(\overline{\mathfrak{X}}, p))\}.$$

Thus, for every set of assignments $W \subseteq \text{Asg}(P \cup \{p\})$, it holds that $W \in \text{EXT}_\alpha(\mathfrak{X}, Q^\ominus p)$ iff there exists a choice function $\Gamma \in \text{Chc}(\text{EXT}_\ominus(\overline{\mathfrak{X}}, p))$ such that $W = \text{img}(\Gamma) = \{\Gamma(Z) \mid Z \in \text{EXT}_\ominus(\overline{\mathfrak{X}}, p)\}$. This means that $W \subseteq \Psi$ iff $\Gamma(Z) \in \Psi$, for all $Z \in \text{EXT}_\ominus(\overline{\mathfrak{X}}, p)$. Now, it is clear that there exists a choice function $\Gamma \in \text{Chc}(\text{EXT}_\ominus(\overline{\mathfrak{X}}, p))$ such that $\Gamma(Z) \in \Psi$, for all $Z \in \text{EXT}_\ominus(\overline{\mathfrak{X}}, p)$ iff, for every $Z \in \text{EXT}_\ominus(\overline{\mathfrak{X}}, p) = \{\text{EXT}(Y, F, p) \mid Y \in \overline{\mathfrak{X}}, F \in \text{Rsp}_\ominus(P)\}$, there exists $\chi_Z \in Z$ such that $\chi_Z \in \Psi$. The latter property, however, means that, for every $F \in \text{Rsp}_\ominus(P)$ and $Y \in \overline{\mathfrak{X}} = \{\text{img}(\Lambda) \mid \Lambda \in \text{Chc}(\mathfrak{X})\}$, there exists $\chi_{F,Y} \in \text{EXT}(Y, F, p)$ such that $\chi_{F,Y} \in \Psi$, which in turn can be written as, for every $F \in \text{Rsp}_\ominus(P)$ and $\Lambda \in \text{Chc}(\mathfrak{X})$, there exists $\chi_{F,\Lambda} \in \text{EXT}(\text{img}(\Lambda), F, p) = \text{EXT}(\{\Lambda(X) \mid X \in \mathfrak{X}\}, F, p)$ such that $\chi_{F,\Lambda} \in \Psi$. Now, notice that $\chi_{F,\Lambda} \in \text{EXT}(\{\Lambda(X) \mid X \in \mathfrak{X}\}, F, p)$ iff there exists $X \in \mathfrak{X}$ such that $\chi_{F,\Lambda} = \text{EXT}(\Lambda(X), F, p)$.

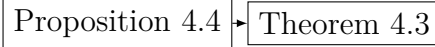
Thus, up to this point, we have shown that the following two properties are equivalent:

- there exists $W \in \text{EXT}_\alpha(\mathfrak{X}, Q^\ominus p)$ such that $W \subseteq \Psi$;
- for all $F \in \text{Rsp}_\ominus(P)$ and $\Lambda \in \text{Chc}(\mathfrak{X})$, there exists $X \in \mathfrak{X}$ such that

$$\text{EXT}(\Lambda(X), F, p) \in \Psi.$$

Now, by deHerbrandizing³ the universal quantification of Λ *w.r.t* the existential quantification of X in the last item and recalling that $\Lambda(X) \in X$, we obtain that, for all $F \in \text{Rsp}_\Theta(P)$, there exists $X \in \mathfrak{X}$ such that $\text{EXT}(\chi, F, p) \in \Psi$, for all $\chi \in X$. But this means that, for all $F \in \text{Rsp}_\Theta(P)$, there exists $X \in \mathfrak{X}$ such that $\text{EXT}(X, F, p) \subseteq \Psi$, as required by Condition 2. \square

Next, we prove Theorem 4.3. Here is the graph of dependency presenting the proposition used for the proof in the main paper.



Theorem 4.3 (Quantification Game I). *For each realizable quantification prefix $\wp \in \mathbf{Q}_R$ and Borelian property $\Psi \subseteq \text{Asg}(AP(\wp))$, the game \mathcal{G}_\wp^Ψ satisfies the following two properties:*

1. *if Eloise wins then $E \subseteq \Psi$, for some $E \in \text{EXT}_{\exists\forall}(\vec{\forall}\vec{\exists}(\wp))$;*
2. *if Abelard wins then $E \not\subseteq \Psi$, for all $E \in \text{EXT}_{\exists\forall}(\vec{\exists}\vec{\forall}(\wp))$.*

Proof. Let \mathcal{G}_\wp^Ψ be the game defined as prescribed in Construction 4.1. Obviously, this is a Borelian game, due to the hypothesis on the property Ψ .

Before continuing, first observe that, thanks to the specific structure of the game, every history $h \cdot p \in \text{Hist}_\alpha$ is bijectively correlated with the sequence of positions $\text{OBS}(h) \cdot p \in O^* \cdot P^\alpha$, for any player $\alpha \in \{\exists, \forall\}$. In other words, the functions $\text{SHR}_\alpha: \text{Hist}_\alpha \rightarrow O^* \cdot P^\alpha$ defined as $\text{SHR}_\alpha(h \cdot p) \stackrel{\text{def}}{=} \text{OBS}(h) \cdot p$ are bijective. Thanks to this observation, it is thus immediate to show that, for each strategy $\sigma_\exists \in \text{Strat}_\exists$, there is a unique function $\hat{\sigma}_\exists: O^* \cdot P^\exists \rightarrow P$ and, *vice versa*, for each function $\hat{\sigma}_\exists: O^* \cdot P^\exists \rightarrow P$, there is a unique strategy $\sigma_\exists \in \text{Strat}_\exists$ such that

$$\hat{\sigma}_\exists(\text{SHR}_\exists(h)) = \sigma_\exists(h), \text{ for all histories } h \in \text{Hist}_\exists.$$

Similarly, for each strategy $\sigma_\forall \in \text{Strat}_\forall$, there is a unique function $\hat{\sigma}_\forall: O^* \cdot P^\forall \rightarrow P$ and, *vice versa*, for each function $\hat{\sigma}_\forall: O^* \cdot P^\forall \rightarrow P$, there is a unique strategy $\sigma_\forall \in \text{Strat}_\forall$ satisfying the equality

$$\hat{\sigma}_\forall(\text{SHR}_\forall(h)) = \sigma_\forall(h), \text{ for all histories } h \in \text{Hist}_\forall.$$

3. The Herbrandization process [Bus98b; Hei67] is the dual of the well known Skolemization process and transforms a logic formula of the form $\exists x \forall y. \psi(x, y)$ into the equivalent (higher-order) formula $\forall F \exists x. \psi(x, F(x))$, where F is the *Herbrand function* for the universally-quantified variable y . The de-Herbrandizing process is the inverse transformation from $\forall F \exists x. \psi(x, F(x))$ to $\exists x \forall y. \psi(x, y)$. Note that here the process is applied at the meta level of the proof.

We can now proceed with the proof of the two properties.

Recall that $\text{EXT}_\alpha(\wp) \stackrel{\text{def}}{=} \text{EXT}_\alpha(\{\{\emptyset\}\}, \wp)$.

— [1] Since Eloise wins the game, she has a winning strategy, i.e., there is $\sigma_\exists \in \mathbf{Strat}_\exists$ such that $\text{OBS}(\text{PLAY}((\sigma_\exists, \sigma_\forall))) \in \mathbf{Win}$, for all $\sigma_\forall \in \mathbf{Strat}_\forall$. We want to prove that there exists $E \in \text{EXT}_{\exists\forall}(\vec{\forall\exists}(\wp))$ such that $E \subseteq \Psi$.

First, recall that $\vec{\forall\exists}(\wp) = \forall^R(\vec{p}) \cdot \exists^{\vec{\Theta}}(\vec{q})$, for some vectors of atomic propositions $(\vec{p}), (\vec{q}) \in s^*$ and quantifier specifications $\vec{\Theta} \in \mathbf{O}^{|\vec{q}|}$. Moreover, thanks to Proposition 4.4, the following claim can be proved by induction on the number of existential variables.

Claim 4.1. *$E \subseteq \Psi$, for some $E \in \text{EXT}_{\exists\forall}(\vec{\forall\exists}(\wp))$, iff there exists a vector of responses $\vec{F} \in \mathbf{Rsp}_{\vec{\Theta}}((\vec{p}))$ such that $\text{EXT}(\chi, \vec{F}, (\vec{q})) \in \Psi$, for all assignments $\chi \in \mathbf{Asg}((\vec{p}))$.*

Proof. As previously observed, $\vec{\forall\exists}(\wp) = \forall^R(\vec{p}) \cdot \exists^{\vec{\Theta}}(\vec{q})$, for some vectors $(\vec{p}), (\vec{q}) \in s^*$ and $\vec{\Theta} \in \mathbf{O}^{|\vec{q}|}$. Thus, $\text{EXT}_{\exists\forall}(\vec{\forall\exists}(\wp)) = \text{EXT}_{\exists\forall}(\forall^R(\vec{p}) \cdot \exists^{\vec{\Theta}}(\vec{q})) = \text{EXT}_{\exists\forall}(\text{EXT}_{\exists\forall}(\forall^R(\vec{p})), \exists^{\vec{\Theta}}(\vec{q})) = \text{EXT}_{\exists\forall}(\{\mathbf{Asg}((\vec{p}))\}, \exists^{\vec{\Theta}}(\vec{q}))$. At this point, the proof proceeds by induction on the length of the vector (\vec{q}) . If $|\vec{q}| = 0$, there is nothing really to prove, as the thesis follows immediately from the fact that $\text{EXT}_{\exists\forall}(\vec{\forall\exists}(\wp)) = \{\mathbf{Asg}((\vec{p}))\}$. Let us now consider the case $|\vec{q}| > 0$ and split both (\vec{q}) and $\vec{\Theta}$ as follows: $(\vec{q}) = (\vec{q}') \cdot q$ and $\vec{\Theta} = \vec{\Theta}' \cdot \Theta$. Obviously, $\text{EXT}_{\exists\forall}(\vec{\forall\exists}(\wp)) = \text{EXT}_{\exists\forall}(\text{EXT}_{\exists\forall}(\{\mathbf{Asg}((\vec{p}))\}, \exists^{\vec{\Theta}'}(\vec{q}')), \exists^\Theta q)$. Now, by Item 1 of Proposition 4.4, $E \subseteq \Psi$, for some $E \in \text{EXT}_{\exists\forall}(\vec{\forall\exists}(\wp))$, iff there exist a response $F \in \mathbf{Rsp}_\Theta((\vec{p}) \cdot (\vec{q}')')$ and a set $X \in \text{EXT}_{\exists\forall}(\{\mathbf{Asg}((\vec{p}))\}, \exists^{\vec{\Theta}'}(\vec{q}')')$ such that $\text{EXT}(X, F, p) \subseteq \Psi$. The latter inclusion can be rewritten as $X \subseteq \text{PRJ}(\Psi, F, q)$, where $\text{PRJ}(\Psi, F, q) \stackrel{\text{def}}{=} \{\chi \in \mathbf{Asg}((\vec{p}) \cdot (\vec{q}')') \mid \text{EXT}(\chi, F, q) \in \Psi\}$. At this point, by the inductive hypothesis applied to the inclusion $X \subseteq \text{PRJ}(\Psi, F, q)$, for some $X \in \text{EXT}_{\exists\forall}(\{\mathbf{Asg}((\vec{p}))\}, \exists^{\vec{\Theta}'}(\vec{q}')')$, we obtain that $E \subseteq \Psi$, for some $E \in \text{EXT}_{\exists\forall}(\vec{\forall\exists}(\wp))$, iff there exist a response $F \in \mathbf{Rsp}_\Theta((\vec{p}) \cdot (\vec{q}')')$ and a vector of responses $\vec{F}' \in \mathbf{Rsp}_{\vec{\Theta}'}, ((\vec{p}))$ such that $\text{EXT}(\mathbf{Asg}((\vec{p})), \vec{F}', (\vec{q}')') \subseteq \text{PRJ}(\Psi, F, q)$. The latter inclusion can now be rewritten as $\text{EXT}(\text{EXT}(\mathbf{Asg}((\vec{p})), \vec{F}', (\vec{q}')'), F, q) \subseteq \Psi$. To conclude the proof, the vector of responses $\vec{F} \in \mathbf{Rsp}_{\vec{\Theta}}((\vec{p}))$ is obtained by juxtaposing the vector \vec{F}' with the response $F^* \in \mathbf{Rsp}_\Theta((\vec{p}))$ obtained by composing F with \vec{F}' as follows: $F^*(\chi) \stackrel{\text{def}}{=} F(\text{EXT}(\chi, \vec{F}', (\vec{q}')'))$. \square

Due to the above characterisation of the existence of a set $E \in \text{EXT}_{\exists\forall}(\vec{\forall\exists}(\wp))$ such

that $E \subseteq \Psi$, the thesis can be proved by defining a suitable vector of responses $\vec{F} \in \text{Rsp}_{\vec{\Theta}}((\vec{p}))$.

Consider an arbitrary assignment $\chi \in \text{Asg}((\vec{p}))$ and define the function $\hat{\sigma}_{\forall}^{\chi}: O^* \cdot P^{\forall} \rightarrow P$ as follows, for all finite sequences of observable positions $u \in O^*$ and Abelard's positions $\nu \in P^{\forall}$:

$$\hat{\sigma}_{\forall}^{\chi}(u \cdot \nu) \stackrel{\text{def}}{=} \begin{cases} \emptyset, f & \text{if } \nu \in O; \\ \nu[x \mapsto \chi(x)(|u|)], & \text{otherwise;} \end{cases}$$

where $x \in (\vec{p})$ is the atomic proposition at position $\#(\nu)$ in the prefix \wp , i.e., $(\wp)_{\#(\nu)} = \forall^R x$. Due to the bijective correspondence previously described, there is a unique strategy $\sigma_{\forall}^{\chi} \in \text{Strat}_{\forall}$ such that $\sigma_{\forall}^{\chi}(h) = \hat{\sigma}_{\forall}^{\chi}(\text{SHR}_{\forall}(h))$, for all histories $h \in \text{Hist}_{\forall}$. Obviously, the induced play $\rho^{\chi} \stackrel{\text{def}}{=} \text{PLAY}((\sigma_{\exists}, \sigma_{\forall}^{\chi}))$ is won by Eloise, i.e., $u^{\chi} \stackrel{\text{def}}{=} \text{OBS}(\rho^{\chi}) \in \text{Win}$.

Thanks to all the infinite sequences u^{χ} , one for each assignment $\chi \in \text{Asg}((\vec{p}))$, we can now define every component $(\vec{F})_i$ of the vector of responses $\vec{F} \in (\text{Rsp}((\vec{p})))^{|\vec{q}|}$ as follows, for all instants of time $t \in \mathbb{N}$, where $i \in \llbracket 0, \dots, |\vec{q}| \rrbracket$:

$$(\vec{F})_i(\chi)(t) \stackrel{\text{def}}{=} (u^{\chi})_t((\vec{q})_i).$$

It is not too hard to show that, by construction, this response complies with the vector $\vec{\Theta}$ of quantifier specifications.

Claim 4.2. $\vec{F} \in \text{Rsp}_{\vec{\Theta}}((\vec{p}))$.

At this point, for all assignments $\chi \in \text{Asg}((\vec{p}))$, let $\chi_{\vec{F}} \stackrel{\text{def}}{=} \text{EXT}(\chi, \vec{F}, (\vec{q}))$. We can argue that $\chi_{\vec{F}} \in \Psi$. Indeed, by construction of the strategy σ_{\forall}^{χ} and the vector of responses \vec{F} , it holds that $\chi_{\vec{F}}(x)(t) = (u^{\chi})_t(x)$, for all instants of time $t \in \mathbb{N}$ and atomic propositions $x \in (\vec{p}) \cdot (\vec{q})$. Hence, $\text{WRD}(\chi_{\vec{F}}) = u^{\chi}$, which implies $\chi_{\vec{F}} \in \Psi$, since $u^{\chi} \in \text{Win}$.

- [2] Since Abelard wins the game, he has a winning strategy, i.e., there is $\sigma_{\forall} \in \text{Strat}_{\forall}$ such that $\text{OBS}(\text{PLAY}((\sigma_{\exists}, \sigma_{\forall}))) \notin \text{Win}$, for all $\sigma_{\exists} \in \text{Strat}_{\exists}$. We want to prove that, for all $E \in \text{EXT}_{\exists\forall}(\vec{\exists}\forall(\wp))$, it holds that $E \not\subseteq \Psi$.

First, recall that $\vec{\exists}\forall(\wp) = \exists^R(\vec{q}) \cdot \forall^{\vec{\Theta}}(\vec{p})$, for some vectors of atomic propositions $(\vec{p}), (\vec{q}) \in s^*$ and quantifier specifications $\vec{\Theta} \in \mathcal{O}^{|\vec{p}|}$. Moreover, thanks to Proposition 4.4, the following claim can be proved by induction on the number of universal variables.

Claim 4.3. $E \not\subseteq \Psi$, for all $E \in \text{EXT}_{\exists\forall}(\overrightarrow{\exists\forall}(\varphi))$, iff there exists a vector of responses $\vec{G} \in \text{Rsp}_{\vec{\Theta}}((\vec{q}))$ such that $\text{EXT}(\chi, \vec{G}, (\vec{p})) \notin \Psi$, for all assignments $\chi \in \text{Asg}((\vec{q}))$.

Proof. For technical convenience, we prove the counter-positive version of the statement: $E \subseteq \Psi$, for some $E \in \text{EXT}_{\exists\forall}(\overrightarrow{\exists\forall}(\varphi))$, iff, for all vectors of responses $\vec{G} \in \text{Rsp}_{\vec{\Theta}}((\vec{q}))$, it holds that $\text{EXT}(\text{Asg}((\vec{q})), \vec{G}, (\vec{p})) \cap \Psi \neq \emptyset$. As previously observed, $\overrightarrow{\exists\forall}(\varphi) = \exists^R(\vec{q}) \cdot \forall^{\vec{\Theta}}(\vec{p})$, for some vectors $(\vec{p}), (\vec{q}) \in s^*$ and $\vec{\Theta} \in \mathcal{O}^{|\vec{p}|}$. Thus, $\text{EXT}_{\exists\forall}(\overrightarrow{\exists\forall}(\varphi)) = \text{EXT}_{\exists\forall}(\exists^R(\vec{q}) \cdot \forall^{\vec{\Theta}}(\vec{p})) = \text{EXT}_{\exists\forall}(\text{EXT}_{\exists\forall}(\exists^R(\vec{q})), \forall^{\vec{\Theta}}(\vec{p})) = \text{EXT}_{\exists\forall}(\{\{\chi\} \mid \chi \in \text{Asg}((\vec{q}))\}, \forall^{\vec{\Theta}}(\vec{p}))$. At this point, the proof proceeds by induction on the length of the vector (\vec{p}) . If $|(\vec{p})| = 0$, there is nothing really to prove, as the thesis follows immediately from the fact that $\text{EXT}_{\exists\forall}(\overrightarrow{\exists\forall}(\varphi)) = \{\{\chi\} \mid \chi \in \text{Asg}((\vec{q}))\}$. Let us now consider the case $|(\vec{p})| > 0$ and split both (\vec{p}) and $\vec{\Theta}$ as follows: $(\vec{p}) = (\vec{p}') \cdot p$ and $\vec{\Theta} = \vec{\Theta}' \cdot \Theta$. Obviously, $\text{EXT}_{\exists\forall}(\overrightarrow{\exists\forall}(\varphi)) = \text{EXT}_{\exists\forall}(\text{EXT}_{\exists\forall}(\{\{\chi\} \mid \chi \in \text{Asg}((\vec{q}))\}, \forall^{\vec{\Theta}'}(\vec{p}')), \forall^{\Theta} p)$. Now, by Item 2 of Proposition 4.4, $E \subseteq \Psi$, for some $E \in \text{EXT}_{\exists\forall}(\overrightarrow{\exists\forall}(\varphi))$, iff, for all responses $G \in \text{Rsp}_{\Theta}((\vec{q}) \cdot (\vec{p}')')$, there exists a set $X \in \text{EXT}_{\exists\forall}(\{\{\chi\} \mid \chi \in \text{Asg}((\vec{q}))\}, \forall^{\vec{\Theta}'}(\vec{p}')')$ such that $\text{EXT}(X, G, p) \subseteq \Psi$. The latter inclusion can be rewritten as $X \subseteq \text{PRJ}(\Psi, G, p)$, where $\text{PRJ}(\Psi, G, p) \stackrel{\text{def}}{=} \{\chi \in \text{Asg}((\vec{q}) \cdot (\vec{p}')') \mid \text{EXT}(\chi, G, p) \in \Psi\}$. At this point, by the inductive hypothesis applied to the inclusion $X \subseteq \text{PRJ}(\Psi, G, p)$, for some $X \in \text{EXT}_{\exists\forall}(\{\{\chi\} \mid \chi \in \text{Asg}((\vec{q}))\}, \forall^{\vec{\Theta}'}(\vec{p}')')$, we obtain that $E \subseteq \Psi$, for some $E \in \text{EXT}_{\exists\forall}(\overrightarrow{\exists\forall}(\varphi))$, iff for all responses $G \in \text{Rsp}_{\Theta}((\vec{q}) \cdot (\vec{p}')')$ and vectors of responses $\vec{G}' \in \text{Rsp}_{\vec{\Theta}'}, ((\vec{q}))$, it holds that $\text{EXT}(\text{Asg}((\vec{q})), \vec{G}', (\vec{p}')') \cap \text{PRJ}(\Psi, G, p) \neq \emptyset$. The latter inequality can now be rewritten as $\text{EXT}(\text{EXT}(\text{Asg}((\vec{q})), \vec{G}', (\vec{p}')'), G, p) \cap \Psi \neq \emptyset$. To conclude the proof, it is enough to observe that the vectors of responses $\vec{G} \in \text{Rsp}_{\vec{\Theta}}((\vec{q}))$ can always be obtained by juxtaposing the vectors \vec{G}' with the responses $G^* \in \text{Rsp}_{\Theta}((\vec{q}))$ obtained by composing G with \vec{G}' as follows: $G^*(\chi) \stackrel{\text{def}}{=} G(\text{EXT}(\chi, \vec{G}', (\vec{p}')'))$. \square

Due to the above characterisation of non-existence of a set $E \in \text{EXT}_{\exists\forall}(\overrightarrow{\exists\forall}(\varphi))$ such that $E \subseteq \Psi$, the thesis can be proved by defining a suitable vector of responses $\vec{G} \in \text{Rsp}_{\vec{\Theta}}((\vec{q}))$.

Consider an arbitrary assignment $\chi \in \text{Asg}((\vec{q}))$ and define the function $\hat{\sigma}_{\exists}^{\chi}: O^* \cdot P^{\exists} \rightarrow P$ as follows, for all finite sequences of observable positions $u \in O^*$ and Eloise's positions $\nu \in P^{\exists}$:

$$\hat{\sigma}_{\exists}^{\chi}(u \cdot \nu) \stackrel{\text{def}}{=} \nu[x \mapsto \chi(x)(|u|)],$$

where $x \in (\vec{q})$ is the atomic proposition at position $\#(\nu)$ in the prefix \wp , i.e., $(\wp)_{\#(\nu)} = \exists^R x$. Due to the bijective correspondence previously described, there is a unique strategy $\sigma_{\exists}^x \in \mathbf{Strat}_{\exists}$ such that $\sigma_{\exists}^x(h) = \hat{\sigma}_{\exists}^x(\text{SHR}_{\exists}(h))$, for all histories $h \in \mathbf{Hist}_{\exists}$. Obviously, the induced play $\rho^x \stackrel{\text{def}}{=} \text{PLAY}((\sigma_{\exists}^x, \sigma_{\forall}))$ is won by Abelard, i.e., $u^x \stackrel{\text{def}}{=} \text{OBS}(\rho^x) \notin \text{Win}$.

Thanks to all the infinite sequences u^x , one for each assignment $\chi \in \mathbf{Asg}((\vec{q}))$, we can now define every component $(\vec{G})_i$ of the vector of responses $\vec{G} \in (\mathbf{Rsp}((\vec{q})))^{|\vec{p}|}$ as follows, for all instants of time $t \in \mathbb{N}$, where $i \in \llbracket 0, \dots, |\vec{p}| \rrbracket$:

$$(\vec{G})_i(\chi)(t) \stackrel{\text{def}}{=} (u^x)_t((\vec{p})_i).$$

It is not too hard to show that, by construction, this response complies with the vector $\vec{\Theta}$ of quantifier specifications.

Claim 4.4. $\vec{G} \in \mathbf{Rsp}_{\vec{\Theta}}((\vec{q}))$.

At this point, for all assignments $\chi \in \mathbf{Asg}((\vec{q}))$, let $\chi_{\vec{G}} \stackrel{\text{def}}{=} \text{EXT}(\chi, \vec{G}, (\vec{p}))$. We can argue that $\chi_{\vec{G}} \notin \Psi$. Indeed, by construction of the strategy σ_{\exists}^x and the vector of responses \vec{G} , it holds that $\chi_{\vec{G}}(x)(t) = (u^x)_t(x)$, for all instants of time $t \in \mathbb{N}$ and atomic propositions $x \in (\vec{q}) \cdot (\vec{p})$. Hence, $\text{WRD}(\chi_{\vec{G}}) = u^x$, which implies $\chi_{\vec{G}} \notin \Psi$, since $u^x \notin \text{Win}$. \square

The two conditions stated in Proposition 4.4 allow us to introduce a different, but equivalent (in terms of the equivalence relation \equiv between hyperteams), definition of evolution function that we call *normal*, in symbols nevl_{α} . This new notion will be useful to show important properties that would be, otherwise, much more cumbersome to prove by appealing directly to the original definition of the evolution function $\text{EXT}_{\alpha}(\cdot)$.

$$\text{nevl}_{\alpha}(\mathfrak{X}, Q^{\Theta} p) \stackrel{\text{def}}{=} \begin{cases} \text{EXT}_{\Theta}(\mathfrak{X}, p), & \text{if } Q \text{ is } \alpha\text{-coherent;} \\ \{\text{EXT}(\vec{\delta}, p) \mid \vec{\delta} \in \mathbf{Rsp}_{\Theta}(\text{AP}(\mathfrak{X})) \rightarrow \mathfrak{X}\}, & \text{otherwise;} \end{cases}$$

where $\text{EXT}(\vec{\delta}, p) \stackrel{\text{def}}{=} \bigcup \{\text{EXT}(\vec{\delta}(F), F, p) \mid F \in \text{DOM}(\vec{\delta})\}$. Intuitively, *w.r.t* $\text{EXT}_{\alpha}(\cdot)$, we just modified the non α -coherent case, in order to avoid the double application of the dualization function, by replacing this with a choice of a selection map $\vec{\delta} \in \mathbf{Rsp}_{\Theta}(\text{AP}(\mathfrak{X})) \rightarrow \mathfrak{X}$ selecting, in fact, for each Θ -response $F \in \mathbf{Rsp}_{\Theta}(\text{AP}(\mathfrak{X}))$, a set of assignments $\vec{\delta}(F) \in \mathfrak{X}$.

The new evolution operator lifts naturally to an arbitrary quantification prefix $\wp \in \mathbf{Q}$ as follows:

1. $\text{nevl}_\alpha(\mathfrak{X}, \epsilon) \stackrel{\text{def}}{=} \mathfrak{X}$;
2. $\text{nevl}_\alpha(\mathfrak{X}, Q^\ominus p. \wp) \stackrel{\text{def}}{=} \text{nevl}_\alpha(\text{nevl}_\alpha(\mathfrak{X}, Q^\ominus p), \wp)$.

As we have done for $\text{EXT}_\alpha(\cdot)$, we also set $\text{nevl}_\alpha(\wp) \stackrel{\text{def}}{=} \text{nevl}_\alpha(\{\{\emptyset\}\}, \wp)$.

Example 4.14. Consider the quantifier $\exists q$ and the hyperteam $\mathfrak{X} = \{X_1, X_2\}$ with $X_i = \{\chi_i\}$, where $i \in \{1, 2\}$ and $\chi_1 \stackrel{\text{def}}{=} \{p \mapsto \top^\omega\}$ and $\chi_2 \stackrel{\text{def}}{=} \{p \mapsto \perp^\omega\}$. Since $\exists q$ is $\exists\forall$ -coherent, we have

$$\text{nevl}_{\exists\forall}(\mathfrak{X}, \exists q) = \text{EXT}(\mathfrak{X}, q).$$

On the other hand, $\exists q$ is not $\forall\exists$ -coherent, thus

$$\text{nevl}_{\forall\exists}(\mathfrak{X}, \exists q) = \{\text{EXT}(\bar{\delta}, q) \mid \bar{\delta} \in \text{Rsp}(AP(\mathfrak{X})) \rightarrow \mathfrak{X}\}.$$

For instance, consider $\bar{\delta}_0 : \text{Rsp}_\Theta(AP(\mathfrak{X})) \rightarrow \mathfrak{X}$ defined as follows:

$$\bar{\delta}_0(F) \stackrel{\text{def}}{=} \begin{cases} X_1, & \text{if } F(\chi_1)(0) = \top \\ X_2, & \text{otherwise.} \end{cases}$$

Intuitively, the selection function $\bar{\delta}_0$ bipartitions the responses according to the value that they assign to χ_1 at time 0, by associating each response with one of the two sets of assignments, X_1 or X_2 . We thus have

$$\begin{aligned} \text{EXT}(\bar{\delta}_0, q) &= \bigcup \{\text{EXT}(\bar{\delta}_0(F), F, q) \mid F \in \text{Rsp}_\Theta(AP(\mathfrak{X}))\} \\ &= \bigcup \{\text{EXT}(X_1, F, q) \mid F \in \text{Rsp}_\Theta(AP(\mathfrak{X})), F(\chi_1)(0) = \top\} \cup \\ &\quad \bigcup \{\text{EXT}(X_2, F, q) \mid F \in \text{Rsp}_\Theta(AP(\mathfrak{X})), F(\chi_1)(0) = \perp\}. \end{aligned}$$

Proposition 4.5. If $\mathfrak{X}_1 \equiv \mathfrak{X}_2$ then $\text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p) \equiv \text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p)$, for all hyperteam $\mathfrak{X}_1, \mathfrak{X}_2 \in \text{HypTeams}$, quantifier symbols $Q \in \{\exists, \forall\}$, quantifier specifications $\Theta \in \mathbb{O}$, and atomic propositions $p \in s \setminus AP(\mathfrak{X})$.

Proof. The proof proceeds by a case analysis on the coherence of α and Q .

- [Q is α -coherent] By definition, $\text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p) = \text{EXT}_\Theta(\mathfrak{X}_1, p)$ and $\text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p) = \text{EXT}_\Theta(\mathfrak{X}_2, p)$. Since $\mathfrak{X}_1 \equiv \mathfrak{X}_2$, by Lemma 2.9, it holds that $\text{EXT}_\Theta(\mathfrak{X}_1, p) \equiv \text{EXT}_\Theta(\mathfrak{X}_2, p)$, which conclude this case of the proof.
- [Q is not α -coherent] By definition, $\text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p) = \{\text{EXT}(\bar{\delta}, p) \mid \bar{\delta} \in \text{Rsp}_\Theta(AP(\mathfrak{X}_1)) \rightarrow \mathfrak{X}_1\}$ and $\text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p) = \text{EXT}_\Theta(\mathfrak{X}_2, p)$. We now prove the two

inclusions $\text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p)$ and $\text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p) \sqsupseteq \text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p)$ separately.

- $[\sqsubseteq]$ To prove $\text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p)$, we need to show that, for any $\Psi \in \text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p)$ there is $W_\Psi \in \text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p)$ such that $W_\Psi \subseteq \Psi$. Obviously, for any $\Psi \in \text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p)$, it holds that $\Psi = \text{EXT}(\bar{\delta}, p) = \bigcup \{ \text{EXT}(\bar{\delta}(F), F, p) \mid F \in \text{DOM}(\bar{\delta}) \}$, for some selection function $\bar{\delta} \in \text{Rsp}_\ominus(\text{AP}(\mathfrak{X}_1)) \rightarrow \mathfrak{X}_1$. This means that, for every $F \in \text{Rsp}_\ominus(\text{AP}(\mathfrak{X}_1))$, there is $X_F \stackrel{\text{def}}{=} \bar{\delta}(F) \in \mathfrak{X}_1$ such that $\text{EXT}(X_F, F, p) \subseteq \Psi$. Now, by Item 2 of Proposition 4.4, there exists $W_1 \in \text{EXT}_\alpha(\mathfrak{X}_1, Q^\ominus p)$ such that $W_1 \subseteq \Psi$. Since, thanks to Lemma 2.9, $\mathfrak{X}_1 \equiv \mathfrak{X}_2$ implies $\text{EXT}_\alpha(\mathfrak{X}_1, Q^\ominus p) \equiv \text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p)$, we have that there is $W_2 \in \text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p)$ such that $W_2 \subseteq W_1 \subseteq \Psi$. Finally, by setting $W_\Psi \stackrel{\text{def}}{=} W_2$, we obtain what is required.
- $[\sqsupseteq]$ To prove $\text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p) \sqsupseteq \text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p)$, we need to show that, for any $\Psi \in \text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p)$ there is $W_\Psi \in \text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p)$ such that $W_\Psi \subseteq \Psi$. By instantiating W with Ψ in Proposition 4.4, since $W = \Psi \in \text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p)$, from Item 2 we derive that, for all $F \in \text{Rsp}_\ominus(\text{AP}(\mathfrak{X}_2))$, there is $X_{F_2} \in \mathfrak{X}_2$ such that $\text{EXT}(X_{F_2}, F, p) \subseteq \Psi$. Now, since $\mathfrak{X}_1 \equiv \mathfrak{X}_2$, there is $X_{F_1} \in \mathfrak{X}_1$ such that $X_{F_1} \subseteq X_{F_2}$, which in turn implies $\text{EXT}(X_{F_1}, F, p) \subseteq \text{EXT}(X_{F_2}, F, p) \subseteq \Psi$. At this point, define the selection map $\bar{\delta} \in \text{Rsp}_\ominus(\text{AP}(\mathfrak{X}_1)) \rightarrow \mathfrak{X}_1$ as follows: $\bar{\delta}(F) \stackrel{\text{def}}{=} X_{F_1}$, for every $F \in \text{Rsp}_\ominus(\text{AP}(\mathfrak{X}_1)) = \text{Rsp}_\ominus(\text{AP}(\mathfrak{X}_2))$. Clearly, by setting $W_\Psi \stackrel{\text{def}}{=} \text{EXT}(\bar{\delta}, p)$, both $W_\Psi = \bigcup \{ \text{EXT}(\bar{\delta}(F), F, p) \mid F \in \text{DOM}(\bar{\delta}) \} \subseteq \Psi$ and $W_\Psi \in \text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p)$ holds true, as required.

This concludes the proof of the second and last case. \square

The following examples is meant to show how the normal α -evolution function for non-coherent quantifier simulates the α -evolution function for the same quantifier.

Example 4.15. *The function $\bar{\delta}_0$ of Example 4.14 can be viewed as a choice function on $\text{EXT}(\bar{\mathfrak{X}}, q)$. First, recall that $\bar{\mathfrak{X}} = \{X_{12}\}$ with $X_{12} = \{\chi_1, \chi_2\}$ and let $\dot{X} \in \text{EXT}(\bar{\mathfrak{X}}, q)$. Then, there is $F \in \text{Rsp}(\text{AP}(\bar{\mathfrak{X}}))$ such that $\dot{X} = \text{EXT}(X_{12}, F, q)$. If we define a choice function $\Gamma \in \text{Chc}(\text{EXT}(\bar{\mathfrak{X}}, q))$ so that*

$$\Gamma(\dot{X}) = \Gamma(\text{EXT}(X_{12}, F, q)) = \begin{cases} \chi_1[q \mapsto F(\chi_1)], & \text{if } F(\chi_1)(0) = \top, \\ \chi_2[q \mapsto F(\chi_2)], & \text{otherwise,} \end{cases}$$

it is straightforward to see that $\text{EXT}(\bar{\delta}_0, q) = \text{img}(\Gamma) \in \text{EXT}_{\forall\exists}(\bar{\mathfrak{X}}, \exists q)$.

Proposition 4.6. *If $\mathfrak{X}_1 \equiv \mathfrak{X}_2$ then $\text{nevl}_\alpha(\mathfrak{X}_1, \wp) \equiv \text{EXT}_\alpha(\mathfrak{X}_2, \wp)$, for all hypertteams $\mathfrak{X}_1, \mathfrak{X}_2 \in \text{HypTeams}$ and quantifier prefixes $\wp \in \mathbf{Q}$, with $\text{AP}(\mathfrak{X}) \cap \text{AP}(\wp) = \emptyset$.*

Proof. The proof proceeds by simple induction on the length of the quantification prefix \wp .

- [Base case $\wp = \varepsilon$] $\text{nevl}_\alpha(\mathfrak{X}_1, \varepsilon) = \mathfrak{X}_1 \equiv \mathfrak{X}_2 = \text{EXT}_\alpha(\mathfrak{X}_2, \varepsilon)$.
- [Inductive case $\wp = Q^\ominus p. \wp'$] By definition, we have $\text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p. \wp') = \text{nevl}_\alpha(\text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p), \wp')$ and $\text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p. \wp') = \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p), \wp')$. Now, by Proposition 4.5, $\text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p) \equiv \text{EXT}_\alpha(\mathfrak{X}_2, Q^\ominus p)$, since $\mathfrak{X}_1 \equiv \mathfrak{X}_2$. Thus, the thesis follows by a straightforward application of the inductive hypothesis. \square

In the following, by \mathcal{O}_R we denote the set of realizable quantifier specifications, i.e., quantifier specifications of the form $\langle \text{AP}, P_S \rangle$ for some set of atomic propositions $P_S \subseteq s$.

Proposition 4.7. *$\text{EXT}_\alpha(\mathfrak{X}, \overline{Q}^R p. Q^{\ominus \cup \langle \emptyset, p \rangle} q) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, Q^\ominus q. \overline{Q}^R p)$, for all hypertteams $\mathfrak{X} \in \text{HypTeams}$, α -coherent quantifier symbols $Q \in \{\exists, \forall\}$, quantifier specifications $\Theta \in \mathcal{O}_R$, and atomic propositions $p, q \in s \setminus \text{AP}(\mathfrak{X})$.*

Proof. Due to the specific definition of the normal evolution function $\text{nevl}_\alpha(\mathfrak{X}, \wp)$, and by exploiting Proposition 4.6, the following claim can be shown.

Claim 4.5. *The following two properties are equivalent:*

- $\text{EXT}_\alpha(\mathfrak{X}, \overline{Q}^R p. Q^{\ominus \cup \langle \emptyset, p \rangle} q) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, Q^\ominus q. \overline{Q}^R p)$;
- for all $(\Theta \cup \langle \emptyset, p \rangle)$ -responses $J \in \text{Rsp}_{\Theta \cup \langle \emptyset, p \rangle}(\text{AP}(\mathfrak{X}) \cup \{p\})$, functions $\tilde{\delta} \in \text{Rsp}_R(\text{AP}(\mathfrak{X})) \rightarrow \mathfrak{X}$, and realizable responses $G \in \text{Rsp}_R(\text{AP}(\mathfrak{X}) \cup \{q\})$, there exists a Θ -response $F \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}))$ and a set of assignments $X \in \mathfrak{X}$ such that $\text{EXT}(\text{EXT}(X, F, q), G, p) \subseteq \text{EXT}(\text{EXT}(\tilde{\delta}, p), J, q)$.

Proof. By Proposition 4.6, the inclusion $\text{EXT}_\alpha(\mathfrak{X}, \overline{Q}^R p. Q^{\ominus \cup \langle \emptyset, p \rangle} q) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, Q^\ominus q. \overline{Q}^R p)$ is equivalent to the inclusion $\text{nevl}_\alpha(\mathfrak{X}, \overline{Q}^R p. Q^{\ominus \cup \langle \emptyset, p \rangle} q) \sqsubseteq \text{nevl}_\alpha(\mathfrak{X}, Q^\ominus q. \overline{Q}^R p)$, which in turn means that, for all sets $W_1 \in \text{nevl}_\alpha(\mathfrak{X}, \overline{Q}^R p. Q^{\ominus \cup \langle \emptyset, p \rangle} q)$, there exists a set $W_2 \in \text{nevl}_\alpha(\mathfrak{X}, Q^\ominus q. \overline{Q}^R p)$ such that $W_2 \subseteq W_1$. Now, by definition of normal evolution function, we have that

$$\text{nevl}_\alpha(\mathfrak{X}, \overline{Q}^R p. Q^{\ominus \cup \langle \emptyset, p \rangle} q) = \text{EXT}_{\Theta \cup \langle \emptyset, p \rangle}(\{\text{EXT}(\tilde{\delta}, p) \mid \tilde{\delta} \in \text{Rsp}_R(\text{AP}(\mathfrak{X})) \rightarrow \mathfrak{X}\}, q)$$

and

$$\text{nevl}_\alpha(\mathfrak{X}, Q^\ominus q. \overline{Q}^R p) = \{\text{EXT}(\tilde{\delta}, p) \mid \tilde{\delta} \in \text{Rsp}_R(\text{AP}(\mathfrak{X}) \cup \{q\}) \rightarrow \text{EXT}_\Theta(\mathfrak{X}, q)\}.$$

Thus, every set W_1 is equal to $\text{EXT}(\text{EXT}(\bar{\delta}, p), J, q)$, for some $(\Theta \cup \langle \emptyset, p \rangle)$ -response $J \in \text{Rsp}_{\Theta \cup \langle \emptyset, p \rangle}(\text{AP}(\mathfrak{X}) \cup \{p\})$ and selection function $\bar{\delta} \in \text{Rsp}_R(\text{AP}(\mathfrak{X})) \rightarrow \mathfrak{X}$, while every set W_2 is equal to $\text{EXT}(\bar{\delta}', p)$, for some selection function $\bar{\delta}' \in \text{Rsp}_R(\text{AP}(\mathfrak{X}) \cup \{q\}) \rightarrow \text{EXT}_{\Theta}(\mathfrak{X}, q)$. As a consequence, the previous property concerning the inclusion $W_2 \subseteq W_1$ can be equivalently rewritten as follows: for all $(\Theta \cup \langle \emptyset, p \rangle)$ -responses $J \in \text{Rsp}_{\Theta \cup \langle \emptyset, p \rangle}(\text{AP}(\mathfrak{X}) \cup \{p\})$ and selection functions $\bar{\delta} \in \text{Rsp}_R(\text{AP}(\mathfrak{X})) \rightarrow \mathfrak{X}$, there exists a selection function $\bar{\delta}' \in \text{Rsp}_R(\text{AP}(\mathfrak{X}) \cup \{q\}) \rightarrow \text{EXT}_{\Theta}(\mathfrak{X}, q)$ such that $\text{EXT}(\bar{\delta}', p) \subseteq \text{EXT}(\text{EXT}(\bar{\delta}, p), J, q)$. Since, $\text{EXT}(\bar{\delta}', p) = \bigcup \{\text{EXT}(\bar{\delta}'(G), G, p) \mid G \in \text{Rsp}_R(\text{AP}(\mathfrak{X}) \cup \{q\})\}$, the inclusion $\text{EXT}(\bar{\delta}', p) \subseteq \text{EXT}(\text{EXT}(\bar{\delta}, p), J, q)$ is equivalent to $\text{EXT}(\bar{\delta}'(G), G, p) \subseteq \text{EXT}(\text{EXT}(\bar{\delta}, p), J, q)$, for all realizable responses $G \in \text{Rsp}_R(\text{AP}(\mathfrak{X}) \cup \{q\})$. Hence, up to this point, we have proved that the following two properties are equivalent:

- $\text{EXT}_{\alpha}(\mathfrak{X}, \bar{Q}^R p \cdot Q^{\Theta \cup \langle \emptyset, p \rangle} q) \sqsubseteq \text{EXT}_{\alpha}(\mathfrak{X}, Q^{\Theta} q \cdot \bar{Q}^R p)$;
- for all $(\Theta \cup \langle \emptyset, p \rangle)$ -responses $J \in \text{Rsp}_{\Theta \cup \langle \emptyset, p \rangle}(\text{AP}(\mathfrak{X}) \cup \{p\})$ and functions $\bar{\delta} \in \text{Rsp}_R(\text{AP}(\mathfrak{X})) \rightarrow \mathfrak{X}$, there exists a function $\bar{\delta}' \in \text{Rsp}_R(\text{AP}(\mathfrak{X}) \cup \{q\}) \rightarrow \text{EXT}_{\Theta}(\mathfrak{X}, q)$ such that, for all realizable responses $G \in \text{Rsp}_R(\text{AP}(\mathfrak{X}) \cup \{q\})$, it holds that $\text{EXT}(\bar{\delta}'(G), G, p) \subseteq \text{EXT}(\text{EXT}(\bar{\delta}, p), J, q)$.

Now, by deSkolemizing the existential quantification of $\bar{\delta}'$ *w.r.t* the universal quantification of G , the second point is equivalent to the following: for all $(\Theta \cup \langle \emptyset, p \rangle)$ -responses $J \in \text{Rsp}_{\Theta \cup \langle \emptyset, p \rangle}(\text{AP}(\mathfrak{X}) \cup \{p\})$, functions $\bar{\delta} \in \text{Rsp}_R(\text{AP}(\mathfrak{X})) \rightarrow \mathfrak{X}$, and realizable responses $G \in \text{Rsp}_R(\text{AP}(\mathfrak{X}) \cup \{q\})$, there exists a set $Y \in \text{EXT}_{\Theta}(\mathfrak{X}, q)$ such that $\text{EXT}(Y, G, p) \subseteq \text{EXT}(\text{EXT}(\bar{\delta}, p), J, q)$. Finally, to obtain what is required by the statement of the claim, it is enough to observe that every set Y is equal to $\text{EXT}(X, F, q)$, for some Θ -response $F \in \text{Rsp}_{\Theta}(\text{AP}(\mathfrak{X}))$ and set $X \in \mathfrak{X}$. \square

Thanks to the given characterisation, we can now show that the inclusion $\text{EXT}_{\alpha}(\mathfrak{X}, \bar{Q}^R p \cdot Q^{\Theta \cup \langle \emptyset, p \rangle} q) \sqsubseteq \text{EXT}_{\alpha}(\mathfrak{X}, Q^{\Theta} q \cdot \bar{Q}^R p)$ actually holds true by proving the existence of a suitable response F and set of assignments X , in dependence of the responses J and G and the selection map $\bar{\delta}$, that satisfy the inclusion $\text{EXT}(\text{EXT}(X, F, q), G, p) \subseteq \text{EXT}(\text{EXT}(\bar{\delta}, p), J, q)$. In order to define such a response F , let us inductively construct, for every given assignment $\chi \in \text{Asg}(\text{AP}(\mathfrak{X}))$, the following infinite families of assignments $\{a_t^{\chi} \in \text{Asg}(\text{AP}(\mathfrak{X}) \cup \{p\})\}_{t \in \mathbb{N}}$, Boolean values $\{v_t^{\chi} \in \{\top, \perp\}\}_{t \in \mathbb{N}}$, and assignments $\{b_t^{\chi} \in \text{Asg}(\text{AP}(\mathfrak{X}) \cup \{q\})\}_{t \in \mathbb{N}}$, indexed by the time instants:

- [**Base step** $t = 0$] as base step, we choose $a_0^{\chi} \in \text{Asg}(\text{AP}(\mathfrak{X}) \cup \{p\})$ as an arbitrary assignment for which the equality $a_0^{\chi} \upharpoonright_{\text{AP}(\mathfrak{X})} = \chi$ holds true, the Boolean value $v_0^{\chi} \in \{\top, \perp\}$ as $J(a_0^{\chi})(0)$, i.e., $v_0^{\chi} \stackrel{\text{def}}{=} J(a_0^{\chi})(0)$, and $b_0^{\chi} \in \text{Asg}(\text{AP}(\mathfrak{X}) \cup \{q\})$ as an

arbitrary assignment with $b_0^x \upharpoonright_{\text{AP}(\mathfrak{X})} = \chi$ such that, at time 0 on the variable q , assumes v_0^x as value, i.e., $b_0^x(q)(0) = v_0^x$;

- [**Inductive step** $t > 0$] as inductive step, we derive the assignment $a_t^x \in \text{Asg}(\text{AP}(\mathfrak{X}) \cup \{p\})$ from $G(b_{t-1}^x)$, i.e., $a_t^x \stackrel{\text{def}}{=} \chi[p \mapsto G(b_{t-1}^x)]$, and the Boolean value $v_t^x \in \{\top, \perp\}$ from $J(a_t^x)(t)$, i.e., $v_t^x \stackrel{\text{def}}{=} J(a_t^x)(t)$; moreover, we choose $b_t^x \in \text{Asg}(\text{AP}(\mathfrak{X}) \cup \{q\})$ as an arbitrary assignment with $b_t^x \upharpoonright_{\text{AP}(\mathfrak{X})} = \chi$ such that, on the variable q , is equal to b_{t-1}^x up to time t excluded and assumes v_t^x as value at time t , i.e., $b_t^x(q)(h) = b_{t-1}^x(q)(h)$, for all $h \in \llbracket 0, \dots, t \rrbracket$, and $b_t^x(q)(t) = v_t^x$.

The above inductive construction can be schematically summarised as follows, where, for every $t \in \mathbb{N}$, both g_t and j_t are temporal assignments, i.e., functions of the form $g_t, j_t \in \mathbb{N} \rightarrow \{\top, \perp\}$:

$a_0^x \stackrel{\text{def}}{=} \chi[p \mapsto g_0]$, for some g_0 ;	$v_0^x \stackrel{\text{def}}{=} J(a_0^x)(0)$;	$b_0^x \stackrel{\text{def}}{=} \chi[q \mapsto j_0]$, for some j_0 with $j_0(0) = v_0^x$;
$a_t^x \stackrel{\text{def}}{=} \chi[p \mapsto G(b_{t-1}^x)]$;	$v_t^x \stackrel{\text{def}}{=} J(a_t^x)(t)$;	$b_t^x \stackrel{\text{def}}{=} \chi[q \mapsto j_t]$, for some j_t such that, for all $h \in \llbracket 0, \dots, t \rrbracket$, $j_t(h) = \begin{cases} j_{t-1}(h), & \text{if } h < t; \\ v_t^x, & \text{if } h = t. \end{cases}$

Thanks to the infinite family of Boolean values $\{v_t^x \in \{\top, \perp\}\}_{t \in \mathbb{N}}$, one for each assignment $\chi \in \text{Asg}(\text{AP}(\mathfrak{X}))$, we can define the response $F \in \text{Rsp}(\text{AP}(\mathfrak{X}))$ as follows, for every instant of time $t \in \mathbb{N}$:

$$F(\chi)(t) \stackrel{\text{def}}{=} v_t^x.$$

It is easy to show that this response complies with the quantifier specification Θ , since the response J , from which F is derived, is compliant with the quantifier specification $\Theta \cup \langle \emptyset, p \rangle$.

Claim 4.6. $F \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}))$.

Before continuing, let us first introduce the response $H \in \text{Rsp}(\text{AP}(\mathfrak{X}))$ as follows, for every assignment $\chi \in \text{Asg}(\text{AP}(\mathfrak{X}))$:

$$H(\chi) \stackrel{\text{def}}{=} \text{EXT}(\text{EXT}(\chi, F, q), G, p)(p).$$

It is not hard to verify that such a response is realizable, since F is Θ -compliant and G is realizable.

Claim 4.7. $H \in \text{Rsp}_R(AP(\mathfrak{X}))$.

At this point, consider the set of assignments $X \stackrel{\text{def}}{=} \bar{\partial}(H)$. Thanks to the specific definitions of the two responses F and H , the following claim can be proved.

Claim 4.8. $\text{EXT}(\text{EXT}(X, F, q), G, p) \subseteq \text{EXT}(\text{EXT}(X, H, p), J, q)$.

Now, it is obvious that $\text{EXT}(X, H, p) \subseteq \text{EXT}(\bar{\partial}, p)$, due to the definition of the latter and the choice of the team X , which immediately implies $\text{EXT}(\text{EXT}(X, H, p), J, q) \subseteq \text{EXT}(\text{EXT}(\bar{\partial}, p), J, q)$. Therefore, $\text{EXT}_{\text{EXT}(X, F, q)}([\]G) \subseteq \text{EXT}(\text{EXT}(\bar{\partial}, p), J, q)$, which concludes the proof. \square

Proposition 4.8. $\text{EXT}_\alpha(\mathfrak{X}, \bar{Q}^R(\vec{p}). Q^{\bar{\Theta} \cup \langle \emptyset, \vec{l} \rangle}(\vec{q})) \subseteq \text{EXT}_\alpha(\mathfrak{X}, Q^{\bar{\Theta}}(\vec{q}). \bar{Q}^R(\vec{p}))$, for all hyper-teams $\mathfrak{X} \in \text{HypTeams}$, α -coherent quantifier symbols $Q \in \{\exists, \forall\}$, vectors of quantifier specifications $\bar{\Theta} \in \mathcal{O}_R^*$, and vectors of atomic propositions $(\vec{p}), (\vec{q}) \in (s \setminus AP(\mathfrak{X}))^*$, with $|(\vec{q})| = |\bar{\Theta}|$.

Proof. The proof of the statements proceeds by combining two independent inductions. In particular, we first show, by exploiting Proposition 4.7 via an induction on the length of the vector of atomic propositions (\vec{p}) , that $\text{EXT}_\alpha(\mathfrak{X}, \bar{Q}^R(\vec{p}). Q^{\Theta \cup \langle \emptyset, \vec{l} \rangle} q) \subseteq \text{EXT}_\alpha(\mathfrak{X}, Q^\Theta q. \bar{Q}^R(\vec{p}))$. Indeed, one can easily verify the correctness of the following chain of equalities/inequalities:

$$\text{EXT}_\alpha(\mathfrak{X}, \bar{Q}^R(\vec{p}). \bar{Q}^R p. Q^{\Theta \cup \langle \emptyset, (\vec{p}) \rangle} q) = \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \bar{Q}^R(\vec{p})), \bar{Q}^R p. Q^{(\Theta \cup \langle \emptyset, \vec{l} \rangle) \cup \langle \emptyset, p \rangle} q) \quad (4.1a)$$

$$\sqsubseteq \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \bar{Q}^R(\vec{p})), Q^{\Theta \cup \langle \emptyset, \vec{l} \rangle} q. \bar{Q}^R p) \quad (4.1b)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \bar{Q}^R(\vec{p}). Q^{\Theta \cup \langle \emptyset, \vec{l} \rangle} q), \bar{Q}^R p) \quad (4.1c)$$

$$\sqsubseteq \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, Q^\Theta q. \bar{Q}^R(\vec{p})), \bar{Q}^R p) \quad (4.1d)$$

$$= \text{EXT}_\alpha(\mathfrak{X}, Q^\Theta q. \bar{Q}^R(\vec{p}). \bar{Q}^R p). \quad (4.1e)$$

Steps 4.1a, 4.1c, and 4.1e are due to the definition of evolution function of a quantifier prefix, Step 4.1b is due to Proposition 4.7 applied to the outer evolution function, and, finally, Step 4.1d is just an application of the inductive hypothesis to the inner evolution function combined with the monotonicity property of Lemma 2.9.

At this point, by exploiting what we have just derived via an induction on the length of the vector of atomic propositions (\vec{q}) , we can prove the correctness of the statement by

means of the following chain of equalities/inequalities:

$$\text{EXT}_\alpha(\mathfrak{X}, \overline{Q}^R(\vec{p}) \cdot Q^{\overline{\Theta \cup \{\emptyset, \vec{l}\}}}(\vec{q}) \cdot Q^{\Theta \cup \{\emptyset, \vec{l}\}} q) = \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \overline{Q}^R(\vec{p}) \cdot Q^{\overline{\Theta \cup \{\emptyset, \vec{l}\}}}(\vec{q})), Q^{\Theta \cup \{\emptyset, \vec{l}\}} q) \quad (4.2a)$$

$$\sqsubseteq \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, Q^{\overline{\Theta}}(\vec{q}) \cdot \overline{Q}^R(\vec{p})), Q^{\Theta \cup \{\emptyset, \vec{l}\}} q) \quad (4.2b)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, Q^{\overline{\Theta}}(\vec{q})), \overline{Q}^R(\vec{p}) \cdot Q^{\Theta \cup \{\emptyset, \vec{l}\}} q) \quad (4.2c)$$

$$\sqsubseteq \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, Q^{\overline{\Theta}}(\vec{q})), Q^\Theta q \cdot \overline{Q}^R(\vec{p})) \quad (4.2d)$$

$$= \text{EXT}_\alpha(\mathfrak{X}, Q^{\overline{\Theta}}(\vec{q}) \cdot Q^\Theta q \cdot \overline{Q}^R(\vec{p})). \quad (4.2e)$$

Steps 4.2a, 4.2c, and 4.2e are due to the definition of evolution function of a quantifier prefix, Step 4.2b is just an application of the inductive hypothesis to the inner evolution function combined with the monotonicity property of Lemma 2.9, and, finally, Step 4.2d is due to the previously proved inequality $\text{EXT}_\alpha(\mathfrak{X}, \overline{Q}^R(\vec{p}) \cdot Q^{\Theta \cup \{\emptyset, \vec{l}\}} q) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, Q^\Theta q \cdot \overline{Q}^R(\vec{p}))$ applied to the outer evolution function. \square

Towards the proof of Proposition 4.3, we show the following more general result.

Proposition 4.9. *Let $\mathfrak{X} \in \text{HypTeams}$ be an hyperteam and $\wp, \wp_1, \wp_2, \wp_3 \in \mathbf{Q}_R$ realizable quantifier prefixes, such that $\wp = \wp_1 \cdot \wp_2 \cdot \wp_3$ and $AP(\wp) \cap AP(\mathfrak{X}) = \emptyset$. Then, it holds that $\text{EXT}_\alpha(\mathfrak{X}, \overrightarrow{\alpha}(\wp)) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \wp_1 \cdot \overrightarrow{\alpha}(\wp_2) \cdot \wp_3) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \wp) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \wp_1 \cdot \overrightarrow{\alpha}(\wp_2) \cdot \wp_3) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \overrightarrow{\alpha}(\wp))$.*

Proof. We separately prove the two chains of inequalities forming the statement, namely $\text{EXT}_\alpha(\mathfrak{X}, \overrightarrow{\alpha}(\wp)) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \wp_1 \cdot \overrightarrow{\alpha}(\wp_2) \cdot \wp_3) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \wp)$ and $\text{EXT}_\alpha(\mathfrak{X}, \wp) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \wp_1 \cdot \overrightarrow{\alpha}(\wp_2) \cdot \wp_3) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \overrightarrow{\alpha}(\wp))$, by using different technical expedients.

— $[\mathbf{ext}_\alpha(\mathfrak{X}, \overrightarrow{\alpha}(\wp)) \sqsubseteq \mathbf{ext}_\alpha(\mathfrak{X}, \wp_1 \cdot \overrightarrow{\alpha}(\wp_2) \cdot \wp_3) \sqsubseteq \mathbf{ext}_\alpha(\mathfrak{X}, \wp)]$ To prove that the first chain of inequalities holds, let us fix a well-founded preorder \preceq over the set of triples of quantifier prefixes $\mathsf{T} = \{\langle \wp_1, \wp_2, \wp_3 \rangle \in \mathbf{Q}_R \times \mathbf{Q}_R \times \mathbf{Q}_R \mid \wp = \wp_1 \cdot \wp_2 \cdot \wp_3\}$ defined as follows: $\langle \wp_1, \wp_2, \wp_3 \rangle \preceq \langle \wp_1', \wp_2', \wp_3' \rangle$ iff $\wp_2' = \wp_l \cdot \wp_2 \cdot \wp_r$, for some $\wp_l, \wp_r \in \mathbf{Q}_R$, i.e., \wp_2 is a (not necessarily proper) infix of \wp_2' . Notice that, given the definition of the set T , the relation $\langle \wp_1, \wp_2, \wp_3 \rangle \preceq \langle \wp_1', \wp_2', \wp_3' \rangle$ also implies $\wp_1 = \wp_1' \cdot \wp_l$ and $\wp_3 = \wp_r \cdot \wp_3'$. In addition, let us introduce $\overrightarrow{\alpha}(T)$ as an abbreviation for $\wp_1 \cdot \overrightarrow{\alpha}(\wp_2) \cdot \wp_3$, given an arbitrary triple $T = \langle \wp_1, \wp_2, \wp_3 \rangle \in \mathsf{T}$. Now, to show that the chain of inequalities

holds true, it is enough to prove that $\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T')) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T))$, for all $T, T' \in \mathbb{T}$ with $T \preceq T'$. The proof shall proceed by structural induction on the preorder \preceq .

- [**Base case** $T = T'$] Obviously $\vec{\alpha}(T) = \vec{\alpha}(T')$. Thus, the property trivially holds, as $\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T')) = \text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T))$.
- [**Inductive case** $T \prec T'$] Since $T \prec T'$, there necessarily exists a triple $T'' = \langle \wp_1'', \wp_2'', \wp_3'' \rangle \in \mathbb{T}$ such that $T \prec T'' \preceq T'$ and either

1. $\wp_1 = \wp_1'' \cdot Q^R p$, $\wp_2'' = Q^R p \cdot \wp_2$, and $\wp_3 = \wp_3''$, or
2. $\wp_1 = \wp_1''$, $\wp_2'' = \wp_2 \cdot Q^R p$, and $\wp_3 = Q^R p \cdot \wp_3''$,

for some quantifier symbol $Q \in \{\exists, \forall\}$ and atomic proposition $p \in s$. By inductive hypothesis, it holds that $\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T')) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T''))$. Thus, to conclude, we need to show that $\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T'')) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T))$. If $\vec{\alpha}(T) = \vec{\alpha}(T'')$, there is nothing really to prove, as $\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T'')) = \text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T))$. Hence, let us assume $\vec{\alpha}(T) \neq \vec{\alpha}(T'')$. The proof now proceeds with the following case analysis.

1. [$\wp_1 = \wp_1'' \cdot Q^R p$, $\wp_2'' = Q^R p \cdot \wp_2$, **and** $\wp_3 = \wp_3''$] First observe that Q is α -coherent. If this were not the case, indeed, we would have had $\vec{\alpha}(\wp_2'') = \vec{\alpha}(Q^R p \cdot \wp_2) = Q^R p \cdot \vec{\alpha}(\wp_2)$, which in turn would have implied $\vec{\alpha}(T'') = \wp_1'' \cdot \vec{\alpha}(\wp_2'') \cdot \wp_3'' = \wp_1'' \cdot \vec{\alpha}(Q^R p \cdot \wp_2) \cdot \wp_3'' = \wp_1'' \cdot Q^R p \cdot \vec{\alpha}(\wp_2) \cdot \wp_3'' = \wp_1 \cdot \vec{\alpha}(\wp_2) \cdot \wp_3 = \vec{\alpha}(T)$, contradicting the previous assumption $\vec{\alpha}(T) \neq \vec{\alpha}(T'')$. Both $\vec{\alpha}(\wp_2)$ and $\vec{\alpha}(\wp_2'')$ are prefix canonicalisation, featuring at most one quantifier alternation starting with a $\bar{\alpha}$ -coherent quantifier \bar{Q} . Specifically, these can be written as $\vec{\alpha}(\wp_2) = \bar{Q}^R(\vec{q}) \cdot Q^{\bar{\Theta}\vec{r}}$ and $\vec{\alpha}(\wp_2'') = \vec{\alpha}(Q^R p \cdot \wp_2) = \bar{Q}^R(\vec{q}) \cdot Q^{R \cup \langle \emptyset, \vec{q} \rangle} p \cdot Q^{\bar{\Theta}\vec{r}}$, for some vectors of atomic propositions (\vec{q}) and \vec{r} , and a vector of quantifiers specifications $\bar{\Theta} \in \mathbb{O}_R^*$ with $|\bar{\Theta}| = |\vec{r}|$. At this point, the induction proof terminates by checking the

following chain of equalities/inequalities:

$$\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T'')) = \text{EXT}_\alpha(\mathfrak{X}, \wp_1'' \cdot \vec{\alpha}(\wp_2'') \cdot \wp_3'') \quad (4.3a)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1''), \vec{\alpha}(\wp_2'')), \wp_3'') \quad (4.3b)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1''), \overline{Q}^R(\vec{q}) \cdot Q^{R \cup \langle \emptyset, (\vec{q}) \rangle} p \cdot Q^{\vec{\Theta} \vec{r}}), \wp_3'') \quad (4.3c)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1''), \overline{Q}^R(\vec{q}) \cdot Q^{R \cup \langle \emptyset, (\vec{q}) \rangle} p), Q^{\vec{\Theta} \vec{r}} \cdot \wp_3'') \quad (4.3d)$$

$$\sqsubseteq \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1''), Q^R p \cdot \overline{Q}^R(\vec{q})), Q^{\vec{\Theta} \vec{r}} \cdot \wp_3'') \quad (4.3e)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1''), Q^R p \cdot \overline{Q}^R(\vec{q}) \cdot Q^{\vec{\Theta} \vec{r}}), \wp_3'') \quad (4.3f)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1'' \cdot Q^R p), \overline{Q}^R(\vec{q}) \cdot Q^{\vec{\Theta} \vec{r}}), \wp_3'') \quad (4.3g)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1), \vec{\alpha}(\wp_2)), \wp_3) \quad (4.3h)$$

$$= \text{EXT}_\alpha(\mathfrak{X}, \wp_1 \cdot \vec{\alpha}(\wp_2) \cdot \wp_3) \quad (4.3i)$$

$$= \text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T)). \quad (4.3j)$$

Step 4.3e is due to Proposition 4.8 applied to $\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1''), \overline{Q}^R(\vec{q}) \cdot Q^{R \cup \langle \emptyset, (\vec{q}) \rangle} p)$ combined with Lemma 2.9. All the other steps are just immediate consequences of the definition of evolution function and the structure of both the quantifier prefixes \wp_1'' , \wp_2'' , and \wp_3'' , and the canonical forms $\vec{\alpha}(T)$ and $\vec{\alpha}(T'')$.

2. [$\wp_1 = \wp_1''$, $\wp_2'' = \wp_2 \cdot Q^R p$, and $\wp_3 = Q^R p \cdot \wp_3''$] Similarly to the previous case, from $\vec{\alpha}(T) \neq \vec{\alpha}(T'')$, one can derive that Q is $\bar{\alpha}$ -coherent. Consequently, $\vec{\alpha}(\wp_2)$ and $\vec{\alpha}(\wp_2'')$ can be written as $\vec{\alpha}(\wp_2) = Q^R(\vec{q}) \cdot \overline{Q}^{\vec{\Theta} \vec{r}}$ and $\vec{\alpha}(\wp_2'') = \vec{\alpha}(\wp_2 \cdot Q^R p) = Q^R(\vec{q}) \cdot Q^R p \cdot \overline{Q}^{\vec{\Theta}' \vec{r}}$, for some vectors of atomic propositions (\vec{q}) and \vec{r} , and vectors of quantifiers specifications $\vec{\Theta}, \vec{\Theta}' \in \mathbb{O}_R^*$ with $|\vec{\Theta}| = |\vec{\Theta}'| = |\vec{r}|$ and $\vec{\Theta}' = \vec{\Theta} \cup \langle \emptyset, (\vec{q}) \rangle$. At this point, the induction

proof terminates by checking the following chain of equalities/inequalities:

$$\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T'')) = \text{EXT}_\alpha(\mathfrak{X}, \wp_1'' \cdot \vec{\alpha}(\wp_2'') \cdot \wp_3'') \quad (4.4a)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1''), \vec{\alpha}(\wp_2'')), \wp_3'') \quad (4.4b)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1''), Q^R(\vec{q}) \cdot Q^R p \cdot \overline{Q}^{\vec{\Theta}'} \vec{r}), \wp_3'') \quad (4.4c)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1'' \cdot Q^R(\vec{q})), Q^R p \cdot \overline{Q}^{\vec{\Theta}'} \vec{r}), \wp_3'') \quad (4.4d)$$

$$\sqsubseteq \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1'' \cdot Q^R(\vec{q})), \overline{Q}^{\vec{\Theta}} \vec{r} \cdot Q^R p), \wp_3'') \quad (4.4e)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1''), Q^R(\vec{q}) \cdot \overline{Q}^{\vec{\Theta}} \vec{r} \cdot Q^R p), \wp_3'') \quad (4.4f)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1''), Q^R(\vec{q}) \cdot \overline{Q}^{\vec{\Theta}} \vec{r}), Q^R p \cdot \wp_3'') \quad (4.4g)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1), \vec{\alpha}(\wp_2)), \wp_3) \quad (4.4h)$$

$$= \text{EXT}_\alpha(\mathfrak{X}, \wp_1 \cdot \vec{\alpha}(\wp_2) \cdot \wp_3) \quad (4.4i)$$

$$= \text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(T)). \quad (4.4j)$$

Step 4.4e is due to Proposition 4.8 applied to $\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp_1'' \cdot Q^R(\vec{q})), Q^R p \cdot \overline{Q}^{\vec{\Theta}'} \vec{r})$ combined with Lemma 2.9. All the other steps are just immediate consequences of the definition of evolution function and the structure of both the quantifier prefixes \wp_1'' , \wp_2'' , and \wp_3'' , and the canonical forms $\vec{\alpha}(T)$ and $\vec{\alpha}(T'')$.

- [$\text{ext}_\alpha(\mathfrak{X}, \wp) \sqsubseteq \text{ext}_\alpha(\mathfrak{X}, \wp_1 \cdot \vec{\alpha}(\wp_2) \cdot \wp_3) \sqsubseteq \text{ext}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp))$] In order to show that the second chain of inequalities holds as well, we first state the following two simple auxiliary results, one regarding a duality property between the two syntactic canonicalisations of a quantifier prefix and the other concerning the dualization of the evolution function.

Claim 4.9. $\overline{\vec{\alpha}(\wp)} = \vec{\alpha}(\overline{\wp})$, for all quantifier prefixes $\wp \in \mathbf{Q}$.

Claim 4.10. $\overline{\text{EXT}_\alpha(\mathfrak{X}, \wp)} \equiv \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{\wp})$, for all hypertteams $\mathfrak{X} \in \mathbf{HypTeams}$ and quantifier prefixes $\wp \in \mathbf{Q}$.

Proof. The proof proceeds by induction on the length of \wp .

- [**Base step** $\wp = \varepsilon$] $\overline{\text{EXT}_\alpha(\mathfrak{X}, \varepsilon)} = \overline{\mathfrak{X}} = \text{EXT}_\alpha(\overline{\mathfrak{X}}, \varepsilon)$.
- [**Inductive step** $\wp = Q^\ominus p. \wp'$] First notice that $\overline{\wp} = \overline{Q^\ominus p. \wp'}$ and observe that, thanks to the definition of evolution function and the inductive hypothesis, the following holds true:

$$\overline{\text{EXT}_\alpha(\mathfrak{X}, \wp)} = \overline{\text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, Q^\ominus p), \wp')} \equiv \text{EXT}_\alpha(\overline{\text{EXT}_\alpha(\mathfrak{X}, Q^\ominus p)}, \overline{\wp'}).$$

Let us now distinguish two cases based on the coherence of α and Q .

- [**Q is α -coherent**]

$$\overline{\text{EXT}_\alpha(\mathfrak{X}, \wp)} \equiv \text{EXT}_\alpha(\overline{\text{EXT}_\alpha(\mathfrak{X}, Q^\ominus p)}, \overline{\wp'}) \quad (4.5a)$$

$$= \text{EXT}_\alpha(\overline{\text{EXT}_\Theta(\mathfrak{X}, p)}, \overline{\wp'}) \quad (4.5b)$$

$$\equiv \text{EXT}_\alpha(\overline{\text{EXT}_\Theta(\overline{\mathfrak{X}}, p)}, \overline{\wp'}) \quad (4.5c)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{Q^\ominus p}), \overline{\wp'}) \quad (4.5d)$$

$$= \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{Q^\ominus p. \wp'}) \quad (4.5e)$$

$$= \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{\wp}) \quad (4.5f)$$

Step 4.5b and 4.5d are due to the definition of evolution function over a single quantifier, for the cases when α and Q are coherent and non-coherent, respectively. Step 4.5c is just a simple consequence of Lemma 2.1, 2.9, and 2.9. Finally, Step 4.5e is given by the definition of evolution function for quantifier prefixes.

- [**Q is not α -coherent**]

$$\overline{\text{EXT}_\alpha(\mathfrak{X}, \wp)} \equiv \text{EXT}_\alpha(\overline{\text{EXT}_\alpha(\mathfrak{X}, Q^\ominus p)}, \overline{\wp'}) \quad (4.6a)$$

$$= \text{EXT}_\alpha(\overline{\overline{\text{EXT}_\Theta(\mathfrak{X}, p)}}, \overline{\wp'}) \quad (4.6b)$$

$$\equiv \text{EXT}_\alpha(\text{EXT}_\Theta(\overline{\mathfrak{X}}, p), \overline{\wp'}) \quad (4.6c)$$

$$= \text{EXT}_\alpha(\text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{Q^\ominus p}), \overline{\wp'}) \quad (4.6d)$$

$$= \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{Q^\ominus p. \wp'}) \quad (4.6e)$$

$$= \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{\wp}) \quad (4.6f)$$

Step 4.6b and 4.6d are due to the definition of evolution function over a single quantifier, for the cases when α and Q are non-coherent and coherent,

respectively. Step 4.6c is just a simple consequence of Lemmas 2.1 and 2.9.

Finally, Step 4.6e is given by the definition of evolution function for quantifier prefixes. \square

In the first item of this proof, we have proved that $\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\varphi)) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \varphi_1. \vec{\alpha}(\varphi_2). \varphi_3) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \varphi)$ holds true for every $\mathfrak{X} \in \text{HypTeams}$ and $\varphi, \varphi_1, \varphi_2, \varphi_3 \in \mathbf{Q}_R$, with $\varphi = \varphi_1. \varphi_2. \varphi_3$. By instantiating \mathfrak{X} and φ with $\overline{\mathfrak{X}}$ and $\overline{\varphi}$, and observing that $\overline{\varphi} = \overline{\varphi_1}. \overline{\varphi_2}. \overline{\varphi_3}$, we obtain $\text{EXT}_\alpha(\overline{\mathfrak{X}}, \vec{\alpha}(\overline{\varphi})) \sqsubseteq \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{\varphi_1}. \vec{\alpha}(\overline{\varphi_2}). \overline{\varphi_3}) \sqsubseteq \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{\varphi})$. Now, thanks to Claims 4.9 and 4.10 above, we obtain $\overline{\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\varphi))} \sqsubseteq \overline{\text{EXT}_\alpha(\mathfrak{X}, \varphi_1. \vec{\alpha}(\varphi_2). \varphi_3)} \sqsubseteq \overline{\text{EXT}_\alpha(\mathfrak{X}, \varphi)}$, as shown in the following two chains of equivalences/inequalities:

$$\overline{\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\varphi))} \equiv \text{EXT}_\alpha(\overline{\mathfrak{X}}, \vec{\alpha}(\overline{\varphi})) \quad (4.7a)$$

$$= \text{EXT}_\alpha(\overline{\mathfrak{X}}, \vec{\alpha}(\overline{\varphi})) \quad (4.7b)$$

$$\sqsubseteq \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{\varphi_1}. \vec{\alpha}(\overline{\varphi_2}). \overline{\varphi_3}) \quad (4.7c)$$

$$= \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{\varphi_1}. \vec{\alpha}(\overline{\varphi_2}). \overline{\varphi_3}) \quad (4.7d)$$

$$= \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{\varphi_1. \vec{\alpha}(\varphi_2). \varphi_3}) \quad (4.7e)$$

$$\equiv \overline{\text{EXT}_\alpha(\mathfrak{X}, \varphi_1. \vec{\alpha}(\varphi_2). \varphi_3)}. \quad (4.7f)$$

$$\overline{\text{EXT}_\alpha(\mathfrak{X}, \varphi_1. \vec{\alpha}(\varphi_2). \varphi_3)} \equiv \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{\varphi_1. \vec{\alpha}(\varphi_2). \varphi_3}) \quad (4.7g)$$

$$= \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{\varphi_1}. \vec{\alpha}(\overline{\varphi_2}). \overline{\varphi_3}) \quad (4.7h)$$

$$= \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{\varphi_1}. \vec{\alpha}(\overline{\varphi_2}). \overline{\varphi_3}) \quad (4.7i)$$

$$\sqsubseteq \text{EXT}_\alpha(\overline{\mathfrak{X}}, \overline{\varphi}) \quad (4.7j)$$

$$\equiv \overline{\text{EXT}_\alpha(\mathfrak{X}, \varphi)}. \quad (4.7k)$$

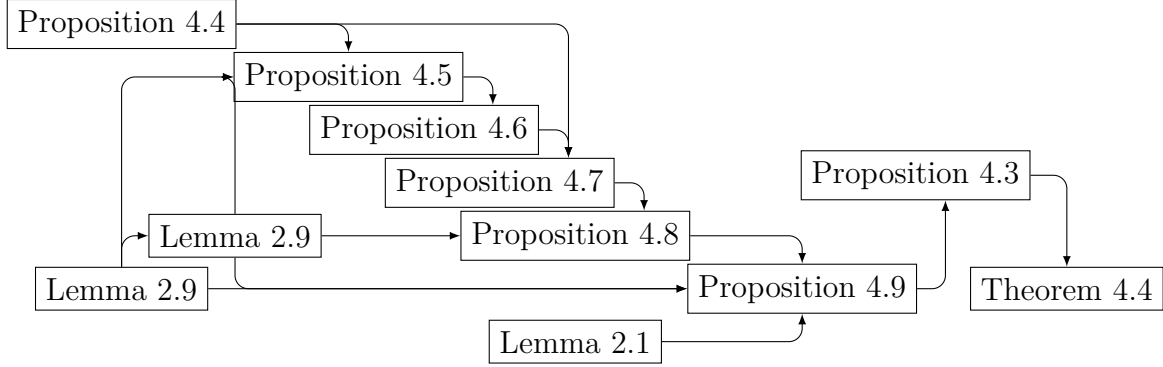
At this point, thanks to Lemmas 2.1 and 2.9, we derive $\text{EXT}_\alpha(\mathfrak{X}, \varphi) \sqsubseteq \overline{\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\varphi))} \sqsubseteq \overline{\text{EXT}_\alpha(\mathfrak{X}, \varphi_1. \vec{\alpha}(\varphi_2). \varphi_3)} \sqsubseteq \overline{\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\varphi))} \sqsubseteq \overline{\text{EXT}_\alpha(\mathfrak{X}, \varphi)}$. \square

The following proposition is now an immediate consequence of the above result.

Proposition 4.3. $\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\varphi)) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \varphi) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\varphi))$, for all hypertteams $\mathfrak{X} \in \text{HypTeams}$ and realizable quantifier prefixes $\varphi \in \mathbf{Q}_R$, with $AP(\varphi) \cap AP(\mathfrak{X}) = \emptyset$.

At this point, we have proven everything used in the proof of Theorem 4.4 from the main paper. Here is the graph of dependency presenting the propositions used for this

proof.



Proposition 4.10. $EXT_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1, \wp_2)) \sqsubseteq \mathfrak{Y} \sqsubseteq EXT_\alpha(\mathfrak{X}, \wp_1, \wp_2)$, for all hyperteams $\mathfrak{X} \in \text{HypTeams}$ and $\mathfrak{Y} \in \{EXT_\alpha(\mathfrak{X}, \wp_1, \vec{\alpha}(\wp_2)), EXT_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1), \wp_2)\}$ and quantifier prefixes $\wp_1, \wp_2 \in \mathbf{Q}_R$, with $AP(\mathfrak{X}) \cap AP(\wp_1, \wp_2) = \emptyset$.

Proof. Before continuing, we need to state the following property of the syntactic canonicalisations.

Claim 4.11. $\vec{\alpha}(\vec{\alpha}(\wp_1), \wp_2) = \vec{\alpha}(\wp_1, \wp_2) = \vec{\alpha}(\wp_1, \vec{\alpha}(\wp_2))$, for all $\wp_1, \wp_2 \in \mathbf{Q}_R$.

Remark 4.1. Strictly speaking, since $\vec{\alpha}(\wp_1)$ and $\vec{\alpha}(\wp_2)$ might not be in \mathbf{Q}_R , the two expressions $\vec{\alpha}(\vec{\alpha}(\wp_1), \wp_2)$ and $\vec{\alpha}(\wp_1, \vec{\alpha}(\wp_2))$ are not well-defined, if one consider the original definition of $\vec{\alpha}$. However, one can easily generalise the canonicalisations to prefixes where the existential (resp., universal) variables are not just realizable, but possibly strongly realizable w.r.t to some set of universal variables.

At this point, we can prove the four inequalities separately; all the inclusion in the following chains of equalities/inequalities are consequences of Proposition 4.3 and, if required, Lemma 2.9.

$$- [\mathbf{ext}_\alpha(\mathfrak{X}, \wp_1, \vec{\alpha}(\wp_2)) \sqsubseteq \mathbf{ext}_\alpha(\mathfrak{X}, \wp_1, \wp_2)]$$

$$\begin{aligned} EXT_\alpha(\mathfrak{X}, \wp_1, \vec{\alpha}(\wp_2)) &= EXT_\alpha(EXT_\alpha(\mathfrak{X}, \wp_1), \vec{\alpha}(\wp_2)) \sqsubseteq \\ &EXT_\alpha(EXT_\alpha(\mathfrak{X}, \wp_1), \wp_2) = EXT_\alpha(\mathfrak{X}, \wp_1, \wp_2). \end{aligned}$$

$$- [\mathbf{ext}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1), \wp_2) \sqsubseteq \mathbf{ext}_\alpha(\mathfrak{X}, \wp_1, \wp_2)]$$

$$\begin{aligned} EXT_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1), \wp_2) &= EXT_\alpha(EXT_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1)), \wp_2) \sqsubseteq \\ &EXT_\alpha(EXT_\alpha(\mathfrak{X}, \wp_1), \wp_2) = EXT_\alpha(\mathfrak{X}, \wp_1, \wp_2). \end{aligned}$$

$$- [\mathbf{ext}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1. \wp_2)) \sqsubseteq \mathbf{ext}_\alpha(\mathfrak{X}, \wp_1. \vec{\alpha}(\wp_2))]$$

$$\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1. \wp_2)) = \text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1. \vec{\alpha}(\wp_2))) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \wp_1. \vec{\alpha}(\wp_2))$$

$$- [\mathbf{ext}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1. \wp_2)) \sqsubseteq \mathbf{ext}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1). \wp_2)]$$

$$\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1. \wp_2)) = \text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\vec{\alpha}(\wp_1). \wp_2)) \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1). \wp_2) \quad \square$$

Proposition 4.11. $\text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1. \wp_2)) \sqsubseteq \mathfrak{Y} \sqsubseteq \text{EXT}_\alpha(\mathfrak{X}, \wp_1. \wp_2)$, for all hyperteam $\mathfrak{X} \in \text{HypTeams}$ and $\mathfrak{Y} \in \{\text{EXT}_\alpha(\mathfrak{X}, \wp_1. \vec{\alpha}(\wp_2)), \text{EXT}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1). \wp_2)\}$ and quantifier prefixes $\wp_1, \wp_2 \in \mathbf{Q}_R$, with $\text{AP}(\mathfrak{X}) \cap \text{AP}(\wp_1. \wp_2) = \emptyset$.

Proof. As we have done for the previous proposition, we prove the four inequalities separately; all the inclusion in the following chains of equalities/inequalities are consequences of Proposition 4.3 and, if required, Lemma 2.9.

$$- [\mathbf{ext}_\alpha(\mathfrak{X}, \wp_1. \wp_2) \sqsubseteq \mathbf{ext}_\alpha(\mathfrak{X}, \wp_1. \vec{\alpha}(\wp_2))]$$

$$\begin{aligned} \text{nevl}_\alpha(\mathfrak{X}, \wp_1. \wp_2) &= \text{nevl}_\alpha(\text{nevl}_\alpha(\mathfrak{X}, \wp_1), \wp_2) \sqsubseteq \\ \text{nevl}_\alpha(\text{nevl}_\alpha(\mathfrak{X}, \wp_1), \vec{\alpha}(\wp_2)) &= \text{nevl}_\alpha(\mathfrak{X}, \wp_1. \vec{\alpha}(\wp_2)). \end{aligned}$$

$$- [\mathbf{ext}_\alpha(\mathfrak{X}, \wp_1. \wp_2) \sqsubseteq \mathbf{ext}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1). \wp_2)]$$

$$\begin{aligned} \text{nevl}_\alpha(\mathfrak{X}, \wp_1. \wp_2) &= \text{nevl}_\alpha(\text{nevl}_\alpha(\mathfrak{X}, \wp_1), \wp_2) \sqsubseteq \\ \text{nevl}_\alpha(\text{nevl}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1)), \wp_2) &= \text{nevl}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1). \wp_2). \end{aligned}$$

$$- [\mathbf{ext}_\alpha(\mathfrak{X}, \wp_1. \vec{\alpha}(\wp_2)) \sqsubseteq \mathbf{ext}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1. \wp_2))]$$

$$\text{nevl}_\alpha(\mathfrak{X}, \wp_1. \vec{\alpha}(\wp_2)) \sqsubseteq \text{nevl}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1. \vec{\alpha}(\wp_2))) = \text{nevl}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1. \wp_2)).$$

$$- [\mathbf{ext}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1). \wp_2) \sqsubseteq \mathbf{ext}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1. \wp_2))]$$

$$\text{nevl}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1). \wp_2) \sqsubseteq \text{nevl}_\alpha(\mathfrak{X}, \vec{\alpha}(\vec{\alpha}(\wp_1). \wp_2)) = \text{nevl}_\alpha(\mathfrak{X}, \vec{\alpha}(\wp_1. \wp_2)). \quad \square$$

For the next proposition, given a set of assignments Y and a set of atomic propositions $P \subseteq s$, we introduce the notation $Y \setminus_P \stackrel{\text{def}}{=} \{\chi \in \text{Asg}(\text{AP}(Y) \setminus P) \mid \exists \chi' \in Y. \chi \subseteq \chi'\}$. We also use the notation $Y \setminus_p$, with $p \in \text{AP}(Y)$, as a shortcut for $Y \setminus_{\{p\}}$.

Proposition 4.12. Let $\mathfrak{X} \in \text{HypTeams}(P)$ be a hyperteam over $P \subseteq s$ and $\wp \in \mathbf{Q}$ a

quantifier prefix, with $AP(\wp) \cap P = \emptyset$. Then, for all sets of assignments $Y \in \text{EXT}_\alpha(\mathfrak{X}, \wp)$, it holds that $Y \setminus_{AP(\wp)} \subseteq \cup \mathfrak{X}$.

Proof. The proof proceeds by induction on the length of the quantification prefix \wp .

- [**Base case** $\wp = \varepsilon$] $\text{EXT}_\alpha(\mathfrak{X}, \varepsilon) = \mathfrak{X}$, thus, the property follows trivially.
- [**Base case** $\wp = Q^\ominus p$ **with** Q α -**coherent**] Since $\text{EXT}_\alpha(\mathfrak{X}, \wp) = \text{EXT}_\Theta(\mathfrak{X}, p)$, there exist $X \in \mathfrak{X}$ and $F \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}))$ such that $Y = \text{EXT}(X, F, p)$. Thus, $Y \setminus_p = X \subseteq \cup \mathfrak{X}$, hence the thesis.
- [**Base case** $\wp = Q^\ominus p$ **with** Q **not** α -**coherent**] In this case, we have $\text{EXT}_\alpha(\mathfrak{X}, Q^\ominus p) = \overline{\text{EXT}_\Theta(\mathfrak{X}, p)}$. Let $Y \in \overline{\text{EXT}_\Theta(\mathfrak{X}, p)}$. By definition of dualization, there is $\Gamma \in \text{Chc}(\text{EXT}_\Theta(\overline{\mathfrak{X}}, p))$ such that $\text{img}(\Gamma) = Y$. Then, for every $F \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}))$ and every $X \in \overline{\mathfrak{X}}$, there is $\chi_{X,F} \in \text{EXT}(X, F, p)$ such that $Y = \{\chi_{X,F} \mid X \in \overline{\mathfrak{X}} \wedge F \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}))\}$. Then for every $\chi_{X,F}$, there is $\chi'_{X,F} \in X$ such that $\chi_{X,F} = \chi'_{X,F}[p \mapsto F(\chi'_{X,F})]$. Naturally, $Y \setminus_{AP(\wp)} = \{\chi'_{X,F} \mid X \in \overline{\mathfrak{X}} \wedge F \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}))\}$. However, $X \in \overline{\mathfrak{X}}$ and then $\chi'_{X,F} \in \cup \mathfrak{X}$. Hence $Y \setminus_{AP(\wp)} \subseteq \cup \mathfrak{X}$.
- [**Inductive case** $\wp = \wp'.Q^\ominus p$] By the inductive hypothesis, we have that $Z \setminus_{AP(\wp')} \subseteq \cup \mathfrak{X}$, for all $Z \in \text{EXT}_\alpha(\mathfrak{X}, \wp')$. Consequently, $(\cup \text{EXT}_\alpha(\mathfrak{X}, \wp')) \setminus_{AP(\wp')} = \cup (\text{EXT}_\alpha(\mathfrak{X}, \wp') \setminus_{AP(\wp')}) \subseteq \cup \mathfrak{X}$. Now, by definition of evolution function, we have that $\text{EXT}_\alpha(\mathfrak{X}, \wp) = \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp'), Q^\ominus p)$. Again by the inductive hypothesis, $Y \setminus_p \subseteq \cup \text{EXT}_\alpha(\mathfrak{X}, \wp')$, since $Y \in \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}, \wp'), Q^\ominus p)$. Hence, $Y \setminus_{AP(\wp)} = (Y \setminus_p) \setminus_{AP(\wp')} \subseteq (\cup \text{EXT}_\alpha(\mathfrak{X}, \wp')) \setminus_{AP(\wp')} \subseteq \cup \mathfrak{X}$, as expected. \square

We now define a refinement of the order \sqsubseteq between two hyperteams $\mathfrak{X}_1, \mathfrak{X}_2 \in \text{HypTeams}$, with $AP(\mathfrak{X}_1) = AP(\mathfrak{X}_2)$, *w.r.t* a set of assignments $X \subseteq \text{Asg}(P)$ over some $P \subseteq s$ as follows: $\mathfrak{X}_1 \sqsubseteq_X \mathfrak{X}_2$ if, for every $X_1 \in \mathfrak{X}_1$, there is $X_2 \in \mathfrak{X}_2$ such that $X_2 \setminus \{\chi \in \text{Asg} \mid \chi|_P \in X\} \subseteq X_1$.

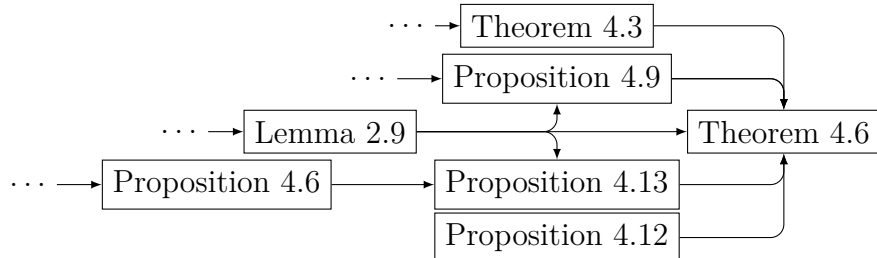
Proposition 4.13. *Let $\mathfrak{X}_1, \mathfrak{X}_2 \in \text{HypTeams}$ be two hyperteams with $\mathfrak{X}_1 \sqsubseteq_X \mathfrak{X}_2$, for some set of assignments $X \subseteq \text{Asg}(P)$ over a set of atomic propositions $P \subseteq s$. Then, the following hold true: $\text{EXT}_\alpha(\mathfrak{X}_1, \wp) \sqsubseteq_X \text{EXT}_\alpha(\mathfrak{X}_2, \wp)$, for all $\wp \in \mathbf{Q}$ with $AP(\mathfrak{X}_1) \cap AP(\wp) = AP(\mathfrak{X}_2) \cap AP(\wp) = \emptyset$.*

Proof. The proof proceeds by induction on the length of \wp .

- [**Base step** $\wp = \varepsilon$] $\text{EXT}_\alpha(\mathfrak{X}_1, \varepsilon) = \mathfrak{X}_1 \sqsubseteq_X \mathfrak{X}_2 = \text{EXT}_\alpha(\mathfrak{X}_2, \varepsilon)$.
- [**Inductive step** $\wp = Q^\ominus p. \wp'$] Let us distinguish two cases based on whether Q is or not α -coherent.

- [**Q is α -coherent**] Since α and Q are coherent, $\text{EXT}_\alpha(\mathfrak{X}_i, \wp) = \text{EXT}_\alpha(\text{EXT}_\Theta(\mathfrak{X}_i, p), \wp')$, for all $i \in \{1, 2\}$. We can now focus on showing that $\text{EXT}_\Theta(\mathfrak{X}_1, p) \sqsubseteq_X \text{EXT}_\Theta(\mathfrak{X}_2, p)$ holds true, as the thesis follows by applying the inductive hypothesis. Since $\text{EXT}_\Theta(\mathfrak{X}_i, p) = \{\text{EXT}(X_i, F_i, p) \mid X_i \in \mathfrak{X}_i, F_i \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_i))\}$, we have to prove that, for every $X_1 \in \mathfrak{X}_1$ and $F_1 \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_1))$, there exist $X_2 \in \mathfrak{X}_2$ and $F_2 \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_2))$ such that $\text{EXT}(X_2, F_2, p) \setminus \{\chi \in \text{Asg} \mid \chi|_P \in X\} \subseteq \text{EXT}(X_1, F_1, p)$. Now, it is easy to see that such a property can be satisfied by choosing $F_2 \stackrel{\text{def}}{=} F_1$, since $\text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_1)) = \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_2))$, and $\mathfrak{X}_2 \stackrel{\text{def}}{=} F(\mathfrak{X}_1)$, where $F: \mathfrak{X}_1 \rightarrow \mathfrak{X}_2$ is a witness for the inclusion $\mathfrak{X}_1 \sqsubseteq_X \mathfrak{X}_2$.
- [**Q is not α -coherent**] Since α and Q are not coherent, by Lemma 2.9 and 4.6, it holds that $\text{EXT}_\alpha(\mathfrak{X}_i, \wp) = \text{EXT}_\alpha(\text{EXT}_\alpha(\mathfrak{X}_i, Q^\ominus p), \wp') \equiv \text{EXT}_\alpha(\text{nevl}_\alpha(\mathfrak{X}_i, Q^\ominus p), \wp')$, for all $i \in \{1, 2\}$. As done in the previous case, we can now focus on showing that $\text{nevl}_\alpha(\mathfrak{X}_1, Q^\ominus p) \sqsubseteq_X \text{nevl}_\alpha(\mathfrak{X}_2, Q^\ominus p)$ holds true, as the thesis follows by applying the inductive hypothesis. Since $\text{nevl}_\alpha(\mathfrak{X}_i, Q^\ominus p) = \{\text{EXT}(\tilde{\mathfrak{d}}_i, p) \mid \tilde{\mathfrak{d}}_i \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_i)) \rightarrow \mathfrak{X}_i\}$, we have to prove that, for every $\tilde{\mathfrak{d}}_1 \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_1)) \rightarrow \mathfrak{X}_1$, there exists $\tilde{\mathfrak{d}}_2 \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_2)) \rightarrow \mathfrak{X}_2$ such that $\text{EXT}(\tilde{\mathfrak{d}}_2, p) \setminus \{\chi \in \text{Asg} \mid \chi|_P \in X\} \subseteq \text{EXT}(\tilde{\mathfrak{d}}_1, p)$. To this end, let us define a function $g: (\text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_1)) \rightarrow \mathfrak{X}_1) \rightarrow (\text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_2)) \rightarrow \mathfrak{X}_2)$ as follows: $g(\tilde{\mathfrak{d}}_1)(F) \stackrel{\text{def}}{=} F(\tilde{\mathfrak{d}}_1(F))$, for every $\tilde{\mathfrak{d}}_1 \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_1)) \rightarrow \mathfrak{X}_1$ and $F \in \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_1)) = \text{Rsp}_\Theta(\text{AP}(\mathfrak{X}_2))$, where $F: \mathfrak{X}_1 \rightarrow \mathfrak{X}_2$ is a witness for the inclusion $\mathfrak{X}_1 \sqsubseteq_X \mathfrak{X}_2$. Clearly, it holds that $g(\tilde{\mathfrak{d}}_1)(F) \setminus \{\chi \in \text{Asg} \mid \chi|_P \in X\} \subseteq \tilde{\mathfrak{d}}_1(F)$. Thus, the required property can be satisfied by choosing $\tilde{\mathfrak{d}}_2 \stackrel{\text{def}}{=} g(\tilde{\mathfrak{d}}_1)$, since $\text{EXT}(g(\tilde{\mathfrak{d}}_1), p) \setminus \{\chi \in \text{Asg} \mid \chi|_P \in X\} \subseteq \text{EXT}(\tilde{\mathfrak{d}}_1, p)$ holds true. \square

Next, we prove Theorem 4.6. Here is the graph of dependency presenting the theorem and the propositions used for this proof.



Theorem 4.6 (Quantification Game II). *Every \mathcal{S} -game \mathcal{G} , for some quantification-game schema $\mathcal{S} \stackrel{\text{def}}{=} \langle \mathfrak{X}, \wp, \Psi \rangle$, satisfies the following two properties:*

1. if Eloise wins then $E \subseteq \Psi$, for some $E \in \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\forall}\exists(\varphi))$;
2. if Abelard wins then $E \not\subseteq \Psi$, for all $E \in \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\exists}\forall(\varphi))$.

Proof. First of all, recall that the game \mathcal{G}_S of Construction 4.2 is obtained directly from the game $\mathcal{G}_{\hat{\varphi}}^{\hat{\Psi}}$ of Construction 4.1, by defining the set of assignments $\hat{\varphi}$ and the quantifier prefix $\hat{\Psi}$ as follows:

- $\hat{\varphi} \stackrel{\text{def}}{=} \forall(\vec{p}). \tilde{\varphi}. \varphi$ and
- $\hat{\Psi} \stackrel{\text{def}}{=} \Psi \cup \{\chi \in \text{Asg}(\text{P}) \mid \chi_{\uparrow(\vec{p})} \notin X\}$,

with $(\vec{p}) \stackrel{\text{def}}{=} \text{AP}(\mathfrak{X}) \setminus \text{AP}(\tilde{\varphi})$ and $\text{P} \stackrel{\text{def}}{=} \text{AP}(\varphi) \cup \text{AP}(\mathfrak{X})$.

We can now proceed with the proof of the two properties.

- [1] If Eloise wins the game, by Item 1 of Theorem 4.3, there exists a set of assignments $\hat{E} \in \text{EXT}_{\exists\forall}(\vec{\forall}\exists(\hat{\varphi}))$ such that $\hat{E} \subseteq \hat{\Psi}$. Thanks to Lemma 2.9 and 4.9, we can show that $\text{EXT}_{\exists\forall}(\vec{\forall}\exists(\hat{\varphi})) \sqsubseteq \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\forall}\exists(\varphi))$. Indeed,

$$\text{EXT}_{\exists\forall}(\vec{\forall}\exists(\hat{\varphi})) = \text{EXT}_{\exists\forall}(\vec{\forall}\exists(\forall(\vec{p}). \tilde{\varphi}. \varphi)) \quad (4.8a)$$

$$\sqsubseteq \text{EXT}_{\exists\forall}(\forall(\vec{p}). \tilde{\varphi}. \vec{\forall}\exists(\varphi)) \quad (4.8b)$$

$$= \text{EXT}_{\exists\forall}(\text{EXT}_{\exists\forall}(\text{EXT}_{\exists\forall}(\forall(\vec{p})), \tilde{\varphi}), \vec{\forall}\exists(\varphi)) \quad (4.8c)$$

$$= \text{EXT}_{\exists\forall}(\text{EXT}_{\exists\forall}(\{\text{Asg}((\vec{p}))\}, \tilde{\varphi}), \vec{\forall}\exists(\varphi)) \quad (4.8d)$$

$$\sqsubseteq \text{EXT}_{\exists\forall}(\text{EXT}_{\exists\forall}(\{X\}, \tilde{\varphi}), \vec{\forall}\exists(\varphi)) \quad (4.8e)$$

$$= \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\forall}\exists(\varphi)), \quad (4.8f)$$

where Step 4.8b is due to Proposition 4.9, Step 4.8d to the equality $\text{EXT}_{\exists\forall}(\forall(\vec{p})) = \{\text{Asg}((\vec{p}))\}$, and Step 4.8e is derived from Lemma 2.9, thanks to the fact that $\{\text{Asg}((\vec{p}))\} \sqsubseteq \{X\}$. Now, due to the definition of the ordering \sqsubseteq between hyper-teams, it follows that $\text{EXT}_{\exists\forall}(\vec{\forall}\exists(\hat{\varphi})) \sqsubseteq \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\forall}\exists(\varphi))$ necessarily implies the existence of a set of assignments $E \in \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\forall}\exists(\varphi))$ such that $E \subseteq \hat{E}$. Therefore, $E \subseteq \hat{\Psi}$. At this point, we can prove that $E \subseteq \Psi$, since $\hat{\Psi} = \Psi \cup \{\chi \in \text{Asg}(\text{P}) \mid \chi_{\uparrow(\vec{p})} \notin X\}$ and $E \cap \{\chi \in \text{Asg}(\text{P}) \mid \chi_{\uparrow(\vec{p})} \notin X\} = \emptyset$. Indeed, $E \in \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\forall}\exists(\varphi)) = \text{EXT}_{\exists\forall}(\{X\}, \tilde{\varphi}. \vec{\forall}\exists(\varphi))$ and, by Proposition 4.12, it follows that $E \setminus_{(\vec{p})} \subseteq X$.

- [2] If Abelard wins the game, by Item 2 of Theorem 4.3, it holds that $\hat{E} \not\subseteq \hat{\Psi}$, for all sets of assignments $\hat{E} \in \text{EXT}_{\exists\forall}(\vec{\exists}\forall(\hat{\varphi}))$. It is easy to observe that $\{X\} \sqsubseteq_{\bar{X}} \{\text{Asg}((\vec{p}))\}$, since $\text{Asg}((\vec{p})) \setminus \{\chi \in \text{Asg} \mid \chi_{\uparrow(\vec{p})} \in \bar{X}\} = \text{Asg}((\vec{p})) \setminus \{\chi \in \text{Asg} \mid \chi_{\uparrow(\vec{p})} \notin X\} = X$. Thus, thanks to Proposition 4.9 and 4.13, we can show that

$\text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\exists\forall}(\varphi)) \sqsubseteq_{\overline{X}} \text{EXT}_{\exists\forall}(\vec{\exists\forall}(\hat{\varphi}))$. Indeed,

$$\text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\exists\forall}(\varphi)) = \text{EXT}_{\exists\forall}(\text{EXT}_{\exists\forall}(\{X\}, \tilde{\varphi}), \vec{\exists\forall}(\varphi)) \quad (4.9a)$$

$$\sqsubseteq_{\overline{X}} \text{EXT}_{\exists\forall}(\text{EXT}_{\exists\forall}(\{\text{Asg}((\vec{p})\}), \tilde{\varphi}), \vec{\exists\forall}(\varphi)) \quad (4.9b)$$

$$= \text{EXT}_{\exists\forall}(\text{EXT}_{\exists\forall}(\text{EXT}_{\exists\forall}(\forall(\vec{p})), \tilde{\varphi}), \vec{\exists\forall}(\varphi)) \quad (4.9c)$$

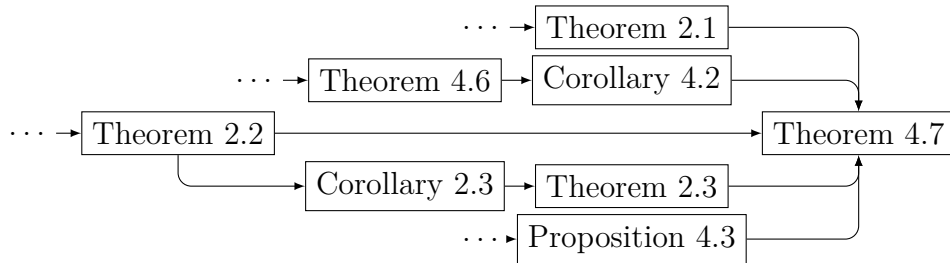
$$= \text{EXT}_{\exists\forall}(\forall(\vec{p}) \cdot \tilde{\varphi} \cdot \vec{\exists\forall}(\varphi)) \quad (4.9d)$$

$$\sqsubseteq \text{EXT}_{\exists\forall}(\vec{\exists\forall}(\forall(\vec{p}) \cdot \tilde{\varphi} \cdot \varphi)) \quad (4.9e)$$

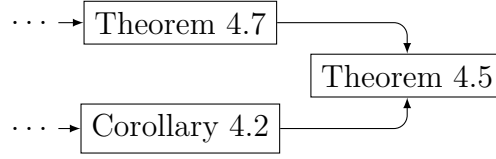
$$= \text{EXT}_{\exists\forall}(\vec{\exists\forall}(\hat{\varphi})), \quad (4.9f)$$

where Step 4.9b is due to Proposition 4.13, thanks to the fact that $\{X\} \sqsubseteq_{\overline{X}} \{\text{Asg}((\vec{p})\})$, Step 4.9c to the equality $\text{EXT}_{\exists\forall}(\forall(\vec{p})) = \{\text{Asg}((\vec{p})\})$, and Step 4.9e is derived from Proposition 4.9. Now, due to the definition of the ordering $\sqsubseteq_{\overline{X}}$ between hyperteam, it follows that $\text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\exists\forall}(\varphi)) \sqsubseteq_{\overline{X}} \text{EXT}_{\exists\forall}(\vec{\exists\forall}(\hat{\varphi}))$ necessarily implies the non existence of a set of assignments $E \in \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\exists\forall}(\varphi))$ such that $E \subseteq \hat{\Psi}$. Indeed, assume towards a contradiction that there is $E \in \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\exists\forall}(\varphi))$ such that $E \subseteq \hat{\Psi}$. By the above inclusion, there is $\hat{E} \in \text{EXT}_{\exists\forall}(\vec{\exists\forall}(\hat{\varphi}))$ such that $\hat{E} \setminus \{\chi \in \text{Asg} \mid \chi_{\upharpoonright(\vec{p})} \in \overline{X}\} \subseteq E \subseteq \hat{\Psi}$. Since $\hat{E} \cap \{\chi \in \text{Asg} \mid \chi_{\upharpoonright(\vec{p})} \in \overline{X}\} \subseteq \{\chi \in \text{Asg}(\text{P}) \mid \chi_{\upharpoonright(\vec{p})} \notin X\} \subseteq \hat{\Psi}$, we have that $\hat{E} \subseteq \hat{\Psi}$, which contradicts the fact that Abelard wins the game. Hence, $E \not\subseteq \hat{\Psi}$ holds, for all $E \in \text{EXT}_{\exists\forall}(\mathfrak{X}, \vec{\exists\forall}(\varphi))$, which implies that $E \not\subseteq \Psi$, being $\Psi \subseteq \hat{\Psi}$. \square

Now, we have proven everything that is used for the proof of Theorem 4.7 from the main paper. Here is the graph of dependency presenting the lemma, the propositions, the corollaries and theorems used for this proof.



And finally, we have proven everything that is used for the proof of Theorem 4.5 from the main paper. Here is the graph of dependency presenting the corollary and the theorem used for this proof.



The following theorem relies on both the notions of parity game and parity automaton [Mos84; EJ91; Mos91] (see also [KN01; GTW02]). Parity games are perfect-information two-player turn-based games of infinite duration, usually played on finite directed graphs. Their vertices, called positions, are labelled by natural numbers, called priorities, and are assigned to one of two players, namely 0 and 1. The game starts at a given position and, during its evolution, players can take a move (an outgoing edge) only at their own positions. The moves selected by the players induce an infinite sequence of vertices, called play. If the maximal priority of the vertices occurring infinitely often in the play is even, then the play is winning for player 0, otherwise, player 1 takes it all. Similarly, the states of a (non-deterministic) parity automaton are labelled with natural numbers (priorities) and an infinite word given in input is accepted by the automaton iff there exists a run induced by such a word, for which the maximal priority seen infinitely often along it has even parity.

Theorem 4.8 (Satisfiability Game). *For every realizable GFG-QPTL sentence Φ there is a parity game, with $2^{2^{|\Phi|}}$ positions and $2^{|\Phi|}$ priorities, won by Eloise iff Φ is satisfiable.*

Proof. Let $\varphi = \wp\psi$ be a realizable GFG-QPTL sentence with \wp a quantification prefix and ψ an LTL formula. Additionally, let $\Psi \stackrel{\text{def}}{=} \{\chi \in \text{Asg}(\text{AP}(\wp)) \mid \chi \models \psi\}$. The idea of the proof is to construct a parity game \mathcal{G}_φ that is equivalent to the game $\mathcal{G}_\varphi^\psi \stackrel{\text{def}}{=} \mathcal{G}_\varphi^\Psi \stackrel{\text{def}}{=} \langle A_\varphi^\psi, O_\varphi^\psi, \text{Win}_\varphi^\psi \rangle$ defined in Construction 4.1, where $A_\varphi^\psi \stackrel{\text{def}}{=} \langle P^{\exists, \wp, \psi}, P^{\forall, \wp, \psi}, p^{\wp, \psi}, M^{\wp, \psi} \rangle$. Intuitively, \mathcal{G}_φ simulates the synchronous product of arena A_φ^ψ with the deterministic automaton \mathcal{A}_ψ recognising models of ψ , where \mathcal{A}_ψ changes state only when Abelard takes a move starting from an observable position containing full valuation of the propositions. Such valuation determines the successor state.

The deterministic automaton \mathcal{A}_ψ recognising models of ψ can be obtained in a standard way, by first constructing a non-deterministic Büchi automaton \mathcal{A}_ψ that recognises models of ψ , using the Vardi-Wolper construction [VW86a], and then by determinising \mathcal{A}_ψ (via a Safra-like determinisation procedure [Pit06]) into an equivalent deterministic parity automaton $\mathcal{A} = \langle Q, q_0, \Sigma, \delta, \text{par} \rangle$, where

- Q is the finite set of states,

- $q_0 \in Q$ is the initial state,
- $\Sigma = \text{Val}(\text{AP}(\wp))$ is the alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function,
- $\text{par} : Q \rightarrow \mathbb{N}$ is the parity condition.

Now, the parity game \mathcal{G}_φ associated with φ is a pair $\mathcal{G}_\varphi \stackrel{\text{def}}{=} \langle A_\varphi, \text{Win}_\varphi \rangle$, where:

- $A_\varphi \stackrel{\text{def}}{=} \langle P^{\exists,\varphi}, P^{\forall,\varphi}, p^\varphi, M^\varphi \rangle$ is the arena;
- the set of positions $P_\mathcal{G}^\varphi \stackrel{\text{def}}{=} P^{\exists,\varphi} \uplus P^{\forall,\varphi} = Q \times (P^{\exists,\wp,\psi} \cup P^{\forall,\wp,\psi})$ contains exactly the pairs consisting of a state of the automaton \mathcal{A} and a valuation $\nu \in \text{Val}$ which is a position of \mathcal{G}_\wp^ψ ;
- the set of Eloise's positions $P^{\exists,\varphi} \subseteq P_\mathcal{G}^\varphi$ only contains the positions $(q, \nu) \in P_\mathcal{G}^\varphi$ where ν is an Eloise's position in \mathcal{G}_\wp^ψ ;
- the initial position $p^\varphi \stackrel{\text{def}}{=} (q_0, \emptyset)$ is just the initial state of \mathcal{A} paired with the initial state of \mathcal{G}_\wp^ψ ;
- the move relation $M^\varphi \subseteq P_\mathcal{G}^\varphi \times P_\mathcal{G}^\varphi$ contains exactly those pairs of positions $((q_1, \nu_1), (q_2, \nu_2)) \in P_\mathcal{G}^\varphi \times P_\mathcal{G}^\varphi$ such that:
 - (ν_1, ν_2) is a move in \mathcal{G}_\wp^ψ ;
 - if $\nu_2 = \emptyset$ then $q_2 = \delta(q_1, \nu_1)$, otherwise, $q_1 = q_2$;
- the winning condition Win_φ is deduced from the accepting condition of the automaton \mathcal{A} . More precisely, the priority of a position $(q, \nu) \in P_\mathcal{G}^\varphi$ is defined as the priority $\text{par}(q)$ of q , i.e., $\text{Win}_\varphi((q, \nu)) = \text{par}(q)$ for all $(q, \nu) \in P_\mathcal{G}^\varphi$.

We want to show that there is a strategy for Eloise to win \mathcal{G}_φ if and only if there is a strategy for her to win \mathcal{G}_\wp^ψ .

Towards the definition of a correspondence between Eloise's strategies in \mathcal{G}_φ and Eloise's strategies in \mathcal{G}_\wp^ψ , we define now a bijection PATH2PATH between initial paths on \mathcal{G}_φ (denoted $\text{Paths}_\varphi^{\text{init}}$) and initial paths on \mathcal{G}_\wp^ψ (denoted $\text{Paths}_\wp^{\text{init}}$). Given two sets S, S' and a pair $(x, y) \in S \times S'$, we let $\pi_1(x, y) = x$ and $\pi_2(x, y) = y$, that is, functions π_1 and π_2 return the first and the second element of their argument, respectively. Furthermore, we denote by $\tau \odot \pi \stackrel{\text{def}}{=} ((\tau)_0, (\pi)_0)((\tau)_1, (\pi)_1) \dots$ the *pairing product* of two sequences. Let $\pi \in \text{Paths}_\varphi^{\text{init}}$, with π , be an initial path on \mathcal{G}_φ . Function f maps π into the initial path on \mathcal{G}_\wp^ψ obtained by projecting on the second component of each position of π , that is, $f(\pi) = \langle \pi_2((\pi)_i) \rangle_{i \in \llbracket 0, \dots, |\pi| \rrbracket}$. The fact that f is a bijection, as stated in Corollary 4.3, is an immediate consequence of the following claim.

Claim 4.12. *For every initial path $\pi \in \text{Paths}_\varphi^{\text{init}}$ there is exactly one sequence of automaton states $\tau \in Q^\infty$ such that $|\tau| = |\pi|$ and $\tau \odot \pi \in \text{Paths}_\varphi^{\text{init}}$.*

Proof. The claim follows from the fact that, according to the definition of M^φ , the first component of each position of a path on \mathcal{G}_φ is univocally determined by the second component of that position and by the previous position in the path (the fact that \mathcal{A} is deterministic plays an important role in this). More formally, τ is constructed inductively as: $(\tau)_0 = q_0$ is the initial state of \mathcal{A} and, for $i \in \mathbb{N}$, with $i > 0$:

$$(\tau)_i = \begin{cases} (\tau)_{i-1} & \text{if } (\pi)_i \neq \emptyset \\ \delta((\tau)_{i-1}, (\pi)_{i-1}) & \text{if } (\pi)_i = \emptyset \end{cases}$$

Clearly, $\tau \odot \pi \in \mathbf{Paths}_\varphi^{\text{init}}$ since π is a path on \mathcal{G}_φ^ψ and τ closely follow the move relation M^φ . It is also easy to see that for any other $\tau' \in Q^\infty$, with $\tau' \neq \tau$, it holds that $\tau' \odot \pi \notin \mathbf{Paths}_\varphi^{\text{init}}$. Indeed, assume, towards a contradiction, that $\tau' \odot \pi \in \mathbf{Paths}_\varphi^{\text{init}}$, and let i be the smallest index such that $(\tau)_i \neq (\tau')_i$. If $i = 0$, then $((\tau')_i, (\pi)_i)$ is not the initial position of \mathcal{G}_φ^ψ , thus contradicting the assumption; if $i > 0$ and $(\pi)_i \neq \emptyset$, we have: $(\tau)_i = (\tau)_{i-1} = (\tau')_{i-1}$, which implies $(\tau')_{i-1} \neq (\tau')_i$, and thus $((\tau')_{i-1}, (\pi)_{i-1}), ((\tau')_i, (\pi)_i)$ is not a move of \mathcal{G}_φ , according M^φ , and the assumption is contradicted; finally, if $i > 0$ and $(\pi)_i = \emptyset$, we have $(\tau)_i = \delta((\tau)_{i-1}, (\pi)_{i-1}) = \delta((\tau')_{i-1}, (\pi)_{i-1})$, which implies $(\tau')_i \neq \delta((\tau')_{i-1}, (\pi)_{i-1})$, and the assumption is contradicted once again, since $((\tau')_{i-1}, (\pi)_{i-1}), ((\tau')_i, (\pi)_i)$ is not a move of \mathcal{G}_φ for any $(\tau')_i \neq \delta((\tau')_{i-1}, (\pi)_{i-1})$, according M^φ . \square

Corollary 4.3. *Function $\text{PATH2PATH} : \mathbf{Paths}_\varphi^{\text{init}} \rightarrow \mathbf{Paths}_\varphi^{\text{init}}$ is a bijection.*

We define now a bijection κ from strategies (for both Eloise (\exists) and Abelard (\forall)) in \mathcal{G}_φ to strategies in \mathcal{G}_φ^ψ . For $\alpha \in \{\exists, \forall\}$, let $\mathbf{Hist}_\alpha(\mathcal{G}_\varphi)$ and $\mathbf{Hist}_\alpha(\mathcal{G}_\varphi^\psi)$ be the sets of histories for α (i.e., the sets of finite initial paths terminating in an α -position) in \mathcal{G}_φ and \mathcal{G}_φ^ψ , respectively, and let $\mathbf{Strat}_\alpha(\mathcal{G}_\varphi)$ and $\mathbf{Strat}_\alpha(\mathcal{G}_\varphi^\psi)$ be the sets of strategies for player α in \mathcal{G}_φ and \mathcal{G}_φ^ψ , respectively. Observe that $\mathbf{Hist}_\alpha(\mathcal{G}_\varphi) \subseteq \mathbf{Paths}_\varphi^{\text{init}}$ and $\mathbf{Hist}_\alpha(\mathcal{G}_\varphi^\psi) \subseteq \mathbf{Paths}_\varphi^{\text{init}}$. We define $\kappa : \mathbf{Strat}_\alpha(\mathcal{G}_\varphi) \rightarrow \mathbf{Strat}_\alpha(\mathcal{G}_\varphi^\psi)$ as follows: for every $\sigma \in \mathbf{Strat}_\alpha(\mathcal{G}_\varphi)$ and every history $h \in \mathbf{Hist}_\alpha(\mathcal{G}_\varphi^\psi)$, we set $\kappa(\sigma)(h) = \pi_2(\sigma(f^{-1}(h)))$. Intuitively, $\kappa(\sigma)$ acts like σ restricted to the second component of positions.

Claim 4.13. *Function $\kappa : \mathbf{Strat}_\alpha(\mathcal{G}_\varphi) \rightarrow \mathbf{Strat}_\alpha(\mathcal{G}_\varphi^\psi)$ is a bijection.*

Proof. In order to see that κ is injective, we show that $\sigma \neq \sigma'$ implies $\kappa(\sigma) \neq \kappa(\sigma')$, for every $\sigma, \sigma' \in \mathbf{Strat}_\alpha(\mathcal{G}_\varphi)$. Let $\sigma, \sigma' \in \mathbf{Strat}_\alpha(\mathcal{G}_\varphi)$ and let $h \in \mathbf{Hist}_\alpha(\mathcal{G}_\varphi)$ be such that $\sigma(h) \neq \sigma'(h)$. We first prove that $\pi_2(\sigma(h)) \neq \pi_2(\sigma'(h))$. Let $h = h(q, \nu)$ with h potentially empty. Let $\sigma(h) = (q^*, \nu^*)$ with $q^* = q$ if $\nu^* \neq \emptyset$ and $q^* = \delta(q, \nu)$ otherwise. Toward contradiction, suppose that $\pi_2(\sigma(h)) = \pi_2(\sigma'(h)) = \nu^*$. Then, by definition, $\pi_1(\sigma(h)) = \pi_1(\sigma'(h)) = q^*$

and then $\sigma(h) = \sigma(h')$ which is a contradiction. Then, we have $\kappa(\sigma)(f(h)) = \pi_2(\sigma(h)) \neq \pi_2(\sigma'(h)) = \kappa(\sigma')(f(h))$, and therefore $\kappa(\sigma) \neq \kappa(\sigma')$.

In order to show that κ is surjective as well, let $\sigma \in \mathbf{Strat}_\alpha(\mathcal{G}_\varphi^\psi)$. We build a strategy $\sigma' \in \mathbf{Strat}_\alpha(\mathcal{G}_\varphi)$ such that $\kappa(\sigma') = \sigma$. Intuitively, σ' returns a pair (a position in \mathcal{G}_φ) whose second component is chosen according to the output of strategy σ in \mathcal{G}_φ^ψ , and whose first component is univocally determined (thanks to Claim 4.12) by the choice of the second component and the argument history. Formally, for every $h \in \mathbf{Hist}_\alpha(\mathcal{G}_\varphi^\psi)$ we denote by $\text{ext}(h)$ the initial path of \mathcal{G}_φ^ψ obtained by appending to h the output the strategy σ on h itself, i.e., $\text{ext}(h) = h \cdot \sigma(h)$; notice that $f^{-1}(\text{ext}(h)) \in \mathbf{Paths}_\varphi^{\text{init}}$. Thus, we define σ' as: $\sigma'(h) \stackrel{\text{def}}{=} \text{lst}(f^{-1}(\text{ext}(f(h))))$, for every history $h \in \mathbf{Hist}_\alpha(\mathcal{G}_\varphi)$. It is not difficult to see that $\kappa(\sigma') = \sigma$: indeed, it holds that $\kappa(\sigma')(h) = \pi_2(\sigma'(f^{-1}(h))) = \pi_2(\text{lst}(f^{-1}(\text{ext}(h)))) = \pi_2(\text{lst}(f^{-1}(h \cdot \sigma(h)))) = \sigma(h)$, for every $h \in \mathbf{Hist}_\alpha(\mathcal{G}_\varphi^\psi)$. This concludes the proof. \square

The next claim states that the bijection κ preserves the possible plays resulting from the application of a strategy by Eloise in \mathcal{G}_φ and its image in \mathcal{G}_φ^ψ , modulo the correspondence between plays of \mathcal{G}_φ and \mathcal{G}_φ^ψ established by the bijection f . Let \mathbf{Plays}_φ be the set of plays of \mathcal{G}_φ .

Claim 4.14. *π is compatible with σ iff $f(\pi)$ is compatible with $\kappa(\sigma)$, for every $\sigma \in \mathbf{Strat}_\exists(\mathcal{G}_\varphi)$ and $\pi \in \mathbf{Plays}_\varphi$.*

Proof. It is easy to verify that a play $\pi \in \mathbf{Plays}_\varphi$ is compatible with a pair of strategies $(\sigma_\exists, \sigma_\forall) \in \mathbf{Strat}_\exists(\mathcal{G}_\varphi) \times \mathbf{Strat}_\forall(\mathcal{G}_\varphi)$ if and only if $f(\pi)$ is compatible with $(\kappa(\sigma_\exists), \kappa(\sigma_\forall))$. The thesis immediately follows. \square

As a final ingredient in our proof, we establish a correspondence $\mathbf{PLAY2ASG}$ between plays of \mathcal{G}_φ that are won by Eloise and models of ψ , recognised by \mathcal{A} . Function $\mathbf{PLAY2ASG} : \mathbf{Plays}_\varphi \rightarrow \mathbf{Val}(\mathbf{AP}(\varphi))^\omega$ is defined as: $\mathbf{PLAY2ASG}(\pi) \stackrel{\text{def}}{=} \mathbf{OBS}(\mathbf{PATH2PATH}(\rho))$ for every $\pi \in \mathbf{Plays}_\varphi$. The correctness of such a correspondence is stated in the next claim.

Claim 4.15. *ρ is won by Eloise in \mathcal{G}_φ iff $\mathbf{PLAY2ASG}(\rho)$ is recognised by \mathcal{A} , for every $\rho \in \mathbf{Plays}_\varphi$.*

Proof. By the definition of \mathcal{G}_φ , if we restrict a play $\pi \in \mathbf{Plays}_\varphi$ to those position $(q, \nu) \in \mathbf{P}_\mathcal{S}^\varphi$ where $\nu \in \mathbf{Val}(\mathbf{AP}(\varphi))$ (thus discharging partial valuations, which do not assign a truth value to all propositions occurring in ψ), we obtain a sequence π' that encodes to the unique run of \mathcal{A} on $f(\pi)$, where the sequence $(\pi')_{|1}$ of first components of each position (i.e., $(\pi')_{|1} \stackrel{\text{def}}{=} \langle \pi_1((\pi')_i) \rangle_{i \in \mathbb{N}}$) corresponds to the states visited by the automaton while

reading the word $(\pi')_{|_2}$ corresponding to the sequence of second components of the positions in π' (i.e., $(\pi')_{|_2} \stackrel{\text{def}}{=} \langle \pi_2((\pi')_i) \rangle_{i \in \mathbb{N}}$ – recall that \mathcal{A} is deterministic). Importantly, notice that such word $(\pi')_{|_2}$ is exactly $\text{PLAY2ASG}(\rho)$. Additionally, observe that the projections of π and π' on the first component of each position, i.e., $(\pi)_{|_1} \stackrel{\text{def}}{=} \langle \pi_1((\pi)_i) \rangle_{i \in \mathbb{N}}$ and $(\pi')_{|_1}$ respectively, are equal if we ideally merge together consecutive occurrences of the same state. This means that, since the winning condition Win_φ of \mathcal{G}_φ mimics the acceptance condition par of \mathcal{A} , the sequence of priorities corresponding to π is the same as the one corresponding to the run π' of \mathcal{A} on $\text{PLAY2ASG}(\rho)$. Therefore, $\text{PLAY2ASG}(\rho)$ is recognised by \mathcal{A} if and only if run π' is accepting if and only if play π is won by Eloise. \square

Finally, from Claim 4.14 and the following one, whose proof makes use of Claim 4.15, it follows that there is a strategy for Eloise to win \mathcal{G}_φ if and only if there is a strategy for her to win \mathcal{G}_φ^ψ . Thanks to this last equivalence and to Theorem 4.2 we conclude that for every realizable GFG-QPTL sentence φ there is a parity game won by Eloise if and only if φ is satisfiable.

Claim 4.16. ρ is won by Eloise in \mathcal{G}_φ iff $\text{PATH2PATH}(\rho)$ is won by Eloise in \mathcal{G}_φ^ψ , for every $\rho \in \text{Plays}_\varphi$.

Proof. Consider a play $\rho \in \text{Plays}_\varphi$. Thanks to Claim 4.15, we know that ρ is won by Eloise iff $\text{PLAY2ASG}(\rho)$ is accepted by \mathcal{A} which means that $\text{WRD}^{-1}(\text{PLAY2ASG}(\rho)) \models \psi$, which, in turn, is equivalent to say that $\text{PLAY2ASG}(\rho) \in \text{Win}_\varphi^\psi = \text{WRD}(\Psi)$, that is, $\text{PATH2PATH}(\rho)$ is won by Eloise in \mathcal{G}_φ^ψ , since $\text{PLAY2ASG}(\pi) = \text{OBS}(\text{PATH2PATH}(\rho))$. \square

The automaton \mathcal{A}_ψ has a size exponential in the size of ψ [VW86b]. The procedure to transform it into a deterministic parity automaton adds one exponential [Pit06]; thus $|\mathcal{A}| = 2^{2^{O(|\psi|)}}$. It is easy to see the number of positions of the quantification game is $O2^{|\varphi|}$. Thus, we conclude that game \mathcal{G}_φ we have just defined has size in $O2^{|\varphi|} \cdot 2^{2^{O(|\psi|)}} = 2^{2^{O(|\varphi|)}}$. The game has the same number of priorities as the automaton \mathcal{A} which is in $2^{O(|\psi|)}$. \square

PLAN LOGIC

As opposed to existing logics for strategic reasoning, such as ATL* [AHK97] and SL [CHP10; MMPV14; MMPV17], where the (implicit or explicit) domain of quantification is composed of strategies, which are quite complex objects, we introduce *Plan Logic*, which relies on the much simpler notion of a *plan*. Plans are infinite sequences $\rho \in \text{Plans} \stackrel{\text{def}}{=} \text{Ac}^\omega$ that describe the course of actions an agent chooses to execute in response to what the other agents have already decided to do.

From a syntactic standpoint, Plan Logic bears a strong similarity with SL. In particular, PL extends LTL by allowing

1. to quantify explicitly over plans,
2. to assign plans to agents by means of a binding mechanism similar to the one of SL that connects agents and plan variables, and
3. 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.

The chapter is organized as follows. In Section 5.1, we recall the definition of SL with Tarskian semantics and the timeline semantics of Gardy [GBM18; GBM20]. Following, in Section 5.2, we present the core concepts for reasoning with plans and the syntax and semantics of PL. Then, in Section 5.3, we compare PL and SL under the timeline semantics. Finally, in Section 5.4, we state that the model checking of PL is 2-EXPTIME-complete.

Throughout this chapter, we implicitly assume an *a priori* fixed countably-infinite set of *variables* Var .

5.1 Strategy Logic

In this section, we present the model of concurrent game structures, which are the usual models for SL. Then, we describe the typical syntax and semantics of SL, and

finally, we introduce the so-called *timeline semantics* for SL.

5.1.1 Concurrent Game Structures.

As said, SL formulas are usually interpreted over *concurrent game structure* (CGS , for short).

Definition 5.1 (Concurrent game structure). *A CGS w.r.t an a priori fixed countably-infinite set of atomic propositions AP is a structure $G \stackrel{\text{def}}{=} \langle \mathbf{Ag}, \mathbf{Ac}, \mathbf{Pos}, p_i, \delta, \lambda \rangle$, where*

- \mathbf{Ag} is a finite non-empty set of agents,
- \mathbf{Ac} and \mathbf{Pos} are countable non-empty sets of actions and positions,
- $p_i \in \mathbf{Pos}$ is an initial position,
- $\delta: \mathbf{Pos} \times \mathbf{Ac}^{\mathbf{Ag}} \rightarrow \mathbf{Pos}$ is a transition function mapping every position $p \in \mathbf{Pos}$ and action profile $\vec{d} \in \mathbf{Ac}^{\mathbf{Ag}}$ to a position $\delta(p, \vec{d}) \in \mathbf{Pos}$,
- $\lambda: \mathbf{Pos} \rightarrow 2^P$ is a labeling function mapping every position $p \in \mathbf{Pos}$ to the finite set of atomic propositions $\lambda(p) \subset_{\text{fin}} P$ true at that position.

The size of G is the number of its positions, i.e., $|G| \stackrel{\text{def}}{=} |\mathbf{Pos}|$.

By abuse of notation, $\delta \subseteq \mathbf{Pos} \times \mathbf{Pos}$ also denotes the *transition relation* between positions such that $(p, q) \in \delta$ iff $\delta(p, \vec{d}) = q$, for some $\vec{d} \in \mathbf{Ac}^{\mathbf{Ag}}$. A *path* $\pi \in \mathbf{Paths} \subseteq \mathbf{Pos}^\infty \setminus \{\varepsilon\}$ is a sequence of positions compatible with the transition function and beginning with the initial position, i.e., $\pi[0] = p_i$ and $\pi[i], \pi[i+1]) \in \delta$, for each $0 \leq i < |\pi| - 1$. The labeling function lifts from positions to paths in the usual way: $\lambda: \mathbf{Paths} \rightarrow (2^{\text{AP}})^+$. A *history* is a finite path $h \in \mathbf{Hist} \stackrel{\text{def}}{=} \mathbf{Paths} \cap \mathbf{Pos}^+$, while a *play* $\pi \in \mathbf{Plays} \stackrel{\text{def}}{=} \mathbf{Paths} \cap \mathbf{Pos}^\omega$ is an infinite one. A *strategy* is a function $\sigma \in \mathbf{Strat} \stackrel{\text{def}}{=} \mathbf{Hist} \rightarrow \mathbf{Ac}$ mapping every history $h \in \mathbf{Hist}$ to an action $\sigma(h) \in \mathbf{Ac}$. A play $\pi \in \mathbf{Plays}$ is *compatible* with a *strategy profile* $\vec{\sigma} \in \mathbf{Strat}^{\mathbf{Ag}}$ if, for all $i \in \mathbb{N}$, it holds that $\pi[i+1] = \delta(\pi[i], \vec{d}[i])$, where $\vec{d}[i] \in \mathbf{Ac}^{\mathbf{Ag}}$ is the action profile with $\vec{d}[i](a) = \vec{\sigma}(a)(\pi[:i])$, for all agents $a \in \mathbf{Ag}$. The function $\text{PLAYS}: \mathbf{Strat}^{\mathbf{Ag}} \rightarrow \mathbf{Plays}$ assigns to each profile $\vec{\sigma} \in \mathbf{Strat}^{\mathbf{Ag}}$ the unique play $\text{PLAYS}(\vec{\sigma}) \in \mathbf{Plays}$ compatible with $\vec{\sigma}$; we also say that $\vec{\sigma}$ *induces* $\text{PLAYS}(\vec{\sigma})$.

From now on, we implicitly assume that we have a CGS that is defined.

5.1.2 Syntax.

A *binding* $b \in \mathbf{Binds} \stackrel{\text{def}}{=} \mathbf{Ag} \rightarrow \mathbf{Var}$ is a function mapping every agent $a \in \mathbf{Ag}$ to a variable $b(a) \in \mathbf{Var}$, commonly represented as a finite sequence of binding pairs

$(a_1, x_1), \dots, (a_k, x_k)$, where each agent occurs exactly once. By $\text{VAR}(b) \subset \text{Var}$ we denote the set of variables occurring in b and lift the notation to sets of bindings as the union of the corresponding sets element-wise.

Definition 5.2. *An SL formula is defined inductively as follows, where ψ is an LTL formula, x is a strategic variable and b is a binding.*

$$\Phi := b\psi \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists x \Phi \mid \forall x \Phi$$

The atom $b\psi$ is called a *goal formula*.

The free variables of an SL formula are defined as follows:

$$\begin{aligned} \text{FREE}(b\psi) &= \text{VAR}(b) && \text{for } \psi \text{ an LTL formula and } b \text{ a binding} \\ \text{FREE}(\Phi_1 \wedge \Phi_2) &= \text{FREE}(\Phi_1) \cup \text{FREE}(\Phi_2) \\ \text{FREE}(\neg\Phi) &= \text{FREE}(\Phi) \\ \text{FREE}(Qx \Phi) &= \text{FREE}(\Phi) \setminus \{x\} && \text{for } Q \in \{\exists, \forall\} \end{aligned}$$

5.1.3 Semantics.

We display the semantics of SL as defined in [MMPV14; MMPV17].

Definition 5.3 (SL Tarski Semantics). *Tarski's semantic relation $\chi \models_{SL} \Phi$ for SL is inductively defined as follows, for all SL formulas Φ and assignments $\chi \in \text{Asg}(V)$ with $\text{FREE}(\Phi) \subseteq V$.*

1. $\chi \models b\psi$, if $\lambda(\text{PLAY}(b \circ \chi)) \models_{LTL} \psi$;
2. the semantics of Boolean connectives is defined as usual;
3. $\chi \models \exists x \Phi$, if $\chi[x \mapsto \sigma] \models \Phi$, for some strategy $\sigma \in \text{Strat}$;
4. $\chi \models \forall x \Phi$, if $\chi[x \mapsto \sigma] \models \Phi$, for all strategies $\sigma \in \text{Strat}$.

5.1.4 Timeline semantics.

The *timeline semantics* has been proposed by Gardy [GBM18; GBM20] and is defined for prenex formulas, i.e., formulas of the form $\wp \mathfrak{f}(\vec{b}\vec{\psi})$, where \wp is a quantifier prefix, $\mathfrak{f}(\vec{b}\vec{\psi})$ is a Boolean combination \mathfrak{f} , i.e. a function from $\{\top, \perp\}^{|\vec{b}\vec{\psi}|}$ to $\{\top, \perp\}$, of the goal formulas $\vec{b}\vec{\psi}$. The fragment of SL composed of only prenex formula is called SL *Boolean goal* denoted by SL[BG]. The fragment of SL[BG] where the Boolean combination is a

conjunction, *resp.* a disjunction, is called SL *conjunctive goal*, *resp.* SL *disjunctive goal* and is denoted by SL[CG], *resp.* SL[DG]. The *one-goal* fragment of SL, denoted by SL[1G], is the fragment where there is only one goal formula, with the Boolean function being the identity function. 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.

Timeline Dependence A map θ is a function $\theta : \mathbf{Strat}^{V_\forall} \rightarrow \mathbf{Strat}^V$ such that for every $\mathbf{m} \in \mathbf{Strat}^{V_\forall}$ and every variable x in V_\forall , we have $\theta(\mathbf{m})(x) = \mathbf{m}(x)$. Given a quantifier prefix $\wp = Q_1x_1, \dots, Q_nx_n$, a map is a *timeline map* if it respects the following.

$$\forall \mathbf{m}_1, \mathbf{m}_2 \in \mathbf{Strat}^{V_\forall}. \forall x_i \in V \setminus V_\forall. \forall h \in \mathbf{Hist},$$

$$\left(\begin{array}{l} \forall x_j \in V_\forall \cap \{x_k \mid k < i\}, \mathbf{m}_1(x_j)(h) = \mathbf{m}_2(x_j)(h) \\ \wedge \forall x_j \in V_\forall, \forall h' \in \mathbf{Pref}(h), \mathbf{m}_1(x_j)(h') = \mathbf{m}_2(x_j)(h') \end{array} \right) \rightarrow \theta(\mathbf{m}_1)(x)(h') = \theta(\mathbf{m}_2)(x)(h')$$

Remark that, even if the type of a map is close to the one of a Skolem function, the strategy of an existentially quantified variable may depend on the strategy of a variable universally quantified afterward.

Given a vector of goal formulas $\vec{b}\psi$ and an assignment χ , we define the Boolean vector $\vec{v}^{\vec{b}\psi, \chi}$ by $\vec{v}_i^{\vec{b}\psi, \chi} := \top$ iff $\chi \models_{\text{SL}} \vec{b}_i\vec{\psi}_i$.

Semantics $G \models_{\text{SL}[T]} \wp f(\vec{b}\vec{\psi})$ iff there is a timeline map θ such that $f(\vec{v}^{\vec{b}\psi, \theta(\mathbf{m})}) = \top$.

5.2 Strategic Reasoning

5.2.1 Syntax.

For simplicity, the syntax of the full logic imposes *flat* formulas to be *flat*, as in the flat fragments of CTL* [Dam99] and ATL* [GV14], where sentences can be combined in a Boolean way, but cannot be nested. Notice that this flatness constraint comes *without loss of generality*, when the model-checking problem is considered, as the latter can always be reduced to reasoning about flat formulas via a relabelling of the underlying structure (see [KVW00; AHK02], for details).

Definition 5.4. Plan Logic (*PL*, for short) is the set of formulas built according to the following context-free grammar, where $b \in \mathbf{Binds}$, $\psi \in \mathbf{LTL}$, $V \subseteq_{\mathbf{fin}} \mathbf{Var}$, and $x \in \mathbf{Var}$:

$$\varphi \stackrel{\text{def}}{=} b\psi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle V \rangle \varphi \mid [V] \varphi \mid \exists x \varphi \mid \forall x \varphi.$$

We shall denote by $\mathbf{FREE}(\varphi) \subseteq \mathbf{VAR}(\varphi) \subset \mathbf{Var}$ the sets of free variables and variables occurring in φ . Specifically, $\mathbf{FREE}(b\psi) \stackrel{\text{def}}{=} \mathbf{VAR}(b)$ and $\mathbf{FREE}(\langle V \rangle \varphi) = \mathbf{FREE}([V] \varphi) \stackrel{\text{def}}{=} V \cup \mathbf{FREE}(\varphi)$; all others cases are as usual. A *sentence* ϕ is a formula without free variables, i.e., $\mathbf{FREE}(\phi) = \emptyset$. Similarly, $\mathbf{BND}(\varphi) \subset \mathbf{Binds}$ denotes the set of bindings occurring in φ .

The binding b in a PL *goal* $b\psi$ 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 formulas 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 realizability, 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 realizable plan ...*” (*resp.*, “*for all realizable 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 $\langle V \rangle$ (*resp.*, $[V]$) 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.

5.2.2 Examples.

To better understand these intuitions, let us discuss some examples of SL formulas and their corresponding PL equivalents. The simple SL sentence $\Psi_W = \exists x \forall y (a, x)(b, y)\psi$ states that an agent a can win a two-player game with LTL objective ψ . Specifically, it requires the existence of a strategy x whose induced plays, each one induced by some strategy y of the adversary b , satisfy ψ . This same property would be expressed in PL by

the sentence $\phi_W = \exists x \forall y \langle x \rangle [y](a, x)(b, y)\psi$, which states that there exists a realizable plan followed by a that is part of some strategy, *e.g.*, the witness strategy for x of the SL sentence, and ensures the objective, regardless of the realizable plans y that are part of possible strategies y followed by the adversary. Note that the realizability 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 behavior 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 $\Psi_{NSD} = \exists x \forall x' \exists y ((a, x')(b, y)\psi \rightarrow (a, x)(b, y)\psi)$. The formula asserts that, for some strategy x and any other strategy x' , both for the same agent a , there is at least one strategy y for the opponent such that following x' instead of x would not give a a better outcome. In PL terms, that property is captured by the sentence $\phi_{NSD} = \exists x \forall x' \exists y_1, y_2 \langle x \rangle [x'] \langle y_1, y_2 \rangle ((a, x')(b, y_1)\psi \rightarrow (a, x)(b, y_2)\psi)$, where we ensure that the two plans y_1 and y_2 are part of the same existentially quantified strategy 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 $\Psi_{NE} = \exists x \exists y \forall z (((a, z)(b, y)\psi_a \rightarrow (a, x)(b, y)\psi_a) \wedge ((a, x)(b, z)\psi_b \rightarrow (a, x)(b, y)\psi_b))$, where x and 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 z instead of x and y . The corresponding PL sentence is

$$\phi_{NE} = \exists x_1, x_2 \exists y_1, y_2 \forall z_1, z_2 \langle x_1, x_2 \rangle \langle y_1, y_2 \rangle [z_1, z_2] \left(\begin{array}{c} ((a, z_1)(b, y_1)\psi_a \rightarrow (a, x_2)(b, y_2)\psi_a) \\ \wedge \\ ((a, x_1)(b, z_2)\psi_b \rightarrow (a, x_2)(b, y_2)\psi_b) \end{array} \right),$$

where the existential strategies x and 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 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 5.3, where a formal translation of some realizable fragments of SL is provided.

5.2.3 Semantics.

To interpret a goal $b\psi$ w.r.t an assignment $\chi \in \mathbf{Asg}(V)$ with $\text{VAR}(b) \subseteq V$, one needs to consider the play $\text{PLAY}_b(\chi)$ that is induced by the *plan profile* $\vec{\rho} \stackrel{\text{def}}{=} \chi \circ b \in \mathbf{Plans}^{\mathbf{Ag}}$ obtained as the functional composition of χ and b and associating a plan in χ with every agent, in accordance with the binding b . Formally, $\text{PLAY}_b(\chi)$ is the unique play $\pi \in \mathbf{Plays}$ such that $\pi[i+1] = \delta(\pi[i], \vec{d}[i])$, for all $i \in \mathbb{N}$, where $\vec{d}[i] \in \mathbf{Ac}^{\mathbf{Ag}}$ is the action profile associating with each agent $a \in \mathbf{Ag}$ the action stipulated at time i by the plan assigned to a in the plan profile $\vec{\rho}$, i.e., $\vec{d}[i](a) = \vec{\rho}(a)[i]$.

The semantics of the tying operators $\langle V \rangle$ and $[V]$ requires some intermediate notions. Two bindings $b_1, b_2 \in \mathbf{Binds}$ agree up to $n \in \mathbb{N}$ on an assignment $\chi \in \mathbf{Asg}$ if $\text{PLAY}_{b_1}(\hat{\chi}) =_{\leq n} \text{PLAY}_{b_2}(\hat{\chi})$, for some extension $\hat{\chi} \in \mathbf{Asg}(W)$ of χ with $\text{VAR}(\{b_1, b_2\}) \subseteq W$. Intuitively, b_1 and b_2 agree up to n on χ if two corresponding plan profiles induce the same history h of length $n+1$, where n evolution steps have occurred since the initial position. Note that b_1 and b_2 agree up to 0 on every assignment, since the initial position is always a common history of length 1. For an assignment $\chi \in \mathbf{Asg}$, two variables $x_1, x_2 \in \text{DOM}(\chi)$, and two bindings $b_1, b_2 \in \mathbf{Binds}$, with $x_1 \in \text{VAR}(b_1)$ and $x_2 \in \text{VAR}(b_2)$, we say that the pair (x_1, x_2) is (b_1, b_2) -*tied in* χ when, for every $n \in \mathbb{N}$, if b_1, b_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 $\sigma(h)$, for some n -evolution-step history h . We lift the notion to sets of variables $V \subseteq \text{DOM}(\chi)$ and bindings $B \subseteq \mathbf{Binds}$ as follows: V is B -*tied in* χ if (x_1, x_2) is (b_1, b_2) -tied in χ , for all $x_1, x_2 \in V$ and $b_1, b_2 \in B$, with $x_1 \in \text{VAR}(b_1)$ and $x_2 \in \text{VAR}(b_2)$.

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

Definition 5.5. For an implicitly given CGS G , Tarski's semantic relation $\chi \models \varphi$ for PL is inductively defined as follows, for all PL formulas φ and assignments $\chi \in \mathbf{Asg}(W)$ with $\text{FREE}(\varphi) \subseteq W$.

1. $\chi \models b\psi$, if $\lambda(\text{PLAY}_b(\chi)) \models_{\text{LTL}} \psi$;
2. the semantics of Boolean connectives is defined as usual;
3. $\chi \models \langle V \rangle \varphi$, if $\chi \models \varphi$ and V is $\text{BND}(\varphi)$ -tied in χ ;
4. $\chi \models [V] \varphi$, if $\chi \models \varphi$ when V is $\text{BND}(\varphi)$ -tied in χ ;
5. $\chi \models \exists x \varphi$, if $\chi[x \mapsto \rho] \models \varphi$, for some plan $\rho \in \mathbf{Plans}$;
6. $\chi \models \forall x \varphi$, if $\chi[x \mapsto \rho] \models \varphi$, for all plans $\rho \in \mathbf{Plans}$.

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 $\text{BND}(\varphi)$ 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 $\neg\langle V \rangle\varphi$ and $[V]\neg\varphi$.

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 the following non-realizably satisfiable SL sentence: $\phi_{NR} \stackrel{\text{def}}{=} \forall y \exists z ((a, y)XXp \leftrightarrow (a, z)Xp)$. The corresponding PL translation, obtained similarly to the previous examples, can be, indeed, shown satisfiable under the Tarskian semantics as follows.

Example 5.1. Consider the sentence $\phi_{NR} = \forall y \exists z [y]\langle z \rangle((a, y)XXp \leftrightarrow (a, z)Xp)$ and the single-agent two-action two-position CGS $\mathbf{G} = \langle \{a\}, \{0, 1\}, \{p_0, p_1\}, p_0, \delta, \lambda \rangle$, where

1. action 0 always leads to p_0 and action 1 always to p_1 , regardless of the current position, i.e., $\delta(p_i, \{a \mapsto j\}) = p_j$, and
2. position p_1 is the only one labelled by p , i.e., $\lambda = \{p_0 \mapsto \emptyset, p_1 \mapsto \{p\}\}$.

Being a sentence, we evaluate ϕ_{NR} against the empty assignment \emptyset . By applying Items 6 and 5 of Definition 5.5, we obtain $\mathbf{G}, \emptyset \models \phi_{NR}$ iff, for every plan ρ_y , there exists a plan ρ_z such that $\mathbf{G}, \{y \mapsto \rho_y, z \mapsto \rho_z\} \models [y]\langle z \rangle((a, y)XXp \leftrightarrow (a, z)Xp)$. 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, $\mathbf{G}, \{y \mapsto \rho_y, z \mapsto \rho_z\} \models [y]\langle z \rangle((a, y)XXp \leftrightarrow (a, z)Xp)$ iff $\mathbf{G}, \{y \mapsto \rho_y, z \mapsto \rho_z\} \models (a, y)XXp \leftrightarrow (a, z)Xp$. At this point, one can simply choose $\rho_z \stackrel{\text{def}}{=} \rho_y[1] \cdot 0^\omega$ to satisfy the formula. Hence, following the naive interpretation of ϕ_{NR} via Tarski's semantics, it holds that \mathbf{G} satisfies ϕ_{NR} in a non-realizable way, since ρ_z requires knowledge of ρ_y one step ahead.

This example clearly shows that a precise formalization 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 realizable 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 realizability 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 hyperteam semantic framework as it is precisely adequate to handle realizable functional dependencies among quantified variables, as we shown in Chapter 4.

We now turn to adapting the hyperteam semantics for PL. We restrict response functions to be realizable, as presented in Section 4.2. Then, we only provide the lift of the concept of tied bindings to the level of hyperteams. First, at the level of a team X , we say that a set of variables $V \subseteq \text{VAR}(X)$ is *B-tied in X* , for a set of bindings $B \subseteq \text{Binds}$, if V is B -tied in every assignment $\chi \in X$. Second, for a hyperteam \mathfrak{X} , we define the *filtering* operator $\text{FLT}(\mathfrak{X}, V, B) \stackrel{\text{def}}{=} \{X \in \mathfrak{X} \mid V \text{ is } B\text{-tied in } X\}$, by filtering out of \mathfrak{X} all teams X in which V is not B -tied.

The compositional semantics of PL based on hyperteams is defined as follows.

Definition 5.6. *The hyperteam semantics relation $\mathfrak{X} \models^\alpha \varphi$ is inductively defined as in Definition 2.1, for all but Items 3, 7a and 8b that are modified, respectively, as follows, with the addition of Items 9 and 10, for all PL formulas φ , alternation flags $\alpha \in \{\exists^\forall, \forall^\exists\}$, and hyperteams $\mathfrak{X} \in \text{HypTeams}_\perp(\text{FREE}(\varphi))$:*

- 3") $\mathfrak{X} \models^{\exists^\forall} \flat\psi$, if there exists $X \in \mathfrak{X}$ such that $\lambda(\text{PLAY}_\flat(\chi)) \models_{\text{LTL}} \psi$, for all $\chi \in X$;
- 7a") $\mathfrak{X} \models^{\exists^\forall} \exists x \varphi$, if $\text{EXT}_R(\mathfrak{X}, x) \models^{\exists^\forall} \varphi$;
- 8b") $\mathfrak{X} \models^{\forall^\exists} \forall x \varphi$, if $\text{EXT}_R(\mathfrak{X}, x) \models^{\forall^\exists} \varphi$;
- 9. (a) $\mathfrak{X} \models^{\exists^\forall} \langle V \rangle \varphi$, if $\text{FLT}(\mathfrak{X}, V, \text{BND}(\varphi)) \models^{\exists^\forall} \varphi$;
- (b) $\mathfrak{X} \models^{\forall^\exists} \langle V \rangle \varphi$, if $\bar{\mathfrak{X}} \models^{\exists^\forall} \langle V \rangle \varphi$;
- 10. (a) $\mathfrak{X} \models^{\exists^\forall} [V] \varphi$, if $\bar{\mathfrak{X}} \models^{\forall^\exists} [V] \varphi$;
- (b) $\mathfrak{X} \models^{\forall^\exists} [V] \varphi$, if $\text{FLT}(\mathfrak{X}, V, \text{BND}(\varphi)) \models^{\forall^\exists} \varphi$;

For the rest of the chapter, as only realizable functions are considered, we might use $\text{EXT}(\mathfrak{X}, x)$ instead of $\text{EXT}_R(\mathfrak{X}, x)$. The semantics of the tying operators (Items 9 and 10) relies on the filtering operations discussed above. It is immediate to observe that, for a fixed CGS \mathbf{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., $\{\{\emptyset\}\} \models^{\exists^\forall} \phi$ iff $\{\{\emptyset\}\} \models^{\forall^\exists} \phi$, due to the self duality of $\{\{\emptyset\}\}$. We shall thus write $\mathbf{G} \models_{\text{PL}} \phi$ to assert both $\{\{\emptyset\}\} \models^{\exists^\forall} \phi$ and $\{\{\emptyset\}\} \models^{\forall^\exists} \phi$.

As expected at this point, the hyperteam semantics of PL has the adequacy theorem with respect to the Tarskian semantics of Definition 5.5.

Theorem 5.1. *For all PL quantifier-free formulas φ and hyperteam $\mathfrak{X} \in \text{HypTeams}_{\supseteq}(\text{FREE}(\varphi))$:*

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

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

Example 5.2. *Consider again the sentence $\phi_{NR} = \forall y \exists z [y]\langle z \rangle ((a, y)XXp \leftrightarrow (a, z)Xp)$ and the CGS \mathbf{G} of Example 5.1. We want to show that $\mathbf{G} \not\models_{PL} \phi_{NR}$, meaning that ϕ_{NR} is not realizably satisfiable on \mathbf{G} , i.e., there is no realizable plan for z ensuring a match of the truth values of p at time instants 1 and 2. Since $\text{FREE}(\phi_{NR}) = \emptyset$, we evaluate ϕ_{NR} against the trivial hyperteam $\{\{\emptyset\}\}$, which, as observed before, implies that the alternation flag is of no consequence. Without loss of generality, we choose $\alpha = \forall\exists$, thus focusing on proving $\{\{\emptyset\}\} \not\models^{\forall\exists} \phi_{NR}$.*

The rule for the universal quantifier $\forall y$ (Item 8'') requires to compute the extension $\mathfrak{X} \stackrel{\text{def}}{=} \text{EXT}(\{\{\emptyset\}\}, y) = \{\{y : 000^\omega\}, \{y : 010^\omega\}, \{y : 100^\omega\}, \{y : 110^\omega\}, \dots\}$ of $\{\{\emptyset\}\}$, containing a singleton team for each one of the uncountably many plans to assign to y . This results in

$$\{\{\emptyset\}\} \models^{\forall\exists} \forall y \exists z [y]\langle z \rangle ((a, y)XXp \leftrightarrow (a, z)Xp) \text{ iff } \mathfrak{X} \models^{\forall\exists} \exists z [y]\langle z \rangle ((a, y)XXp \leftrightarrow (a, z)Xp).$$

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. Since every team of \mathfrak{X} is a singleton set, there is only one possible choice function for it, thus, the result is

$$\mathfrak{X} \models^{\forall\exists} \exists z [y]\langle z \rangle ((a, y)XXp \leftrightarrow (a, z)Xp) \text{ iff } \bar{\mathfrak{X}} \models^{\exists\forall} \exists z [y]\langle z \rangle ((a, y)XXp \leftrightarrow (a, z)Xp),$$

where $\bar{\mathfrak{X}} = \{\{y : 000^\omega, y : 010^\omega, y : 100^\omega, y : 110^\omega, \dots\}\}$ 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 $\mathfrak{X}' \stackrel{\text{def}}{=} \text{EXT}(\bar{\mathfrak{X}}, z)$. The result is

$$\bar{\mathfrak{X}} \models^{\exists\forall} \exists z [y]\langle z \rangle ((a, y)XXp \leftrightarrow (a, z)Xp) \text{ iff } \mathfrak{X}' \models^{\exists\forall} [y]\langle z \rangle ((a, y)XXp \leftrightarrow (a, z)Xp),$$

where $\left\{ \left\{ \begin{array}{l} y:000^\omega \ y:010^\omega \ y:100^\omega \ y:110^\omega \\ z:00^\omega \ , \ z:00^\omega \ , \ z:00^\omega \ , \ z:00^\omega \ , \dots \end{array} \right\} \left\{ \begin{array}{l} y:000^\omega \ y:010^\omega \ y:100^\omega \ y:110^\omega \\ z:10^\omega \ , \ z:10^\omega \ , \ z:00^\omega \ , \ z:00^\omega \ , \dots \end{array} \right\}, \dots \right\}$ is the hyperteam \mathfrak{X}' containing one team $\text{EXT}(X, F, z)$ for every response function $F \in \text{Rsp}_R$, where $X = \{y:000^\omega, y:010^\omega, y:100^\omega, y:110^\omega, \dots\}$ is the unique team in $\overline{\mathfrak{X}}$. For instance, the first team in \mathfrak{X}' is obtained by applying the constant function $F(\chi) = 0^\omega$, while, for the second one, we use the time-0-flip function $F(\chi) = (1 - \chi(y)[0]) \cdot 0^\omega$. In general, by the realizable restriction, the action $F(\chi)[0]$ 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 $\text{PLAY}_{(a,y)}(\chi)[2] \neq \text{PLAY}_{(a,z)}(\chi)[1]$. Therefore, for all $X' \in \mathfrak{X}'$, it holds that $\chi \not\models [y]\langle z \rangle ((a, y)XXp \leftrightarrow (a, z)Xp)$, for some $\chi \in X'$. As a consequence of Item 1 of Theorem 5.1, it holds that $\mathfrak{X}' \not\models^{\exists\forall} [y]\langle z \rangle ((a, y)XXp \leftrightarrow (a, z)Xp)$, which, in turn, means that $\{\{\emptyset\}\} \not\models^{\forall\exists} \phi_{NR}$ and, so, $\mathbf{G} \not\models_{PL} \phi_{NR}$, as expected.

5.3 Adequacy with Strategy Logic under Timeline Semantics

While the PL semantics – thanks to the tying operators – ensures that the strategies involved are also realizable, we have shown for instance that the SL formula $\forall y. \exists z. ((a, y)XXp \circ (a, z)Xp)$ involves strategies that are not. As an immediate consequence, the two logics are not directly comparable. Still, as shown in [MMPV14], for the *one-goal fragment* of SL, a formula is satisfiable iff it is satisfiable when quantifying only over realizable strategies [MMPV14].

We relate SL with timeline semantics (Section 5.1.4) and PL, by showing that the SL *conjunctive goal* and the *disjunctive goal* fragments, can be translated into PL. We introduce a game-theoretic semantics of SL with timeline semantics (whose correctness is established in Theorem 5.3), 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 [EG22].

5.3.1 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 Q of existential $\exists x$ and universal $\forall x$ quantifiers, where each variable occurs at most once. The

set of variables occurring in a quantifier prefix Q is $\text{VAR}(Q)$, and we let $\text{VAR}_{\exists}(\varphi)$ (*resp.* $\text{VAR}_{\forall}(\varphi)$) 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.

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 that use the considered strategy variable. 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 subformula with the corresponding plan variable for that goal. More in detail, let $\Phi = \varphi\phi$ be a SL[BG] formula. Each quantifier Qx in φ is transformed into a sequence of quantifiers of the form Qx_b , one for every $b \in \text{BND}(\varphi)$ with $x \in \text{VAR}(b)$. Formally, the quantifier prefix of the translation is $\varphi_{\text{GTS}}(\Phi) \stackrel{\text{def}}{=} ((Q_x x_b)_{b \in B_x})_{x \in \text{VAR}(\varphi)}$ with $B_x = \{b \in \text{BND}(\varphi) \mid x \in \text{VAR}(b)\}$ and $Q_x = \exists$ if $x \in \text{VAR}_{\exists}(\varphi)$ and $Q_x = \forall$ if $x \in \text{VAR}_{\forall}(\varphi)$.

We now keep track of the fact that the various obtained variables x_b stem from a single variable x by tying them in a coherent manner via a tying prefix $T_{\text{GTS}}(\Phi)$: when variable x was existentially (*resp.* universally) quantified, the tying of the x_b 's is existential (*resp.* universal) as follows. Writting $V_x \stackrel{\text{def}}{=} \{x_b \mid b \in B_x\}$ for the set of plan variables associated with the strategy variable x , we let $T_{\text{GTS}}(\Phi) \stackrel{\text{def}}{=} ((V_x)_{x \in \text{VAR}_{\exists}(\varphi)})([V_x]_{x \in \text{VAR}_{\forall}(\varphi)})$. Finally, in each original goal subformula $b\psi$, we replace every occurrence of variable x with the new variable x_b . The complete translation of the subformula ϕ (a Boolean combination of goals) is denoted by $\phi_{\text{GTS}}(\Phi)$. Gathering all the translation components we have defined, we obtain $\text{GTS}_{\text{SL}}(\Phi) \stackrel{\text{def}}{=} \varphi_{\text{GTS}}(\Phi)T_{\text{GTS}}(\Phi)\phi_{\text{GTS}}(\Phi)$, 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.

Theorem 5.2. $G \models_{\text{SL}} \Phi$ iff $G \models \text{GTS}_{\text{SL}}(\Phi)$, for all SL[C/DG] sentence Φ and CGS G .

The proof of Theorem 5.2 is reported in Section 5.A. In a nutshell, we first introduce a game-theoretic semantics for SL by reducing the evaluation of an SL[C/DG] sentence Φ in a given CGS G to the evaluation of a corresponding PL formula $\text{GTS}_{\text{SL}}(\Phi)$ in a modified CGS $\text{GTS}_{\text{SL}}(G, \Phi)$. This construction turns out to be a game-theoretic semantics for the conjunctive and disjunctive goal fragments of SL.

5.3.2 Game-theoretic Semantics of SL[C/DG]

The game-theoretic semantics of SL[C/DG] employs an additional *operator agent*, who plays the role of the single Boolean operator involved in the quantifier-free subformula 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 $G = \langle \text{Ag}, \text{Ac}, \text{Pos}, p_i, \delta, \lambda \rangle$ and an SL[C/DG] sentence $\Phi = \wp\phi$, we build the new CGS $GTS_{SL}(G, \Phi)$ and the new formula $GTS_{SL}(\Phi)$ as follows, where $B_\Phi = \text{BND}(\Phi)$.

Construction 5.1. *In CGS $GTS_{SL}(G, \Phi)$, a position $\hat{p} = (p, B)$ stems from a position p in G and is decorated with a set B of bindings, precisely those that agree so far along the history that led to \hat{p} . We set $\widehat{\text{Pos}} \stackrel{\text{def}}{=} \{\hat{p}_\exists, \hat{p}_\forall, \hat{p}_\odot\} \cup \text{Pos} \times 2^{B_\Phi}$, where three special sink positions $\hat{p}_\exists, \hat{p}_\forall$ and \hat{p}_\odot are explained later. The initial position of $GTS_{SL}(G, \Phi)$ is $\hat{p}_i = (p_i, B_\Phi)$, since at the beginning all the bindings agree on the empty history. The set of agents in $GTS_{SL}(G, \Phi)$ gathers the variable agents, one for each variable quantified in \wp , and the extra operator agent, written x_\odot , i.e. $\widehat{\text{Ag}} \stackrel{\text{def}}{=} \text{VAR}(\Phi) \cup \{x_\odot\}$. The actions of $GTS_{SL}(G, \Phi)$ include all the actions of the original CGS G and a new binding action for each binding occurring in the original formula Φ , i.e. $\widehat{\text{Ac}} \stackrel{\text{def}}{=} \text{Ac} \cup B_\Phi$. The variable agents are only allowed to choose an action from the original CGS, while binding actions are reserved to agent x_\odot , 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 $\hat{p}_\exists, \hat{p}_\forall$ and \hat{p}_\odot . Specifically, position \hat{p}_\exists (resp. \hat{p}_\forall) is reached every time the agent for a universally (resp. existentially) quantified variable mischooses a binding action instead of a proper one. Conversely, \hat{p}_\odot is reached any time agent x_\odot 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 $\vec{d} \in (\text{Ac} \cup \text{BND}(\wp))^{\text{VAR}(\wp) \cup \{x_\odot\}}$ is Q -ill-typed, for $Q \in \{\forall, \exists\}$, if the leftmost variable x in the quantifier prefix \wp such that $\vec{d}(x) \notin \text{Ac}$ is Q -quantified, and that \vec{d} is \odot -ill-typed in position $\hat{p} = (p, B)$ if $\vec{d}(x) \notin B$. An action profile is well-typed in position \hat{p} if it is neither Q -ill-typed nor \odot -ill-typed in position \hat{p} . The notion of bindings that agree with the choice of x_\odot is formalized as follows. We say that two bindings $b_1, b_2 \in \text{Binds}$ (whose variables are in $\text{VAR}(\wp)$) are indistinguishable at position $p \in \text{Pos}$ w.r.t action assignment $\vec{d} \in \text{Ac}^{\text{VAR}(\wp)}$ of variable*

agents, in symbols $b_1 \equiv_p^{\vec{d}} b_2$, whenever $\delta(p, \vec{d} \circ b_1) = \delta(p, \vec{d} \circ b_2)$, i.e., the same position is reached by playing either $\vec{d} \circ b_1$ or $\vec{d} \circ b_2$. A move at position $\hat{p} = (p, B)$ with well-typed action profile \vec{d} in \hat{p} leads to position $\hat{q} = (q, C)$ where $q = \delta(p, \vec{d} \circ b)$ for the choice $b = \vec{d}(x_\odot)$ of agent x_\odot , and $C \subseteq B$ retains only the bindings that are indistinguishable from b (at p w.r.t \vec{d}). Formally,

$$\widehat{\delta}(\hat{p}, \vec{d}) \stackrel{\text{def}}{=} \begin{cases} \hat{p}_Q & \text{if } \hat{p} = \hat{p}_Q, \text{ or } \hat{p} \neq \hat{p}_\odot \text{ and } \vec{d} \text{ is } \overline{Q}\text{-ill-typed, with } Q \in \{\exists, \forall\}; \\ \hat{p}_\odot & \text{if } \hat{p} = \hat{p}_\odot \text{ or } \vec{d} \text{ is } \odot\text{-ill-typed in } \hat{p}; \\ (\delta(p, \vec{d} \circ b), \{b' \in B \mid b' \equiv_p^{\vec{d}} b\}) & \text{with } (p, B) = \hat{p} \text{ and } b = \vec{d}(x_\odot), \text{ otherwise.} \end{cases}$$

Finally, the label of $\hat{p} = (p, B)$ inherits from the label of p in \mathbf{G} with the extra propositions q_b , one for each binding $b \in B$. Formally, $\widehat{\lambda}(\hat{p}_\exists) \stackrel{\text{def}}{=} \{p_\exists\}$, $\widehat{\lambda}(\hat{p}_\forall) \stackrel{\text{def}}{=} \{p_\forall\}$, $\widehat{\lambda}(\hat{p}_\odot) \stackrel{\text{def}}{=} \emptyset$, and $\widehat{\lambda}((p, B)) \stackrel{\text{def}}{=} \lambda(p) \cup \{q_b \in AP \mid b \in B\}$.

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

$$\text{GTS}_{\text{SL}}(\wp \wedge_{b \in B} b\psi_b) \stackrel{\text{def}}{=} Q\forall x_\odot b_{\text{id}}((\diamond p_\exists) \vee ((\Box \neg p_\forall) \wedge \bigwedge_{b \in B} ((\Box q_b) \rightarrow \psi_b))).$$

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

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

$$\text{GTS}_{\text{SL}}(\wp \vee_{b \in B} b\psi_b) \stackrel{\text{def}}{=} Q\exists x_\odot b_{\text{id}}((\Box \neg p_\forall) \wedge ((\diamond p_\exists) \vee \bigvee_{b \in B} ((\Box q_b) \wedge \psi_b))).$$

The following theorem states that the above constructions provide a proper game-theoretic semantics for $\text{SL}[\text{C/DG}]$.

Theorem 5.3. $\mathbf{G} \models_{\text{SL}} \Phi$ iff $\text{GTS}_{\text{SL}}(\mathbf{G}, \Phi) \models \text{GTS}_{\text{SL}}(\Phi)$, for all $\text{SL}[\text{C/DG}]$ sentences Φ .

We sketch the proof road-map of Theorem 5.3 that consists in showing both

1. that the truth of an SL[CG] formula entails the truth of its GTS_{SL} translation, and
2. that the truth of an SL[DG] formula entails the truth of its GTS_{SL} translation.

Observe that the if direction of Theorem 5.3 follows from (the contrapositions of) Items 1 and 2, the determinacy of SL, and the duality of the GTS_{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 5.4. $G \models GTS(\Phi)$ iff $GTS_{SL}(G, \Phi) \models GTS_{SL}(\Phi)$, for all SL[C/DG] sentences Φ .

Similarly to the preceding proof approach, we show that

1. the truth of $GTS_{SL}(\Phi)$, where $\Phi \in SL[CG]$, entails the truth of its GTS_{SL} translation, and
2. the truth of $GTS(\Phi)$, where $\Phi \in SL[DG]$, entails the truth of its GTS_{SL} translation.

Notice that both formulas are in PL, but that formula $GTS_{SL}(\Phi)$ is based on duplicates x_b 's of the original variables x in Φ , while in $GTS(\Phi)$ 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_b 's is made possible thanks to the tying operators introduced in the formula $GTS_{SL}(\Phi)$.

Theorems 5.3 and 5.4 entail Theorem 5.2.

5.4 Decision Problems

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. In [BGM15], 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]) [MMPV14], 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 2-EXPTIME-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 G to the evaluation of a PL[1G] sentence $\hat{\phi}$ in a modified structure \hat{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.

5.4.1 Goal Fragments of PL.

The *Boolean-Goal fragment* of PL (PL[BG]) comprises all positive Boolean combinations of formulas (in prenex form) $\varphi T\phi$, where φ is a quantifier prefix, T 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 ϕ_{NE} of the existence of a Nash equilibrium discussed in Section 5.2 is an example of PL[BG] formula, as well as the sentence ϕ_{NB} of Example 5.1. The sentence ϕ_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 ϕ_{NSD} . By turning $\neg\phi_{NSD}$ 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 [ABM19], 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 $GTS: SL \rightarrow PL$, when applied to an SL[C/DG] sentence, necessarily returns a PL[C/DG] one. Finally, $GTS_{SL}: SL \rightarrow PL$ always produces a PL[1G] sentence.

5.4.2 One-goal Fragment

A simple inspection of the syntactic translation $GTS: SL \rightarrow PL$ 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 $\wp T \flat \psi$:

1. if T contains $\langle V \rangle$, with two variables $x, y \in V$, where y is universally quantified after x in \wp , then the subformula originating in $\langle V \rangle$ is equivalent to \perp ;
2. dually, if T contains $[V]$, with two variables $x, y \in V$, where y is existentially quantified after x in \wp , then the subformula originating in $[V]$ is equivalent to \top ;
3. 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

1. $\wp[x, y]\langle y, z \rangle \flat \psi \equiv \top$,
2. $\wp\langle y, z \rangle[x, y] \flat \psi \equiv \perp$, and
3. $\wp\langle x, y \rangle[y, z] \flat \psi \equiv \forall x. (a, x)(b, x)(c, x)\psi$.

Thus, the following tying-elimination property holds true.

Proposition 5.1. *For each sentence of the form $\wp T \flat \psi$, there is an equivalent one $\wp' \flat' \psi$.*

By combining this proposition with Theorem 5.2, we obtain that the One-Goal fragments of SL and PL semantically coincide.

Theorem 5.5. *For every SL[1G] sentence Ψ , there is a PL[1G] sentence ϕ and, vice versa, for every PL[1G] sentence ϕ , there is an SL[1G] sentence Ψ such that: $|\Psi| = \Theta|\phi|$ and $\mathbf{G} \models_{SL} \Psi$ iff $\mathbf{G} \models_{PL} \phi$.*

Due to the known $2^{2^{O(|\phi|)}} \cdot |\mathbf{G}|^{O1}$ complexity of the model-checking problem of SL[1G] [MMPV14], we can immediately derive the following theorem.

Theorem 5.6. *PL[1G] model-checking problem is 2-EXPTIME-complete in the length of the specification ϕ and PTIME-complete in the size of the model \mathbf{G} .*

5.4.3 Boolean-Goal Fragment

The encoding underlying Theorem 5.3 of the game-theoretic semantics for SL[C/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 \mathbf{G} and a sentence $\phi = \wp T \phi$, with binding

set $B_\varphi \stackrel{\text{def}}{=} \text{BND}(\varphi)$, we build a new CGS $\text{GTS}_{\text{BG}}(\mathbf{G}, \phi)$, whose plays are bundles of plays from \mathbf{G} , one per binding in ϕ . This is done, intuitively, by composing in parallel as many copies of \mathbf{G} as there are bindings in ϕ , resulting in positions that correspond to vectors $\hat{p} \in \text{Pos}^{B_\varphi}$ of original positions of \mathbf{G} . Agents of the new game coincide with the variables quantified in φ , while actions carries over unchanged. A move from a position \hat{p} to a position \hat{q} is then a vector of parallel moves in \mathbf{G} , one per original position contained in \hat{p} , while forbidding incoherent concurrent choices *w.r.t* the tying operators occurring in T . Formally, an action assignment $\vec{d} \in \text{Ac}^{\text{VAR}(\varphi)}$ is *V-incoherent at \hat{p} w.r.t ϕ* , where $V \subset \text{Var}$, if there are two variables $x_1, x_2 \in V$ and two bindings $b_1, b_2 \in B_\varphi$, with $x_1 \in \text{VAR}(b_1)$ and $x_2 \in \text{VAR}(b_2)$, such that $\hat{p}(b_1) = \hat{p}(b_2)$, but $\vec{d}(x_1) \neq \vec{d}(x_2)$. In other words, a concurrent move \vec{d} is *V-incoherent at \hat{p} w.r.t ϕ* , if there are variables in V whose different associated actions in \vec{d} should have been equal, being part of bindings that are indistinguishable at \hat{p} . We say that \vec{d} is *\exists -incoherent (resp., \forall -incoherent) at \hat{p} w.r.t ϕ* if the leftmost set of variables V in T , such that \vec{d} is *V-incoherent at \hat{p} w.r.t ϕ* , occurs in a tying operator of type $\langle V \rangle$ (resp., $[V]$). Intuitively, \vec{d} is *\exists/\forall -incoherent at \hat{p} w.r.t ϕ* if the first violated tying constraint specified in T is existential/universal. If \vec{d} is neither \exists -incoherent nor \forall -incoherent at \hat{p} w.r.t ϕ , we say that \vec{d} is *coherent at \hat{p} w.r.t ϕ* .

Construction 5.2. *Given a CGS $\mathbf{G} = \langle \text{Ag}, \text{Ac}, \text{Pos}, p_i, \delta, \lambda \rangle$ and a PL[BG] sentence ϕ , with binding set $B_\varphi \stackrel{\text{def}}{=} \text{BND}(\varphi)$, let $\text{GTS}_{\text{BG}}(\mathbf{G}, \phi) \stackrel{\text{def}}{=} \langle \widehat{\text{Ag}}, \widehat{\text{Ac}}, \widehat{\text{Pos}}, \widehat{p}_i, \widehat{\delta}, \widehat{\delta} \rangle$ be the CGS obtained as follows:*

1. *agents are the variables quantified in ϕ , i.e., $\widehat{\text{Ag}} \stackrel{\text{def}}{=} \text{VAR}(\phi)$;*
2. *$\widehat{\text{Ac}} \stackrel{\text{def}}{=} \text{Ac}$;*
3. *positions are B_φ -indexed vectors of original positions from \mathbf{G} , plus two distinguished sink positions \widehat{p}_\exists and \widehat{p}_\forall , i.e., $\widehat{\text{Pos}} \stackrel{\text{def}}{=} \{\widehat{p}_\exists, \widehat{p}_\forall\} \cup \text{Pos}^{B_\varphi}$;*
4. *the initial position is the p_i -constant vector, i.e., $\widehat{p}_i \stackrel{\text{def}}{=} \{b \in B_\varphi \mapsto p_i\}$;*
5. *every position, but the distinguished ones, are labelled with a set of fresh atomic propositions, one per binding and original labelling, i.e., $\widehat{\delta}(\widehat{p}_\exists) \stackrel{\text{def}}{=} \{p_\exists\}$, $\widehat{\delta}(\widehat{p}_\forall) \stackrel{\text{def}}{=} \{p_\forall\}$, and $\widehat{\delta}(\widehat{p}) \stackrel{\text{def}}{=} \{p_b \in AP \mid b \in B_\varphi, p \in \delta(\widehat{p}(b))\}$;*
6. *the transition function $\widehat{\delta}$ maps every position $\widehat{p} \in \widehat{\text{Pos}} \setminus \{\widehat{p}_\exists, \widehat{p}_\forall\}$ and action profile $\vec{d} \in \text{Ac}^{\text{VAR}(\phi)}$ coherent at \widehat{p} w.r.t ϕ to position $\widehat{q} \in \widehat{\text{Pos}} \setminus \{\widehat{p}_\exists, \widehat{p}_\forall\}$, where, for each binding $b \in B_\varphi$, the position $\widehat{q}(b)$ is the successor of $\widehat{p}(b)$ in \mathbf{G} following the action profile $\vec{d} \circ b$, which associates with each agent $a \in \text{Ag}$ the action stipulated by \vec{d} for*

the variable $\flat(a)$; formally,

$$\widehat{\delta}(\widehat{p}, \vec{d}) \stackrel{\text{def}}{=} \begin{cases} \widehat{p}_\wp, & \text{if } \widehat{p} = \widehat{p}_\wp \text{ or } \vec{d} \text{ is } \overline{\wp}\text{-incoherent at } \widehat{p} \text{ w.r.t } \phi, \text{ with } \wp \in \{\exists, \forall\}; \\ \widehat{q}, & \text{otherwise, where } \widehat{q}(\flat) \stackrel{\text{def}}{=} \delta(\widehat{p}(\flat), \vec{d} \circ \flat), \text{ for all } \flat \in B_\wp. \end{cases}$$

The PL[1G] encoding of the game-theoretic semantics for the PL[BG] sentence $\phi = \wp T\phi$ is relatively easy to formalize at this point: besides verifying the coherence constraints dictated by the tying prefix T , we only need to check that the bundles of plays induced by plans in the CGS $\text{GTS}_{\text{BG}}(\mathbf{G}, \phi)$ satisfy the subformula ϕ . Checking the constraints amounts to requiring avoidance of the two distinguished sink positions \widehat{p}_\exists and \widehat{p}_\forall . The verification of the subformula 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., $\widehat{\phi} \stackrel{\text{def}}{=} \phi[\flat\psi/\widehat{\psi}]$, with $\widehat{\psi} \stackrel{\text{def}}{=} \psi[p/p_\flat]$. Altogether, we get the following:

$$\text{GTS}_{\text{BG}}(\wp T\phi) \stackrel{\text{def}}{=} \wp \flat_{\text{id}} \left((\diamond p_\exists) \vee ((\Box \neg p_\forall) \wedge \widehat{\phi}) \right).$$

Since the original PL[BG] sentence ϕ and its PL[1G] translation $\text{GTS}_{\text{BG}}(\phi)$ share the same quantifier prefix \wp , thanks to Theorem 5.1, we can prove the correctness of the encoding, on the basis that $\chi \models T\phi$ iff $\chi \models \flat_{\text{id}}((\diamond p_\exists) \vee ((\Box \neg p_\forall) \rightarrow \widehat{\phi}))$, for all $\chi \in \text{Asg}(V)$ with $\text{VAR}(\wp) \subseteq V$, which can be done by structural induction on $T\phi$ (using the simple semantic rules of Definition 5.5).

Theorem 5.7. $\mathbf{G} \models \phi$ iff $\text{GTS}_{\text{BG}}(\mathbf{G}, \phi) \models \text{GTS}_{\text{BG}}(\phi)$, for every PL[BG] sentence ϕ .

Once we observe that $|\text{GTS}_{\text{BG}}(\mathbf{G}, \phi)| = 2 + |\mathbf{G}|^{|\text{BND}(\phi)|}$ and $|\text{GTS}_{\text{BG}}(\phi)| = O|\phi|$, thanks to Theorem 5.6, we can derive the following result, where FPT means *fixed-parameter tractable*.

Theorem 5.8. *The model-checking problem for PL[BG] is 2-EXPTIME-complete $[|\phi|]$ in the length of the specification ϕ and FPT $^{|\phi|}(|\mathbf{G}|)$ in the size of the model \mathbf{G} , with the length of the specification ϕ as parameter, once the maximum number of bindings is fixed.*

5.4.4 Collusion-free Fragment

The simpler conjunctive/disjunctive nature of goal combinations in PL[C/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 Construction 5.1 and Construction 5.2: from the former we inherit the topology of the structure, 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 5.1, since strategies are considered). We end-up with an *ad hoc* game-theoretic semantics for PL[C/DG], whose resulting CGS encoding $GTS_{C/DG}(\mathbf{G}, \phi)$ is virtually identical to Construction 5.1, where the notion of well-typed action assignment is generalized to take into account the coherence constraints introduced for Construction 5.2. The sentence encoding $GTS_{C/DG}(\phi)$ is also identical to the one used for SL[C/DG] in association with Construction 5.1.

The following theorem can be obtained as a slight adaptation of the proof of Theorem 5.4.

Theorem 5.9. $\mathbf{G} \models \phi$ iff $GTS_{C/DG}(\mathbf{G}, \phi) \models GTS_{C/DG}(\phi)$, for every PL[C/DG] sentence ϕ .

Once we observe that $|GTS_{C/DG}(\mathbf{G}, \phi)| = 2 + 2^{|\text{BND}(\phi)|} \cdot |\mathbf{G}|$ and $|GTS_{C/DG}(\phi)| = O|\phi|$, we can derive the following result, again thanks to Theorem 5.6.

Theorem 5.10. *The model-checking problem for PL[C/DG] is 2-EXPTIME-complete in the length of the specification ϕ and PTIME-complete in the size of the model \mathbf{G} .*

5.5 Conclusion

In this chapter, 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 realizability of the satisfiable sentences via *realizability 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 [GBM18; GBM20], which, however, exhibits neither compositionality nor determinacy. The authors of [GBM20] show, indeed, that such a semantics is not adequate already when applied to SL[BG], as it is undetermined on sentences of that fragment.

We showed that, thanks to the realizable 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] [BGM15]. This further highlights the importance of enforcing realizability 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 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 5.10). 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).

To improve the presented results, one can generalize the construction of the game-theoretic semantics to bigger fragments of the logic. We conjecture that the equivalence between PL and SL with timeline semantics can be extended up to the SL[EG] fragment, introduced by Gardy and al. [GBM20] as the largest fragment for which the model checking for SL (with timeline semantics) is 2-EXPTIME-*complete*.

5.A Missing Proofs of Section 5.3

In order to prove Theorem 5.3 and Theorem 5.4, we resort to the Meta semantics of PL, as introduced in Section 2.5.

5.A.1 Meta-PL

$\text{Rsp}_R(W)$ is the set of response functions $F \in \text{Rsp}_R$ such that for every $n \in \mathbb{N}$, for every $\chi_1, \chi_2 \in \text{Asg}$, if $\chi_1|_W =^{\leq n} \chi_2|_W$, then $F(\chi_1) =^{\leq n} F(\chi_2)$.

Because of the semantics uses functional assignments, all values of quantified variables are not necessarily stored in the hyperteam. Thus, we have to adapt the semantics of the tying operators. The *filtering under functional assignment* \mathcal{F} is defined by $\text{FLT}_{\mathcal{F}}(\mathfrak{X}, V, B) \stackrel{\text{def}}{=} \{X \in \mathfrak{X} \mid V \text{ is } B\text{-tied in } \text{EXT}(X, \mathcal{F})\}$ w.r.t the sets of variables $V \subseteq \text{VAR}(X)$ and bindings $B \subseteq \text{Binds}$.

The semantics rules for second-order PL are the following.

Definition 5.7. *The Meta semantic relation $\mathsf{G}, \mathfrak{X}, \mathcal{F} \models_{\text{Meta}}^{\alpha} \varphi$ for PL is inductively defined as in Definition 2.3 except for the rule Item 3 which is replaced by the rule Item 1'. and the rules for the Meta quantifiers, which are replaced by the rules Items 9'. and 10'. with the addition of the two rules and Items 11. and 12., where φ is a second-order PL formula, $\mathfrak{X} \in \text{HypTeams}(\text{FREE}\varphi)$ a hyperteam, \mathcal{F} a functional assignment, ψ an LTL formula, \flat a binding and $W \subseteq \text{Var}$ a finite subset of variables:*

- 1'. $\mathsf{G}, \mathfrak{X}, \mathcal{F} \models_{\text{Meta}}^{\alpha} \flat\psi$ if $\mathsf{G}, \text{EXT}(\mathfrak{X}, \mathcal{F}) \models^{\alpha} \flat\psi$;
- 9'. $\mathsf{G}, \mathfrak{X}, \mathcal{F} \models_{\text{Meta}}^{\alpha} \Sigma^W x. \varphi$ if there is $F \in \text{Rsp}_R(W)$ such that $\mathsf{G}, \mathfrak{X}, \mathcal{F}[x \mapsto F] \models_{\text{Meta}}^{\alpha} \varphi$;
- 10'. $\mathsf{G}, \mathfrak{X}, \mathcal{F} \models_{\text{Meta}}^{\alpha} \Pi^W x. \varphi$ if for all $F \in \text{Rsp}_R(W)$ we have $\mathsf{G}, \mathfrak{X}, \mathcal{F}[x \mapsto F] \models_{\text{Meta}}^{\alpha} \varphi$;
- 11. $\mathsf{G}, \mathfrak{X}, \mathcal{F} \models_{\text{Meta}}^{\exists\forall} \langle V \rangle \varphi$, if $\mathsf{G}, \text{FLT}_{\mathcal{F}}(\mathfrak{X}, V, B), \mathcal{F} \models_{\text{Meta}}^{\exists\forall} \varphi$;
- 12. $\mathsf{G}, \mathfrak{X}, \mathcal{F} \models_{\text{Meta}}^{\forall\exists} [V] \varphi$, if $\mathsf{G}, \text{FLT}_{\mathcal{F}}(\mathfrak{X}, V, B), \mathcal{F} \models_{\text{Meta}}^{\forall\exists} \varphi$;

The following theorems are directly extracted from Section 2.5. The first one assert the adequacy of the second-order semantics with respect to the original PL semantics on PL formulas.

Theorem 5.11. *For every CGS G , hyperteam \mathfrak{X} and every PL formula φ , we have the following.*

$$\mathsf{G}, \mathfrak{X}, \emptyset \models_{\text{Meta}}^{\alpha} \varphi \text{ iff } \mathsf{G}, \mathfrak{X} \models^{\alpha} \varphi$$

The following theorem relates second-order quantifiers with PL quantifiers when the sub-formula is quantifier-free. Its proof uses a generalization of Theorem 5.1 for **Meta**-PL by replacing the hyperteam \mathfrak{X} in the left part of each equivalence with $\text{EXT}(\mathfrak{X}, \mathcal{F})$. The proof of this generalized Theorem 5.1 is a simple induction on the quantifier-free PL formula, since \mathcal{F} is unused for all cases but the base case and the tyings. The base case is immediate, thus only the case of tyings is to be developed.

Lemma 5.1. *For all PL quantifier-free formulas φ , hyperteam \mathfrak{X} , functional assignment \mathcal{F} and set of variables $V \subseteq \text{VAR}(\mathfrak{X}) \cup \text{DOM}(\mathcal{F})$, if we have*

- 1'. $\mathfrak{X}', \mathcal{F} \models^{\exists\forall} \varphi$ iff there exists $X' \in \text{EXT}(\mathfrak{X}', \mathcal{F})$ such that $\chi' \models \varphi$, for all $\chi' \in X'$;
- 2'. $\mathfrak{X}', \mathcal{F} \models^{\forall\exists} \varphi$ iff, for all $X' \in \text{EXT}(\mathfrak{X}', \mathcal{F})$, it holds that $\chi' \models \varphi$, for some $\chi' \in X'$.

then,

1. $\mathfrak{X}, \mathcal{F} \models^{\exists\forall} \langle V \rangle \varphi$ iff there exists $X \in \text{EXT}(\mathfrak{X}, \mathcal{F})$ such that $\chi \models \langle V \rangle \varphi$, for all $\chi \in X$;
2. $\mathfrak{X}, \mathcal{F} \models^{\forall\exists} [V] \varphi$ iff, for all $X \in \text{EXT}(\mathfrak{X}, \mathcal{F})$, it holds that $\chi \models [V] \varphi$, for some $\chi \in X$.

Proof. Remark first that $\text{EXT}(\text{FLT}_{\mathcal{F}}(\mathfrak{X}, V, B), \mathcal{F}) = \text{FLT}(\text{EXT}(\mathfrak{X}, \mathcal{F}), V, B)$. Then, we have that

1. By Definition 5.7, $\mathfrak{X}, \mathcal{F} \models^{\exists\forall} \langle V \rangle \varphi$ iff $\mathbf{G}, \text{FLT}_{\mathcal{F}}(\mathfrak{X}, V, B), \mathcal{F} \models_{\text{Meta}}^{\exists\forall} \varphi$. By Hypothesis 1', we have that there exists $X' \in \text{EXT}(\text{FLT}_{\mathcal{F}}(\mathfrak{X}, V, B), \mathcal{F})$ such that $\chi' \models \varphi$, for all $\chi' \in X'$. Which in turn, amounts to the existence of $X' \in \text{FLT}(\text{EXT}(\mathfrak{X}, \mathcal{F}), V, B)$ such that $\chi' \models \varphi$, for all $\chi' \in X'$. And by definition of $\text{FLT}()$, there is $X \in \text{EXT}(\mathfrak{X}, \mathcal{F})$ such that $\chi \models \langle V \rangle \varphi$, for all $\chi \in X$.
2. The proof of this item follows the same reasoning as the previous one.

□

We now have the following theorem.

Theorem 5.12. *The following equivalences hold true, for all quantifier-free PL formulas ϕ , variables $x \in \text{Var}$, hyperteam $\mathfrak{X} \in \text{HypTeams}_{\geq}(V)$ with $V \stackrel{\text{def}}{=} \text{FREE}\phi \setminus \{x\}$.*

1. $\mathfrak{X}, \mathcal{F} \models_{\text{Meta}} \exists x. \phi$ iff $\mathfrak{X}, \mathcal{F} \models_{\text{Meta}} \Sigma^{\text{VAR}(\mathfrak{X})} x. \phi$.
2. $\mathfrak{X}, \mathcal{F} \models_{\text{Meta}} \forall x. \phi$ iff $\mathfrak{X}, \mathcal{F} \models_{\text{Meta}} \Pi^{\text{VAR}(\mathfrak{X})} x. \phi$.

With these two theorems, we can translate a prenex PL formula $\varphi\phi$ in second-order by reversing the order of φ .

5.A.2 Skolem semantics for Meta-PL

As the Second-order semantics for PL resemble a Tarski-like semantics, we can develop a Skolem semantics on it. A Skolem function is an object closer to a map used in the timeline semantics for SL, then establishing the Skolem semantics of second-order PL makes the first step toward the proof of the game-theoretic semantics of SL.

Given a quantifier prefix $\wp = Q_1x_1, \dots, Q_nx_n$, and a variable $x_i \in \text{VAR}(\wp)$, we define $\text{VAR}^{>x_i}(\wp) = \{x_j \in \text{VAR}(\wp) \mid j > i\}$ and $\text{VAR}^{<x_i}(\wp) = \{x_j \in \text{VAR}(\wp) \mid j < i\}$ and $\text{VAR}_{\forall}^{>x_i}(\wp) = \text{VAR}^{>x_i}(\wp) \cap \text{VAR}_{\forall}(\wp)$ and $\text{VAR}_{\forall}^{<x_i}(\wp) = \text{VAR}^{<x_i}(\wp) \cap \text{VAR}_{\forall}(\wp)$.

Given a quantifier prefix \wp , the domain of a Skolem function for \wp is defined bellow.

$$\text{RspMeta}(\wp, V) = \{R \in \text{Rsp}_R^V \mid \text{for all } x \in V, R(x) \in \text{Rsp}_R(\text{VAR}^{<x}(\wp))\}$$

When $V = \text{VAR}_{\forall}(\wp)$, we simply write $\text{RspMeta}(\wp)$.

Definition 5.8 (Skolem map). *Given a quantifier prefix $\wp = Q_1x_1, \dots, Q_nx_n$, a Skolem map Λ for this prefix is a function $\Lambda : \text{VAR}_{\exists}(\wp) \rightarrow \text{Rsp}_R^{\text{VAR}(\wp)} \rightarrow \text{Rsp}_R$ such that for every $x \in \text{VAR}_{\exists}(\wp)$, we have $\Lambda(x)$ is defined on $\text{RspMeta}(\text{VAR}_{\forall}^{>x}(\wp))$ and $\Lambda(x)(R) \in \text{Rsp}_R(\text{VAR}_{\forall}^{<x}(\wp))$.*

Remark that the input of $\Lambda(x)$ is the set of responses of variables universally quantified *after* x . This is because, when translating the quantifier prefix in second-order PL, the order of quantification is reversed.

Given, $R \in \text{RspMeta}(\wp)$, we write $\Lambda(R)$ the function defined by

$$\Lambda(R)(x_i) = \begin{cases} R(x_i) & \text{if } x_i \in \text{VAR}_{\forall}(\wp) \\ \Lambda(x_i)(R_{\upharpoonright_{\text{VAR}_{\forall}^{>x_i}(\wp)}}) & \text{otherwise} \end{cases}$$

Now we state the equivalence of the Skolem semantics.

Lemma 5.2. *For every PL sentence in prenex form $\wp = \wp\phi$ and functional assignment \mathcal{F} , we have that*

$$\{\{\emptyset\}\}, \mathcal{F} \models_{\text{Meta}} \wp \text{ iff there is a Skolem map } \Lambda \text{ for } \wp \text{ such that} \\ \text{for every } R \in \text{RspMeta}(\wp), \text{ we have } \{\{\emptyset\}\}, \mathcal{F} \uplus \Lambda(R) \models_{\text{Meta}} \phi \quad (5.1)$$

The proof is a simple induction on the quantifier prefix.

5.A.3 Proof of Theorem 5.4

Skolem maps are already enough to prove Theorem 5.4. We first prove that, given an $SL[C/DG]$ formula Φ if the sentence $SL2PL(\Phi)$ holds true for some CGS G , then $GTS_{SL}(\Phi)$ holds true in $GTS_{SL}(G, \Phi)$. The following theorem states this result for $SL[CG]$.

Theorem 5.13. *For a CGS G and an $SL[CG]$ formula Φ , the following holds true:*

$$- G \models^\alpha SL2PL(\Phi) \Rightarrow GTS_{SL}(G, \Phi) \models^\alpha GTS_{SL}(\Phi);$$

Proof. Let $\Phi = \wp \wedge_{b \in B} b \psi_b$ with $\wp = Q_1 X_1, \dots, Q_m X_m$. By Theorem 5.11, $G \models^\alpha SL2PL(\Phi)$ iff $G \models_{\text{Meta}} SL2PL(\Phi)$. By application of Lemma 5.2, there is a Skolem map Λ_{CG} for $\hat{\wp} = \wp_{GTS}(\Phi)$ such that for every $R_{CG} \in \text{RspMeta}(\hat{\wp})$, we have $G, \{\{\emptyset\}\}, \Lambda_{CG}(R_{CG}) \models_{\text{Meta}} T_{GTS}(\Phi) \phi_{GTS}(\Phi)$.

We define a Skolem map Λ_{1G} for $\wp \forall x_\odot$ such that $GTS_{SL}(G, \Phi), \{\{\emptyset\}\}, \Lambda_{1G}(R_{1G}) \models_{\text{Meta}} b_{\text{id}}(\diamond p_\exists) \vee ((\Box \neg p_\forall) \wedge \wedge_{b \in B} ((\Box q_b) \rightarrow \psi_b))$ for every $R_{1G} \in \text{RspMeta}(\wp \forall x_\odot)$. Given $R_{1G} \in \text{RspMeta}(\wp \forall x_\odot)$, for every assignment χ , we have $R_{1G}(\chi) \in (\text{Ac} \cup \text{BND}(\wp))^\omega$. To use Λ_{CG} , we need to enforce that the plans in output are only in Ac^ω . Thus, we define $\hat{R}_{1G} \in \text{RspMeta}(\wp \forall x_\odot)$ by replacing every occurrence of a binding in the plans output by Λ_{1G} with some fixed action in Ac . We have $\hat{R}_{1G}(x)(\chi) \in \text{Ac}^\omega$, we define $R_{CG} \in \text{RspMeta}(\emptyset, \hat{\wp})$ as follows. For every $y \in \text{VAR}(\wp)$, we fix a binding b_y such that $y \in \text{img}(b_y)$. For every x_b , given an assignment χ_{CG} in $\text{Asg}(\text{VAR}^{<x_b}(\hat{\wp}))$, we define $\chi_{1G} \in \text{Asg}(\text{VAR}^{<x}(\wp))$ as $\chi_{1G}(y) = \chi_{CG}(y_b)$ if $y \in \text{VAR}(b)$ and $\chi_{1G}(y) = \chi_{CG}(y_{b_y})$ otherwise. Then, we define $R_{CG}(x_b)(\chi_{CG}) = \hat{R}_{1G}(x)(\chi_{1G})$. We can check easily that the resulting function $R_{CG}(x_b)$ is a response function.

Now we can define $\Lambda_{1G}(R_{1G})$ with a step by step construction. Let $d \in \text{Ac}$ be a fixed action. We define a sequence of bindings $(b^n)_{n \in \mathbb{N}}$ and for every $i \in \llbracket 0, \dots, 1, m \rrbracket$ (recall that $m = |\wp|$), we define a sequence of actions $(a_n^i)_{n \in \mathbb{N}} \in \text{Ac}^\mathbb{N}$, a sequence of plans $(\rho_n^i)_{n \in \mathbb{N}} \in (\text{Ac}^\omega)^\mathbb{N}$ and a sequence of assignment $(\chi_n^{i, R_{1G}})_{n \in \mathbb{N}} \in \text{Asg}(\text{VAR}^{<x_i}(\wp))^\mathbb{N}$ as follows. We start with defining $\chi_n^{0, R_{1G}} = \emptyset$ for every $n \in \mathbb{N}$ the empty assignment. Then, for $i \in \llbracket 0, \dots, m-1 \rrbracket$,

- $b^0 = b_0$
- $b^{n+1} = R_{1G}(x_\odot)(\chi_n^{m, R_{1G}})[n]$
- if $Q_{i+1} = \forall$, then
 - $a_n^{i+1} = R_{1G}(x_{i+1})(\chi_n^{i, R_{1G}})[n]$
- if $Q_{i+1} = \exists$, then

- $a_n^{i+1} = \Lambda_{\text{CG}}(R_{\text{CG}})(x_{i+1})(\chi_n^{i,R_{1G}})[n]$
- $\rho_n^{i+1} = a_0^{i+1} \dots a_n^{i+1} d^\omega$
- $\chi_n^{i+1,R_{1G}} = \chi_n^{i,R_{1G}}[x_{i+1} \mapsto \rho_n^{i+1}]$

We define $\rho^i \in \text{Ac}^\omega$ as $\rho^i[n] = a_n^i$ and then $\Lambda_{1G}(R_{1G})(x_i) = \rho^i$.

Now we prove that if

1. for every $R_{\text{CG}} \in \text{RspMeta}(\hat{\wp})$, we have $\mathbf{G}, \{\{\emptyset\}\}, \Lambda_{\text{CG}}(R_{\text{CG}}) \models_{\text{Meta}}$
 $(\langle \{x_b \mid b \in B_x\} \rangle)_{x \in \text{VAR}_{\exists}(\wp)} (\langle \{x_b \mid b \in B_x\} \rangle)_{x \in \text{VAR}_{\forall}(\wp)} \wedge_{b \in B} b[x/x_b]_{x \in \text{img}(b)} \psi_b$, then
2. for every $R_{1G} \in \text{RspMeta}(\wp \forall x_{\odot})$, we have $\text{GTS}_{\text{SL}}(\mathbf{G}, \Phi), \{\{\emptyset\}\}, \Lambda_{1G}(R_{1G}) \models_{\text{Meta}}$
 $b_{\text{id}}(\diamond p_{\exists}) \vee ((\Box \neg p_{\forall}) \wedge \wedge_{b \in B} ((\Box q_b) \rightarrow \psi_b))$.

Let $\chi_0 = \emptyset$ and for every $i \in \llbracket 0, \dots, 1, m \rrbracket$, we define $\chi_i = \chi_{i-1}[x_i \mapsto \Lambda_{1G}(R_{1G})(\chi_{i-1})]$. Then, let $\chi = \chi_m[x_{\odot} \mapsto \Lambda_{1G}(R_{1G})(\chi_m)]$ and let $\pi = \text{PLAY}_{\text{GTS}_{\text{SL}}(\mathbf{G}, \Phi), b_{\text{id}}}(\chi)$. We have to prove that $\hat{\lambda}(\pi) \models_{\text{LTL}} (\diamond p_{\exists}) \vee ((\Box \neg p_{\forall}) \wedge \wedge_{b \in B} ((\Box q_b) \rightarrow \psi_b))$ where $\hat{\lambda}$ is the label function of $\text{GTS}_{\text{SL}}(\mathbf{G}, \Phi)$. Remark that, by definition, for every $x \in \text{VAR}_{\exists}(\wp)$, we have $\chi(x) \in \text{Ac}^\omega$, so for every $n \in \mathbb{N}$, we have $(\pi)_n \neq \hat{p}_{\forall}$. If there is $x_i \in \text{VAR}_{\forall}(\wp)$ such that $\chi(x_i) \notin \text{Ac}^\omega$, then there is $n \in \mathbb{N}$ such that $(\pi)_n = \hat{p}_{\exists}$. If $\chi(x_{\odot}) \notin B^\omega$, then there is $n \in \mathbb{N}$ such that $(\pi)_n = \hat{p}_{\odot}$ and then every $\Box q_b$ is false and thus, every implication is trivially true. Otherwise, let $B_\infty = \{b \in B_\Phi \mid \exists^\infty n \in \mathbb{N}, (\chi(x_{\odot}))_n = b\}$. If there is $n \in \mathbb{N}$ such that $(\chi(x_{\odot}))_n$ and b do not agree up to n for some binding $b \in B_\infty$, then $(\pi)_N = \hat{p}_{\odot}$ for some $N > n$. Then, we can assume that every two bindings in B_∞ agree up to every natural number. Consider $R_{\text{CG}} \in \text{RspMeta}(\hat{\wp})$ defined as before ($R_{\text{CG}}(x_b)(\chi_{\text{CG}}) = \hat{R}_{1G}(x)(\chi_{1G})$). Let $\dot{\chi}$ coherent with $\Lambda_{\text{CG}}(R_{\text{CG}})$ and let $\dot{\pi}(b) = \text{path}_{\mathbf{G}}(\dot{\chi}, b)$. For every two bindings $b^1, b^2 \in B_\infty$, thanks to the tyings in $\text{SL2PL}(\Phi)$, we have that $\dot{\chi}(x_{b^1}) = \dot{\chi}(x_{b^2})$. By immediate induction, we have that $\lambda(\dot{\pi}(b)) = \hat{\lambda}(\pi) \setminus B_\Phi$ for every binding $b \in B_\infty$. Then, for every $b \in B_\Phi$, if $\hat{\lambda}(\pi) \models \Box b$ then, $b \in B_\infty$. Then, $\hat{\lambda}(\pi) \setminus B_\Phi = \lambda(\dot{\pi}(b))$. However, $\lambda(\dot{\pi}(b)) \models_{\text{LTL}} \psi_b$, and so, $\hat{\lambda}(\pi) \models_{\text{LTL}} \psi_b$. Hence, for every $R_{1G} \in \text{RspMeta}(\wp \forall x_{\odot})$, we have $\text{GTS}_{\text{SL}}(\mathbf{G}, \Phi), \{\{\emptyset\}\}, \Lambda_{1G}(R_{1G}) \models_{\text{Meta}} b_{\text{id}}(\diamond p_{\exists}) \vee ((\Box \neg p_{\forall}) \wedge \wedge_{b \in B} ((\Box q_b) \rightarrow \psi_b))$. \square

Given a set of variables V and $R : V \rightarrow \text{Rsp}_R$, we say that an assignment $\chi \in \text{Asg}_{\geq}(V)$ is coherent with R if for every $x \in V$, we have $\chi(x) = R(x)(\chi)$.

Given a quantifier prefix \wp , let $\hat{\wp} = ((Q_x x_b)_{b \in B_x})_{x \in \text{VAR}(\wp)}$. Given a set of variables $V \subseteq \text{VAR}(\wp)$, a sequence of action profiles $u \in (\text{Ac}^{\text{VAR}(\wp)})^*$ and a set of bindings B , we define $\text{RspMeta}_{B,u}(\hat{\wp}, V)$ as follows.

$\text{RspMeta}_{B,u}(\hat{\wp}, V)$ is the set of $R \in \text{RspMeta}(\wp, V)$ such that, for every $\chi \in \text{Asg}(\text{VAR}(\hat{\wp}))$ coherent with R , for every $y \in V$, for every $b \in B$, if for every

$x \in \text{VAR}(b) \cap \text{VAR}^{<y}(\wp)$, we have $(\chi(x_b))_{<|u|} = u(x)$, then $(\chi(y_b))_{<|u|} = u(y)$.

Given an SL formula Φ , we define $\hat{B}_\Phi = \{b[x/x_b]_{x \in \text{img}(b)} \mid b \in \text{BND}(\Phi)\}$ and $\hat{b}_i = b[x/x_b]_{x \in \text{img}(b)}$ and $\hat{B}_x = \{\hat{b} \in \hat{B}_\Phi \mid x \in \text{VAR}(b)\}$. For the following proof, we consider that having $\hat{b} \in \hat{B}_\Phi$ implicitly gives the binding $b \in B_\Phi$ such that $\hat{b} = b[x/x_b]_{x \in \text{img}(b)}$.

Given a quantifier prefix \wp and an SL formula Φ and Λ_{DG} , a Skolem map for $\hat{\wp} = ((Q_x x_b)_{b \in B_x})_{x \in \text{VAR}(\wp)}$, we say that $u = u_1 \cdots u_n \in (\text{Ac}^{\text{VAR}(\hat{\wp})})^*$ is coherent with Λ_{DG} for a set of bindings B if for every $R_{\text{DG}} \in \text{RspMeta}(\hat{\wp})$ such that u is coherent with R_{DG} , we have that for every binding $b \in B$, we have $\text{path}_{\mathbf{G}, b}(u) = \text{path}_{\mathbf{G}, \hat{b}}(\chi)[:, |u| - 1]$ where χ is an assignment coherent with $\Lambda_{\text{DG}}(R_{\text{DG}})$.

For the simplicity of the following lemma, we define $\Phi_{\text{DG}} := T_\exists T_\forall \Psi_{\text{DG}}$ with $T_\exists = ((\{x_b \mid b \in B_x\}))_{x \in \text{VAR}_\exists(\wp)}$ and $T_\forall = ((\{x_b \mid b \in B_x\}))_{x \in \text{VAR}_\forall(\wp)}$ and $\Psi_{\text{DG}} = \bigvee_{b \in B} b[x/x_b]_{x \in \text{img}(b)} \psi_b$

Lemma 5.3. *For every disjunctive formula in prenex form $\Phi = \wp \bigvee_{b \in B} b \psi_b$, given Λ_{DG} , a Skolem map for $\hat{\wp} = ((Q_x x_b)_{b \in B_x})_{x \in \text{VAR}(\hat{\wp})}$ such that for every $R_{\text{DG}} \in \text{RspMeta}(\hat{\wp})$, we have $\mathbf{G}, \{\{\emptyset\}\}, \Lambda_{\text{DG}}(R_{\text{DG}}) \models_{\text{Meta}} \Phi_{\text{DG}}$, there is a function $f_{\text{bind}}^{\text{PL}} : (\text{Ac}^{\text{VAR}(\hat{\wp})})^* \rightarrow (2^{\hat{B}_\Phi} \setminus \emptyset)$ such that for every $u = u_1 \cdots u_n \in (\text{Ac}^{\text{VAR}(\hat{\wp})})^*$ that is coherent with Λ_{DG} for $f_{\text{bind}}^{\text{PL}}(u)$, and every action profile $\vec{d} \in \text{Ac}^{\text{VAR}(\hat{\wp})}$, we have*

1. $f_{\text{bind}}^{\text{PL}}(u\vec{d}) \subseteq f_{\text{bind}}^{\text{PL}}(u)$.
2. for every $\hat{b}, \hat{b}' \in f_{\text{bind}}^{\text{PL}}(u)$ we have $\text{path}_{\mathbf{G}, b}(u) = \text{path}_{\mathbf{G}, b'}(u)$. We denote by $\pi_{f_{\text{bind}}^{\text{PL}}(u)}$ such path.
3. For every $R_{\text{DG}} \in \text{RspMeta}_{f_{\text{bind}}^{\text{PL}}(u), u}(\hat{\wp})$ and for every χ coherent with $\Lambda(R_{\text{DG}})$, there is $\hat{b}_{i_0} \in f_{\text{bind}}^{\text{PL}}(u)$ such that if $\{x_b \mid b \in B_x\}$ are tied in χ for $\text{BND}(\wp)$ for every $x \in \text{VAR}_\forall(\wp)$, then $\{x_b \mid b \in B_x\}$ are tied in χ for $\text{BND}(\wp)$ for every $x \in \text{VAR}_\exists(\wp)$ and $\mathbf{G}, \chi \models_{\hat{b}_{i_0}} \psi_b$.

Proof. We define $f_{\text{bind}}^{\text{PL}}$ by induction on sequence of action profiles u and verify that $f_{\text{bind}}^{\text{PL}}$ satisfies the properties.

$$(\varepsilon) f_{\text{bind}}^{\text{PL}}(\varepsilon) = \hat{B}_\Phi.$$

2) For every binding $\hat{b} \in \hat{B}_\Phi$, we have $\text{path}_b(u) = p_i$.

3) By assumption, for every $R_{\text{DG}} \in \text{RspMeta}(\hat{\wp})$, we have $\mathbf{G}, \{\{\emptyset\}\}, \Lambda_{\text{DG}}(R_{\text{DG}}) \models_{\text{Meta}} T_\exists T_\forall \Psi_{\text{DG}}$. By application Theorem 5.1, we immediately obtain the result.

($u\vec{d}$) Suppose the following.

for every $R_{\text{DG}} \in \text{RspMeta}_{f_{\text{bind}}^{\text{PL}}(u), \pi_{f_{\text{bind}}}(u)}(\hat{\varphi})$ and for every χ coherent with $\Lambda_{\text{DG}}(R_{\text{DG}})$, there is $b_i \in f_{\text{bind}}^{\text{PL}}(u)$ such that if $\{x_b \mid b \in B_x\}$ are tied in χ for \hat{B}_Φ for every $x \in \text{VAR}_\forall(\varphi)$, then $\{x_b \mid b \in B_x\}$ are tied in χ for every $x \in \text{VAR}_\exists(\varphi)$ and $\mathbf{G}, \chi \models_{\hat{b}_i} \psi_b$ (5.2)

Let $p = \pi_{f_{\text{bind}}}(u)[|u| - 1]$. Consider the equivalence relation \sim on $f_{\text{bind}}^{\text{PL}}(u)$ defined by $\hat{b} \sim \hat{b}'$ if $\delta(p, \{a \mapsto \vec{d}(\hat{b}(a))\}) = \delta(p, \{a \mapsto \vec{d}(\hat{b}'(a))\})$. We denote by $[\hat{b}]$ the equivalence class of \hat{b} .

Claim 5.1. *There is $\hat{b}_{i_0} \in f_{\text{bind}}^{\text{PL}}(u)$ such that for every $R_{\text{DG}} \in \text{RspMeta}_{[\hat{b}_{i_0}], u\vec{d}}(\hat{\varphi})$, for every χ coherent with $\Lambda_{\text{DG}}(R_{\text{DG}})$, there is $\hat{b} \in [\hat{b}_{i_0}]$ such that, if $\{x_b \mid \hat{b} \in [\hat{b}_{i_0}]\}$ are tied in χ for $[\hat{b}_{i_0}]$ for every $x \in \text{VAR}_\forall(\varphi)$, then $\{x_b \mid \hat{b} \in [\hat{b}_{i_0}]\}$ are tied in χ for $[\hat{b}_{i_0}]$ for every $x \in \text{VAR}_\exists(\varphi)$ and $\text{PLAY}_{\mathbf{G}}(\chi, \hat{b}) \models_{\text{LTL}} \psi_b$*

Proof. We do a proof by contradiction. Suppose that for every binding $\hat{b}_{i_0} \in f_{\text{bind}}^{\text{PL}}(u)$ there is $R_{\text{DG}}^{\hat{b}_{i_0}} \in \text{RspMeta}_{[\hat{b}_{i_0}], u\vec{d}}(\hat{\varphi})$ such that, for every χ coherent with $\Lambda_{\text{DG}}(R_{\text{DG}}^{\hat{b}_{i_0}})$, we have

1. $\{x_b \mid \hat{b} \in [\hat{b}_{i_0}]\}$ are tied in χ for $[\hat{b}_{i_0}]$ for every $x \in \text{VAR}_\forall(\varphi)$ and
2. One of the following holds
 - (a) $\{x_b \mid \hat{b} \in [\hat{b}_{i_0}]\}$ are not tied in χ for $[\hat{b}_{i_0}]$ for some $x \in \text{VAR}_\exists(\varphi)$, or
 - (b) $\text{PLAY}_{\mathbf{G}}(\chi, \hat{b}) \not\models_{\text{LTL}} \psi_b$

Consider $R_{\text{DG}} \in \text{RspMeta}_{f_{\text{bind}}^{\text{PL}}(u), u}(\hat{\varphi})$ defined by $R_{\text{DG}}(x_b) = R_{\text{DG}}^{\hat{b}_{i_0}}(x_b)$ for every $x \in \text{VAR}_\forall(\varphi)$ and $\hat{b} \in f_{\text{bind}}^{\text{PL}}(u) \cap B_x$.

Let $\chi \in \text{Asg}(\text{VAR}(\hat{\varphi}))$ be coherent with $\Lambda_{\text{DG}}(R_{\text{DG}})$. We need to show that, for every $\hat{b} \in \hat{B}_\Phi$,

1. $\{x_b \mid \hat{b} \in f_{\text{bind}}^{\text{PL}}(u)\}$ are tied in χ for $f_{\text{bind}}^{\text{PL}}(u)$ for every $x \in \text{VAR}_\forall(\varphi)$
2. $\text{PLAY}_{\mathbf{G}}(\chi, \hat{b}) \not\models_{\text{LTL}} \psi_b$

Remark that, by assumption on Λ_{DG} , we have that if $\{x_b \mid \hat{b} \in f_{\text{bind}}^{\text{PL}}(u)\}$ are tied in χ for $f_{\text{bind}}^{\text{PL}}(u)$ for every $x \in \text{VAR}_\forall(\varphi)$, then the same holds for every $x \in \text{VAR}_\exists(\varphi)$.

1. Let $x \in \text{VAR}_\forall(\varphi)$.
 - Let \hat{b} and $\hat{b}' \in \hat{B}_x$ that agree up to $n \leq |u| - 1$ on χ and let $a \in \text{Ag}$. Then, by definition, we have that $\chi(\hat{b}(a)) = \Lambda_{\text{DG}}(R_{\text{DG}})(\hat{b}(a))(\chi) =$

$\Lambda_{\text{DG}}(R_{\text{DG}}^{\hat{b}})(\hat{b}(a))(\chi)$. Recall that by assumption, $u\vec{d}$ is coherent with Λ_{DG} and $R_{\text{DG}}^{\hat{b}} \in \text{RspMeta}_{[\hat{b}], u\vec{d}}(\hat{\phi})$. Then, $\Lambda_{\text{DG}}(R_{\text{DG}}^{\hat{b}})(\hat{b}(a))(\chi)[n] = u(x)[n]$. The same holds for \hat{b}' , hence $\chi(x_b) \stackrel{\leq n}{=} \chi(x_{b'})$.

— Let \hat{b} and $\hat{b}' \in \hat{B}_x$ that agree up to $n > |u| - 1$, then $\hat{b} \sim \hat{b}'$. Then by Item 1, we have x_b and $x_{b'}$ tied.

2. We have that there is $\dot{\chi}$ coherent with $R_{\text{DG}}^{\hat{b}}$ such that for every $x \in \text{VAR}(\hat{b})$, we have $\dot{\chi}(x_b) = \chi(x_b)$. Then $\text{PLAY}_{\text{G}}(\chi, \hat{b}) = \text{PLAY}_{\text{G}}(\dot{\chi}, \hat{b})$ and by Item 2b, we have $\text{PLAY}_{\text{G}}(\chi, \hat{b}) \not\models_{\text{LTL}} \psi_b$

This is in contradiction with Equation (5.2). \square

Let b_{i_0} given by Claim 5.1. We define $f_{\text{bind}}^{\text{PL}}(u\vec{d}) = [\hat{b}_{i_0}]$.

1. As $[\hat{b}_{i_0}] \subseteq f_{\text{bind}}^{\text{PL}}(u)$, we have $f_{\text{bind}}^{\text{PL}}(u\vec{d}) \subseteq f_{\text{bind}}^{\text{PL}}(u)$
2. By definition of \sim , we have $\text{path}(u\vec{d}, b) = \text{path}(u\vec{d}, b')$ for every $b, b' \in [\hat{b}_{i_0}]$
3. Claim 5.1 proves immediately this point.

\square

Theorem 5.14. *For a CGS G and an SL formula in disjunctive prenex form Φ , the following hold true:*

- $\text{G} \models^{\alpha} \phi_{\text{GTS}}(\Phi) \Rightarrow \text{GTS}_{\text{SL}}(\text{G}, \Phi) \models^{\alpha} \text{GTS}_{\text{SL}}(\Phi)$;

Proof. Let $\Phi = \wp \bigvee_{b \in B} b\psi_b$ with $\wp = Q_1 X_1, \dots, Q_m X_m$. By Theorem 5.11, $\text{G} \models^{\alpha} \phi_{\text{GTS}}(\Phi)$ iff $\text{G} \models_{\text{Meta}} \phi_{\text{GTS}}(\Phi)$. By application of Lemma 5.2, there is a Skolem map Λ_{DG} for $\hat{\phi} = ((Q_x x_b)_{b \in B_x})_{x \in \text{VAR}(\wp)}$ such that for every $R_{\text{DG}} \in \text{RspMeta}(\hat{\phi})$, we have $\text{G}, \{\{\emptyset\}\}, \Lambda_{\text{DG}}(R_{\text{DG}}) \models_{\text{Meta}} T\exists T\forall \Psi_{\text{DG}}$.

We define a Skolem map $\Lambda_{1\text{G}}$ for $\wp\exists x_{\odot}$ as follows. Given $R_{1\text{G}} \in \text{RspMeta}(\wp\exists x_{\odot})$, for every assignment χ and variable $x_i \in \text{VAR}_{\forall}(\wp)$, we have $R_{1\text{G}}(x_i)(\chi) \in (\text{Ac} \cup \text{BND}(\wp))^{\omega}$. To use Λ_{DG} , we need to enforce that the plans in output are only in Ac^{ω} . Thus, we define $\hat{R}_{1\text{G}} \in \text{RspMeta}(\emptyset, \wp\exists x_{\odot})$ by replacing every occurrence of a binding in the plans output by $\Lambda_{1\text{G}}$ with some fixed action in Ac . We have $\hat{R}_{1\text{G}}(x_i)(\chi) \in \text{Ac}^{\omega}$, we define $R_{\text{DG}} \in \text{RspMeta}(\hat{\phi})$ as follows. For every $y \in \text{VAR}^{<x}(\wp)$, we fix a binding b_y such that $y \in \text{VAR}(b_y)$. For every x_b , given an assignment χ_{DG} in $\text{Asg}(\text{VAR}^{<x_b}(\hat{\phi}))$, we define $\chi_{1\text{G}} \in \text{Asg}(\text{VAR}^{<x}(\wp))$ as $\chi_{1\text{G}}(y) = \chi_{\text{DG}}(y_b)$ if $y \in \text{VAR}(b)$ and, otherwise, $\chi_{1\text{G}}(y) = \chi_{\text{DG}}(y_{b_y})$. We then can complete $\chi_{1\text{G}}$ with the values of $x_{b'}$ for binding b' such that $x_{b'} \in \text{VAR}^{<x_b}(\hat{\phi})$. Then, we define $R_{\text{DG}}(x_b)(\chi_{\text{DG}}) = \hat{R}_{1\text{G}}(x)(\chi_{1\text{G}})$. We can check easily that the resulting function $R_{\text{DG}}(x_b)$ is a response function.

Now we can define $\Lambda_{1G}(x_\odot)$ thanks to Lemma 5.3. The proof is basically the same as the one for Theorem 5.13 with the summarized step by step construction leading to: $\Lambda_{1G}(x)(R_{1G})(\chi)[n] = \Lambda_{DG}(x_{b_n})(R_{DG})(\dot{\chi})[n]$ with $\dot{\chi}$ defined by $\dot{\chi}(x_b) = \chi(x)$ and $b_n = \Lambda_{1G}(x_\odot)(\chi)[n]$. \square

We need a last preliminary result to prove Theorem 5.4.

Proposition 5.2. *For every Φ in $SL[C/DG]$, we have that $GTS_{SL}(\mathbf{G}, \Phi) \models^\alpha GTSS_{SL}(\Phi)$ iff $GTS_{SL}(\mathbf{G}, \hat{\Phi}) \not\models_{PL} GTSS_{SL}(\hat{\Phi})$ where $\hat{\Phi}$ is the negation of Φ in positive normal form.*

Proof. Remark that the difference between $GTS_{SL}(\mathbf{G}, \Phi)$ and $GTS_{SL}(\mathbf{G}, \neg\Phi)$ is that the sink states \hat{p}_\exists and \hat{p}_\forall are swapped. Also, the difference between $GTS_{SL}(\neg\Phi)$ and $\neg GTSS_{SL}(\Phi)$, after simplification through De Morgan’s laws, is that p_\exists and p_\forall are swapped. From this remark, the proof of the theorem is straightforward.

We inherit proofs from [BBDM23a] for the De Morgan’s laws in PL. \square

Theorem 5.4. $\mathbf{G} \models GTSS(\Phi)$ iff $GTS_{SL}(\mathbf{G}, \Phi) \models GTSS_{SL}(\Phi)$, for all $SL[C/DG]$ sentences Φ .

Proof. We prove each implication independently.

(\Rightarrow) This first implication is an immediate application of Theorems 5.13 and 5.14.

(\Leftarrow) We prove the contraposition. Suppose that $\mathbf{G} \not\models_{PL} SL2PL(\Phi)$. We do a case study on the fragment hosting Φ . Suppose first that $\Phi \in SL[CG]$. Thanks to PL determinacy, we obtain $\mathbf{G} \models^\alpha \neg SL2PL(\Phi)$. By immediate induction on Φ , we have that $\mathbf{G} \models^\alpha SL2PL(\neg\Phi)$. Using De Morgan’s law in SL, we have that $\neg\Phi$ is equivalent to a disjunctive formula $\hat{\Phi}$ in positive normal form. Then, by Theorem 5.14, we obtain $GTS_{SL}(\mathbf{G}, \hat{\Phi}) \models^\alpha GTSS_{SL}(\hat{\Phi})$. By Proposition 5.2, it amounts to $GTS_{SL}(\mathbf{G}, \Phi) \not\models_{PL} GTSS_{SL}(\Phi)$. The case $\Phi \in SL[CG]$ is proven symmetrically. \square

5.A.4 Realizable meta map

We now focus on the material for proving Theorem 5.3. The approach is similar to the one for Theorem 5.4: we prove that the truth of an $SL[C/DG]$ formula Φ in a CGS \mathbf{G} under timeline semantics entails the truth of the corresponding formula $GTS_{SL}(\Phi)$ in the CGS $GTS_{SL}(\mathbf{G}, \Phi)$. The reciprocal is obtained through determinacy of SL and the

duality result of Proposition 5.2. In this subsection, we propose an interpretation of the SL timeline semantics in the setting of PL.

First, we introduce a new relation on assignments. Given two assignments χ_1 and $\chi_2 \in \mathbf{Asg}(V)$, we define $\chi_1 \stackrel{n}{\sim}_W \chi_2$ by $\chi_1 =^{<n} \chi_2$ and $\chi_1 \upharpoonright (V \setminus W) =^{\leq n} \chi_2 \upharpoonright (V \setminus W)$. This relation defines the concept of *strict dependency* for response functions. A response function F has a strict dependency toward a set of variable $W \subseteq V$ if, for every assignments χ_1 and $\chi_2 \in \mathbf{Asg}_{\supseteq}(V)$, for every $n \in \mathbb{N}$, if $\chi_1 \stackrel{n}{\sim}_W \chi_2$, then $F(\chi_1) =^{\leq n} F(\chi_2)$. The set of responses strictly dependent on W is denoted by $\mathbf{Rsp}_S(W)$

Given a quantifier prefix \wp and set of variables $V \subseteq \mathbf{VAR}(\wp)$, we define $\mathbf{Maps}_{\wp}(V) = \{R \in \mathbf{Rsp}_R^V \mid \forall x \in V \text{ we have } R(x) \in \mathbf{Rsp}_S(\mathbf{VAR}^{>x}(\wp))\}$. Remark that we have the following inclusion: $\mathbf{RspMeta}(\wp) \subseteq \mathbf{Maps}_{\wp}(\mathbf{VAR}_{\forall}(\wp))$.

Definition 5.9 (Realizable meta map). *Given a quantifier prefix \wp , a meta map Δ for this prefix is a function $\Delta : \mathbf{Maps}_{\wp}(\mathbf{VAR}_{\forall}(\wp)) \rightarrow \mathbf{Maps}_{\wp}(\mathbf{VAR}(\wp))$ such that for every $x \in \mathbf{VAR}_{\forall}(\wp)$, we have $\Delta(R)(x) = R(x)$ for every $R \in \mathbf{Maps}_V$.*

Such a map is said realizable when for any $R \in \mathbf{Maps}_{\wp}(\mathbf{VAR}_{\forall}(\wp))$ and $R' \in \mathbf{Maps}_{\wp}(\mathbf{VAR}_{\forall}(\wp))$, for every $x \in \mathbf{VAR}_{\exists}(\wp)$, for every $k \in \mathbb{N}$, for every $\chi_1, \chi_2 \in \mathbf{Asg}$ such that $\chi_1 \upharpoonright \mathbf{VAR}(x) =^{\leq k} \chi_2 \upharpoonright \mathbf{VAR}(x)$, if

1. *for every $y \in \mathbf{VAR}_{\forall}^{<x_i}(\wp)$, we have $R(y)(\chi_1) =^{\leq k} R'(y)(\chi_2)$ and*
2. *for every $y \in \mathbf{VAR}_{\forall}^{>x_i}(\wp)$, we have $R(y)(\chi_1) =^{\leq k-1} R'(y)(\chi_2)$,*

then, $\Delta(R)(x_i)(\chi_1) =^{\leq k} \Delta(R')(x_i)(\chi_2)$.

The difference between a realizable meta map and a Skolem map is that the response function $\Delta(x)(R)$ formally depends on the value of variables quantified after x but only on their strict past. This added information is mitigated by the *realizable* condition.

Proposition 5.3. *Given a team $X \subseteq \mathbf{Asg}(V)$, a vector of responses $R \in \mathbf{RspMeta}(\wp)$ and a Skolem map Λ , there is a bijection between X and $\mathbf{EXT}(X, \Lambda(R))$.*

Proof. Let $f : \mathbf{EXT}(X, \Lambda(R)) \rightarrow X$ defined as $f(\chi) = \chi \upharpoonright V$. We prove that f is a bijection.

- 1) f is injective. Let χ_1 and $\chi_2 \in \mathbf{EXT}(X, \Lambda(R))$. Suppose that $\chi_1 \upharpoonright V = \chi_2 \upharpoonright V$. Toward contradiction, suppose that there is $x \in \mathbf{VAR}(\wp)$ such that $\chi_1(x) \neq \chi_2(x)$. Let $i_0 = \min\{i \in \llbracket 0, \dots, 1, n \rrbracket \mid \chi_1(x_i) \neq \chi_2(x_i)\}$. Then, we have that $\chi_1 \upharpoonright V \cup \mathbf{VAR}^{<x_{i_0}}(\wp) = \chi_2 \upharpoonright V \cup \mathbf{VAR}^{<x_{i_0}}(\wp)$. However, by Definition 5.8, we have $\Lambda(R)(x_{i_0}) \in \mathbf{Rsp}_R(V \cup \mathbf{VAR}^{<x_{i_0}}(\wp))$. Then, we have $\chi_1(x_{i_0}) = \Lambda(R)(x_{i_0})(\chi_1) = \Lambda(R)(x_{i_0})(\chi_2) = \chi_2(x_{i_0})$ which is a contradiction.

2) f is surjective. Let $\chi \in X$. For $i \in \llbracket 0, \dots, 0, n \rrbracket$, define $\dot{\chi}_i \in \mathbf{Asg}(V \cup \text{VAR}^{<x_i}(\varphi))$ as follows.

— $\dot{\chi}_0 = \chi$,

— for $i \in \llbracket 0, \dots, 0, n-1 \rrbracket$, we set $\dot{\chi}_{i+1} = \dot{\chi}_i[x_{i+1} \mapsto \Lambda(R)(x_i)(\dot{\chi}_i)]$

It is straightforward to see that $\dot{\chi}_n \in \text{EXT}(X, \Lambda(R))$ and that $f(\dot{\chi}_n) = \chi$.

□

Proposition 5.4. *Given a team $X \subseteq \mathbf{Asg}(V)$, a PL map $R \in \mathbf{Maps}_\varphi(\text{VAR}_\forall(\varphi))$ and a Realizable meta map Δ , there is a bijection between X and $\text{EXT}(X, \Delta(R))$.*

Proof. Let $f : \text{EXT}(X, \Delta(R)) \rightarrow X$ defined as $f(\chi) = \chi|_V$. We prove that f is a bijection.

1) f is injective. Let χ_1 and $\chi_2 \in \text{EXT}(X, \Delta(R))$. Suppose that $\chi_1|_V = \chi_2|_V$. Toward contradiction, suppose that there is $x \in \text{VAR}(\varphi)$ such that $\chi_1(x) \neq \chi_2(x)$. Let $t_0 = \min\{t \in \mathbb{N} \mid \chi_1(x)[t] \neq \chi_2(x)[t] \text{ for some } x \in \text{VAR}(\varphi)\}$. Let $i_0 = \min\{i \in \llbracket 0, \dots, 1, n \rrbracket \mid \chi_1(x_i)[t_0] \neq \chi_2(x_i)[t_0]\}$. So, we have that $\chi_1 \stackrel{t_0}{\sim}_{\text{VAR}^{<x_{i_0}}(\varphi)} \chi_2$. However, $\Delta(R)(x_{i_0}) \in \mathbf{Rsp}_S(\text{VAR}^{<x_{i_0}}(\varphi))$. Then, we have $\chi_1(x_{i_0})[t_0] = \Delta(R)(x_{i_0})(\chi_1)[t_0] = \Delta(R)(x_{i_0})(\chi_2)[t_0] = \chi_2(x_{i_0})[t_0]$ which is a contradiction.

2) f is surjective. Let $\chi \in X$ and let $d \in \mathbf{Ac}$. For $t \in \mathbb{N}$ and $i \in \llbracket 0, \dots, 1, n \rrbracket$, define $\dot{\chi}_i^t \in \mathbf{Asg}(V \cup \text{VAR}(\varphi))$ and the initial $\dot{\chi}_0^0 \in \mathbf{Asg}(V \cup \text{VAR}(\varphi))$ as follows.

— Let $\dot{\chi}_0^0 = \chi[x \mapsto d^\omega]$,

— for $i \geq 1$, we set $\dot{\chi}_{i+1}^t = \dot{\chi}_i^t[x_{i+1} \mapsto \Delta(R)(x_i)(\dot{\chi}_i^t)]$

— for $t \geq 0$, we set $\dot{\chi}_1^{t+1} = \dot{\chi}_n^t[x_1 \mapsto \Delta(R)(x_1)(\dot{\chi}_n^t)]$

We now define $\dot{\chi}(x_i)[t] = \dot{\chi}_i^t[x_i][t]$. It is straightforward to see that $\dot{\chi} \in \text{EXT}(X, \Delta(R))$ and that $f(\dot{\chi}_n) = \chi$.

□

Remark that these two propositions imply that on the trivial team containing only the trivial empty assignment, there is only one assignment obtained by the application of a Skolem map or of a realizable meta map.

The existence of a realizable meta map that satisfies a formula entails the existence of a Skolem map that satisfies the formula.

Lemma 5.4. *Given a quantifier prefix φ , for every realizable meta map Δ for this prefix, there is a Skolem map Λ such that for every quantifier-free formula ϕ , if for every $R' \in \mathbf{Maps}_{\text{VAR}_\forall(\varphi)}$, we have $\mathbf{G}, \text{EXT}(\{\{\emptyset\}\}, \Delta(R')) \models^\alpha \phi$, then for every $R \in \mathbf{RspMeta}(\varphi)$, we have $\mathbf{G}, \text{EXT}(\{\{\emptyset\}\}, \Lambda(R)) \models^\alpha \phi$.*

Proof. Let $x \in \text{VAR}_{\exists}(\varphi)$ and $R \in \text{RspMeta}(\text{VAR}^{>x}(\varphi))$. Given $\dot{\chi} \in \text{Asg}(V)$ with $\text{VAR}_{\forall}^{<x}(\varphi) \subseteq V$, we define $\Lambda(x)(R)(\dot{\chi})$. For every $y \in \text{VAR}_{\forall}^{<x}(\varphi)$ we introduce $F_y^{\dot{\chi}} \in \text{Rsp}_R$ defined by $F_y^{\dot{\chi}}(\chi) = \dot{\chi}(y)$ for every $\chi \in \text{Asg}$. Then, let $R^{\dot{\chi}} = R[y \mapsto F_y^{\dot{\chi}}]_{y \in \text{VAR}_{\forall}^{<x}(\varphi)}$. We can now define the following.

$$\Lambda(x)(R)(\dot{\chi}) = \Delta(R^{\dot{\chi}})(x)(\dot{\chi}|_{\text{VAR}^{<x}(\varphi)})$$

Claim 5.2. $\Lambda(x)(R) \in \text{Rsp}_S(\text{VAR}^{<x}(\varphi))$.

Proof. $\Lambda(x)(R)$ depends only on $\text{VAR}^{<x_i}(\varphi)$ thanks to the restriction when applying $\Delta(R^{\dot{\chi}})(x)$. We can remark also that two assignment that agree on $\text{VAR}_{\forall}^{<x_i}(\varphi)$ lead to the definition of the same $R^{\dot{\chi}}$.

Now, let $\dot{\chi}_1, \dot{\chi}_2 \in \text{Asg}(V)$ such that $\dot{\chi}_1 \stackrel{\leq n}{=} \dot{\chi}_2$ for some $n \in \mathbb{N}$. Then, for every $y \in \text{VAR}_{\forall}^{<x}(\varphi)$, we have $R^{\dot{\chi}_1}(y)(\dot{\chi}_1) = \dot{\chi}_1(y) \stackrel{\leq n}{=} \dot{\chi}_2(y) = R^{\dot{\chi}_2}(y)(\dot{\chi}_2)$. Furthermore, since $R^{\dot{\chi}_1}|_{\text{VAR}_{\forall}^{>x_i}(\varphi)} = R = R^{\dot{\chi}_2}|_{\text{VAR}_{\forall}^{>x_i}(\varphi)}$, we have that, for every $y \in \text{VAR}_{\forall}^{>x_i}(\varphi)$, it holds that $R^{\dot{\chi}_1}(y)(\dot{\chi}_1) \stackrel{\leq n-1}{=} R^{\dot{\chi}_2}(y)(\dot{\chi}_2)$. Then, since Δ is a realizable meta map, we have that $\Delta(R^{\dot{\chi}_1})(x)(\dot{\chi}_1|_{\text{VAR}(\mathfrak{X}) \cup \text{VAR}^{<x_i}(\varphi)}) \stackrel{\leq n}{=} \Delta(R^{\dot{\chi}_2})(x)(\dot{\chi}_2|_{\text{VAR}(\mathfrak{X}) \cup \text{VAR}^{<x_i}(\varphi)})$. \square

Claim 5.3. For every $R \in \text{RspMeta}(\varphi)$, variable $x \in \text{VAR}(\varphi)$, and assignment $\chi \in \text{EXT}(\emptyset, \Delta(R))$ with \emptyset the empty assignment, we have $\Lambda(R)(x)(\chi) = \Delta(R)(x)(\chi)$.

Proof. The property is immediate for $x \in \text{VAR}_{\forall}(\varphi)$. For $x \in \text{VAR}_{\exists}(\varphi)$, we have $\Lambda(R)(x)(\chi) = \Delta(R^x)(x)(\chi|_{\text{VAR}^{<x}(\varphi)})$. We have that,

- for every $y \in \text{VAR}_{\forall}^{<x_i}(\varphi)$, we have $R^x(y)(\chi|_{\text{VAR}^{<x}(\varphi)}) = \chi(y)$ and by definition of $\text{EXT}(X, \Delta(R))$, we have $\chi(y) = R(y)(\chi)$ and
- for every $y \in \text{VAR}_{\forall}^{>x_i}(\varphi)$, we have $R^x(y) = R(y)$,

Then, by Definition 5.9, we have $\Delta(R^x)(x)(\chi|_{\text{VAR}^{<x}(\varphi)}) = \Delta(R)(x)(\chi)$. \square

Thanks to Claim 5.2, the Λ defined is indeed a Skolem map. Thanks to Claim 5.3, we have $\text{EXT}(\emptyset, \Delta(R)) \subseteq \text{EXT}(\emptyset, \Lambda(R))$. Thanks to Proposition 5.4 and Proposition 5.3, we have equality of size. Hence $\text{EXT}(\{\{\emptyset\}\}, \Delta(R)) = \text{EXT}(\{\{\emptyset\}\}, \Lambda(R))$. \square

5.A.5 Proof of Theorem 5.3

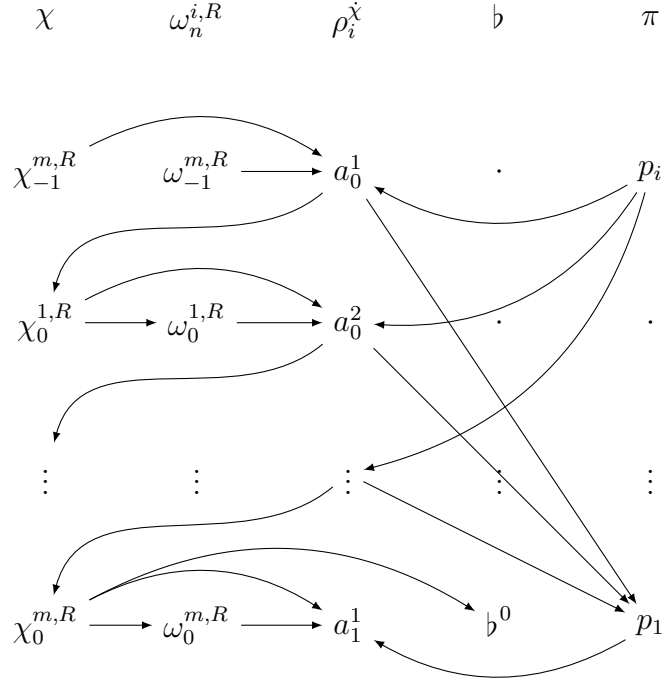
We now link timeline maps and realizable meta map in the same way we linked realizable meta map to Skolem maps. We start by the conjunctive fragment of SL.

Lemma 5.5. *Given a quantifier prefix $\varphi = Q_1 X_1, \dots, Q_m X_m$, for every timeline map θ for this prefix, there is a Realizable meta map Δ for $\varphi \forall x_\odot$ such that for every quantifier-free conjunctive formula $\phi = \bigwedge_{b \in B} b \psi_b$, if for every $\mathbf{m} \in \mathbf{Strat}^{V_\forall}$, we have $\mathbf{G}, \theta(\mathbf{m}) \models_{SL} \phi$, then for every $R \in \mathbf{Maps}_\varphi(\mathbf{VAR}_\forall(\varphi) \cup x_\odot)$, we have $GTS_{SL}(\mathbf{G}, \Phi), EXT(\{\{\emptyset\}\}, \Delta(R)) \models^\alpha b_{id}(\diamond p_\exists) \vee ((\square \neg p_\forall) \wedge \bigwedge_{b \in B} ((\square q_b) \rightarrow \psi_b))$.*

Proof. We define Δ step by step. Given a PL map $R \in \mathbf{Maps}_{\varphi \forall x_\odot}(\mathbf{VAR}_\forall(\varphi) \cup \{x_\odot\})$, we fix an action $d \in \mathbf{Ac}$ and we consider an assignment $\dot{\chi} \in \mathbf{Asg}(\mathbf{VAR}(\varphi))$, a sequence of paths $(p_n^{\dot{\chi}, R})_{n \in \mathbb{N}}$ in \mathbf{G} and of bindings $(b^n)_{n \in \mathbb{N}} \in \mathbf{BND}(\phi)^\mathbb{N}$ and, for every variable index $i \in \llbracket 0, \dots, 1, m \rrbracket$, sequences of strategic maps $(\omega_n^{i, R})_{n \in \mathbb{N}} \in ((\mathbf{Strat}^{V_\forall})^{\mathbf{VAR}(\varphi)})^\mathbb{N}$, of assignments $(\chi_n^{i, R})_{n \in \mathbb{N}} \in \mathbf{Asg}(\mathbf{VAR}(\varphi))$, of actions $(a_n^i)_{n \in \mathbb{N}} \in (\mathbf{Ac}^{\mathbf{VAR}^{< x_i}(\varphi)})^\mathbb{N}$ and of plans $(\rho_n^y)_{n \in \mathbb{N}} \in (\mathbf{Plans}^{\mathbf{VAR}^{< x_i}(\varphi)})^\mathbb{N}$ defined co-inductively as follows, where π is a path in \mathbf{G} and $X \in \mathbf{VAR}(\varphi)$.

- $\omega_0^{0, R}(X) : \pi \mapsto d$
- if $Q_1 = \exists$, then $\omega_{n+1}^{1, R} = \omega_n^{m, R}$
- if $Q_1 = \forall$, then $\omega_{n+1}^{1, R} : \begin{cases} X_1 & \mapsto \begin{cases} p_n^{\dot{\chi}, R} & \mapsto R(X_1)(\chi_{n+1}^{i, R})[n+1] \\ \pi \neq p_n^{\dot{\chi}, R} & \mapsto \omega_n^{m, R}(X_1)(\pi) \end{cases} \\ X \neq X_1 & \mapsto \omega_n^{m, R}(X) \end{cases}$
- for $i+1 \in I_\exists$, we set $\omega_n^{i+1, R} = \omega_n^{i, R}$
- for $i+1 \in I_\forall$, we set $\omega_n^{i+1, R} : \begin{cases} X_{i+1} & \mapsto \begin{cases} p_n^{\dot{\chi}} & \mapsto R(X_{i+1})(\chi_n^{i, R})[n] \\ \pi \neq p_n^{\dot{\chi}} & \mapsto \omega_n^{i, R}(X_{i+1})(\pi) \end{cases} \\ X \neq X_{i+1} & \mapsto \omega_n^{i, R}(X) \end{cases}$
- For $i \in I_\exists$, we set $a_n^i = \theta(\omega_n^{i, R})(X_i)(p_n^{\dot{\chi}, R})$
- For $i \in I_\forall$, we set $a_n^i = R(x_i)(\chi_n^{i, R})[n]$
- $\rho_0^i = a_0^i d^\omega$
- $\rho_{n+1}^i = \rho_{n+1}^i[n+1 \mapsto a_{n+1}^i]$
- $\chi_0^{0, R} = x_i \mapsto \rho_0^i$
- $\chi_{n+1}^{1, R} = \chi_n^{m, R}[x_1 \mapsto \rho_n^1]$
- $\chi_n^{i+1, R} = \chi_n^{i, R}[x_i \mapsto \rho_n^i]$
- $p_0^{\dot{\chi}, R} = p_i$
- $p_{n+1}^{\dot{\chi}, R} = p_n^{\dot{\chi}, R} \cdot p_{n+1}$
- $b^n = R(x_\odot)(\chi_n^{m, R})$
- $p_0 = p_i$
- $p_{n+1} = \delta(p, \{a \in \mathbf{Ag} \mapsto a^{b^n(a)}\})$

We then define for $i \in I_\exists$, we set $\rho_i^{\dot{\chi}}[n] = a_n^i$ and then, we define $\Delta(R)(x_i)(\dot{\chi}) = \rho_i^{\dot{\chi}}$.



Claim 5.4. For every $n \in \mathbb{N}$ and for every $i \in \llbracket 0, \dots, m \rrbracket$, if $\dot{\chi}$ is coherent with $\Delta(R)$, then $\chi_n^{i,R} =^{<n} \dot{\chi}$.

Proof. The proof is done by induction on $n \in \mathbb{N}$.

($n = 0$) The relation $=^{<0}$ is universal.

($n + 1$) Suppose that the property holds for some $n \in \mathbb{N}$. Toward contradiction, assume that $\chi_{n+1}^{i,R} \neq^{<n+1} \dot{\chi}$ for some $i \in \llbracket 0, \dots, 1, m \rrbracket$. Then, let $j_0 = \min\{i \in \llbracket 0, \dots, 1, m \rrbracket \mid \chi_{n+1}^{i,R}(x_i)[n] \neq \dot{\chi}(x_i)[n]\}$. We do a case study.

- If $x_{j_0} \in \text{VAR}_{\forall}(\varphi)$, then $\chi_{n+1}^{i,R}(x_{j_0})[n] = R(x_{j_0})(\chi_n^{i,R})[n]$. However, by definition of j_0 and by inductive hypothesis, we have $\chi_{n+1}^{i,R}|_{\text{VAR}^{<x_{j_0}}(\varphi)} =^{<n+1} \dot{\chi}|_{\text{VAR}^{<x_{j_0}}(\varphi)}$. Then, since $R(x_{j_0}) \in \text{Rsp}_S(\text{VAR}^{<x_{j_0}}(\varphi))$, we have $R(x_{j_0})(\chi_n^{i,R})[n] = R(x_{j_0})(\dot{\chi})[n]$. And since $\dot{\chi}$ is coherent with $\Delta(R)$, we have $R(x_{j_0})(\dot{\chi})[n] = \dot{\chi}(x_{j_0})[n]$, which is a contradiction.
- If $x_{j_0} \in \text{VAR}_{\exists}(\varphi)$, then, by definition, $\chi_{n+1}^{i,R}(x_{j_0})[n] = a_n^{j_0} = \Delta(R)(x_{j_0})(\dot{\chi})[n]$. But we have that $\dot{\chi}$ is coherent with $\Delta(R)$ so $[\Delta](R)(x_{j_0})(\dot{\chi})n = \dot{\chi}(x_{j_0})[n]$. Thus, we have $[\chi_n^{i,R}][n+1](x_{j_0})n = \dot{\chi}(x_{j_0})[n]$ which is a contradiction with the definition of j_0 .

We proved the property by induction. □

Claim 5.5. Δ is a Realizable meta map.

Proof. Let $R, R' \in \mathbf{Maps}_{\text{VAR}_{\forall}(\varphi) \cup x_{\odot}}$ and $x_i \in \text{VAR}_{\exists}(\varphi)$ and $\chi_1, \chi_2 \in \mathbf{Asg}$ such that χ_1 is coherent with $\Delta(R)$ and χ_2 is coherent with $\Delta(R')$. We prove by induction over $n \in \mathbb{N}$ that if

1. $\chi_1 \upharpoonright_{\text{VAR}(\mathfrak{x})} =^{\leq n} \chi_2 \upharpoonright_{\text{VAR}(\mathfrak{x})}$
2. for every $y \in \text{VAR}_{\forall}^{< x_i}(\varphi)$, we have $R(y)(\chi_1) =^{\leq n} R'(y)(\chi_2)$ and
3. for every $y \in \text{VAR}_{\forall}^{> x_i}(\varphi)$, we have $R(y)(\chi_1) =^{\leq n-1} R'(y)(\chi_2)$,

then $\Delta(R)(x_i)(\chi_1) =^{\leq n} \Delta(R')(x_i)(\chi_2)$.

($n = 0$) Assume that Items 1 and 2 hold for $n = 0$ (Item 3 is trivial for $n = 0$). By definition of Δ , we have $(\Delta(R)(x_i)(\chi_1))[0] = \theta(\omega_0^{i,R})(X_i)(p_i)$ and $(\Delta(R')(x_i)(\chi_1))[0] = \theta(\omega_0^{i,R'})(X_i)(p_i)$. Toward contradiction, suppose that $\omega_0^{i,R} \neq \omega_0^{i,R'}$. Let $j_0 = \min\{j \leq i \mid \omega_0^{j,R} \neq \omega_0^{j,R'}\}$. By definition, $j_0 = 0$ is an immediate contradiction as $\omega_0^{0,R} = \omega_0^{0,R'} = X \mapsto (\pi \mapsto d)$. Then, we can assume that $j_0 > 0$. If $X_{j_0} \in \text{VAR}_{\exists}(\varphi)$, we have $\omega_0^{j_0,R} = \omega_0^{j_0-1,R}$ and $\omega_0^{j_0,R'} = \omega_0^{j_0-1,R'}$ by definition. Then, $\omega_0^{j_0-1,R} \neq \omega_0^{j_0-1,R'}$ which is in contradiction with the minimality of j_0 . Finally, if $j_0 \in \text{VAR}_{\forall}(\varphi)$, then, by definition of $\omega_0^{j_0,R}$ and $\omega_0^{j_0,R'}$ and by minimality of j_0 , for every $X \neq X_{j_0}$, we have $\omega_0^{j_0,R}(X) = \omega_0^{j_0-1,R}(X) = \omega_0^{j_0-1,R'}(X) = \omega_0^{j_0,R'}(X)$. Furthermore, for every path $\pi \neq p_i$ (remark that $\mathbf{p}_0^{\dot{x},R} = \mathbf{p}_0^{\dot{x},R'} = p_i$), we have $\omega_0^{j_0,R}(X_{j_0})(\pi) = \omega_0^{j_0-1,R}(X_{j_0})(\pi) = \omega_0^{j_0-1,R'}(X_{j_0})(\pi) = \omega_0^{j_0,R'}(X_{j_0})(\pi)$. Then, $\omega_0^{j_0,R}(X_{j_0})(p_i) \neq \omega_0^{j_0,R'}(X_{j_0})(p_i)$. However, by definition, we have $\chi_0^{j_0,R} \upharpoonright_{\text{VAR}^{< x_{j_0}}(\varphi)} =^{\leq 0} \chi_1^{j_0,R} \upharpoonright_{\text{VAR}^{< x_{j_0}}(\varphi)}$ and by Claim 5.4, we have $\chi_1^{j_0,R} \upharpoonright_{\text{VAR}^{< x_{j_0}}(\varphi)} =^{\leq 0} \chi_1$ and then $\omega_0^{j_0,R}(X_{j_0})(p_i) = (R(x_{j_0})(\chi_0^{j_0,R})) [0] = (R(x_{j_0})(\chi_1)) [0]$. Symmetrically, $\omega_0^{j_0,R'}(X_{j_0})(p_i) = (R'(x_{j_0})(\chi_2)) [0]$. By Item 2, we have $(R(x_{j_0})(\chi_1)) [0] = (R'(x_{j_0})(\chi_2)) [0]$, thus, $\omega_0^{j_0,R}(X_{j_0})(p_i) = \omega_0^{j_0,R'}(X_{j_0})(p_i)$, which is a contradiction with the definition of j_0 . Hence $(\Delta(R)(x_i)(\chi_1)) [0] = (\Delta(R')(x_i)(\chi_1)) [0]$.

($n + 1$) Assume that the property holds for some natural n . Also assume that Items 1 to 3 hold for $n + 1$. Then, we can deduce that $\mathbf{p}_n^{\dot{x},R} = \mathbf{p}_n^{\dot{x},R'}$ and apply the same reasoning as the base case.

Without loss of generality, we can assume that k is such that $\Delta(R)(x_i)(\chi_1) =^{\leq n-1} \Delta(R')(x_i)(\chi_2)$. Then, we have $(\Delta(R)(x_i)(\chi_1)) [n] \neq (\Delta(R')(x_i)(\chi_2)) [n]$, which amounts to $\theta(\omega_n^{i,R})(X_i)(\mathbf{p}_n^{\dot{x},R}) \neq \theta(\omega_n^{i,R'})(X_i)(\mathbf{p}_n^{\dot{x},R'})$. However, by assumption, we have that $\omega_n^{i,R} = \omega_n^{i,R'}$ and that $\mathbf{p}_n^{\dot{x},R} = \mathbf{p}_n^{\dot{x},R'}$. Hence the contradiction. \square

Claim 5.6. For every quantifier-free conjunctive formula $\phi = \bigwedge_{b \in B} b\psi_b$, if for every $\mathbf{m} \in \text{Strat}^{V_\forall}$, we have $\mathbf{G}, \theta(\mathbf{m}) \models_{SL} \phi$, then for every $R \in \text{Maps}_{\text{VAR}_\forall(\varphi) \cup x_\odot}$, we have $\text{GTS}_{SL}(\mathbf{G}, \Phi), \text{EXT}(\{\{\emptyset\}\}, \Delta(R)) \models^\alpha b_{id}(\diamond p_\exists) \vee ((\Box \neg p_\forall) \wedge \bigwedge_{b \in B} ((\Box q_b) \rightarrow \psi_b))$.

Proof. Let $R \in \text{Maps}_{\text{VAR}_\forall(\varphi) \cup x_\odot}$. We define $\mathbf{m} \in \text{Strat}^{V_\forall}$ as follow, where d is a fixed action and $x_i \in \text{VAR}_\forall(\varphi)$:

$$\mathbf{m}(x_i)(\pi) = \begin{cases} a_n^i & \text{if } \pi = \mathbf{p}_n^{\dot{\chi}, R} \\ d & \text{otherwise} \end{cases}$$

Let $\chi \in \text{EXT}(\{\{\emptyset\}\}, \Delta(R))$. Furthermore, let $\dot{\pi} = \text{path}_{\text{GTS}_{SL}(\mathbf{G}, \Phi)}(\chi, b_{id})$ and $\pi_b = \text{path}_{\text{GTS}_{SL}(\mathbf{G}, \Phi)}(\theta(\mathbf{m}), b)$. We prove by induction that for every $n \in \mathbb{N}$, we have $\dot{\pi}[n] = (p_n, B_n)$ with and $p_n = \pi_{\chi(x_\odot)[n]}[n]$ and $B_n = \{b \in B_\Phi \mid b \text{ and } \chi(x_\odot)[n] \text{ agree up to } n\}$.

($n = 0$) The initial position in $\text{GTS}_{SL}(\mathbf{G}, \Phi)$ is (p_i, B_Φ) . We can see that, by definition, every bindings agree up to 0.

($n + 1$) Assume that for some $n \in \mathbb{N}$, we have $\dot{\pi}[n] = (p_n, B_n)$ with and $p_n = \pi_{\chi(x_\odot)[n]}[n]$ and $B_n = \{b \in B_\Phi \mid b \text{ and } \chi(x_\odot)[n] \text{ agree up to } n\}$. By definition, $\dot{\pi}[n + 1] = \hat{\delta}(\dot{\pi}[n], x \mapsto \chi(x)[n + 1]) = (p_{n+1}, B_{n+1})$ with the following.

- $p_{n+1} = \delta(p_n, \{a \in \mathbf{Ag} \mapsto \chi(\chi(x_\odot)[n + 1](a))[n + 1]\})$. However, given $a \in \mathbf{Ag}$, let $x_i = \chi(x_\odot)[n + 1](a)$. We have $\chi(\chi(x_\odot)[n + 1](a))[n + 1] = a_{n+1}^i = \theta(\omega_{n+1}^{i,R})(X_i)(\pi[: n + 1])$. By definition of θ , we have $\theta(\omega_{n+1}^{i,R})(X_i)(\pi[: n + 1]) = \theta(\mathbf{m})(X_i)(\pi[: n + 1])$. So we have $p_{n+1} = \delta(p_n, \theta(\mathbf{m})(\pi[: n + 1])) = \pi[: n + 1]$.
- $B_{n+1} = \{b \in B_n \mid \delta(p_n, \{a \in \mathbf{Ag} \mapsto \chi(b(a))[n + 1]\}) = p_{n+1}\}$. It is straight forward to see that $\chi(x_\odot)[n + 1] \in B_{n+1}$. From there, by construction, $B_{n+1} = \{b \in B_\Phi \mid b \text{ and } \chi(x_\odot)[n + 1] \text{ agree up to } n + 1\}$.

Then, because of the definition of the label of $\text{GTS}_{SL}(\mathbf{G}, \Phi)$, we have easily that for every goal $b\psi_b$ such that $\dot{\pi} \models_{LTL} \Box q_b$, we have $\hat{\lambda}(\dot{\pi}) = \pi_b$. By hypothesis on θ , we have that $\pi_b \models \psi_b$. Hence, the result. \square

Which concludes the proof. \square

Given $\mathbf{m} \in \text{Strat}^V$, for $V \subseteq \text{Var}$, and a binding $b \in \text{Binds}$, we define

$$\text{OUT}_B(\mathbf{m}) = \{\pi \in \text{Paths} \mid \exists \mathbf{m} \in \text{Strat}^{\text{img}(b) \setminus V} \text{ s.t. } \forall b \in B, \text{ we have } \pi = \text{path}(\mathbf{m} \uplus \mathbf{m}, b)[: |\pi| - 1]\}.$$

When $B = \{b\}$, we simply write $\text{OUT}_b(\pi)$.

$\text{Strat}_B(\pi)^{\text{VAR}_\forall(\varphi)} \subseteq \text{Strat}^{\text{VAR}_\forall(\varphi)}$ is the set of strategy map \mathbf{m} for $\text{VAR}_\forall(\varphi)$ such that $\pi \in \text{OUT}_B(\mathbf{m})$.

Given a quantifier prefix \wp , a set of bindings B and a timeline map θ for \wp , a (finite) sequence of actions $u = u_1 \cdots u_k \in (\mathbf{Ac}^{\text{VAR}(\wp)})^*$ is coherent with θ for B when for every $b \in B$, we have $\text{PLAY}_b(u) \in \text{OUT}_b(\theta)$.

Lemma 5.6. *For every disjunctive formula in prenex form $\phi = \wp \bigvee_{b \in B} b\psi_b$, given θ , a timeline map for \wp such that for every $\mathbf{m} \in \text{Strat}^{\text{V}\wp}$ there is $b \in B$ such that $\text{PLAY}_G(\theta(\mathbf{m}), b) \models_{\text{LTL}} \psi_b$, there is a function $f_{\text{bind}} : (\mathbf{Ac}^{\text{VAR}(\wp)})^* \rightarrow 2^{\text{BND}(\phi)}$ such that for every $u = u_1 \cdots u_k \in (\mathbf{Ac}^{\text{VAR}(\wp)})^*$ that is coherent with θ for $\text{BND}(\phi)$, and every action profile $\vec{d} \in \mathbf{Ac}^{\text{VAR}(\wp)}$, we have*

1. $f_{\text{bind}}(u\vec{d}) \subseteq f_{\text{bind}}(u)$.
2. for every $b, b' \in f_{\text{bind}}(u)$ we have $\text{path}(u, b) = \text{path}(u, b')$. We denote by $\pi_{f_{\text{bind}}}(u)$ such path.
3. For every $\mathbf{m} \in \text{Strat}_{f_{\text{bind}}(u)}(\pi_{f_{\text{bind}}}(u))^{\text{VAR}\wp}$, there is $b_i \in f_{\text{bind}}(u)$ such that $\theta(\mathbf{m}) \models_{\text{SL}} b_i\psi_i$.

Proof. We define f_{bind} by induction on sequence of action profiles u and verify that f_{bind} satisfies the properties.

$$(\varepsilon) \quad f_{\text{bind}}(\varepsilon) = \text{BND}(\phi).$$

2) For every binding $b \in \text{BND}(\phi)$, we have $\text{path}(u, b) = p_i$.

3) By assumption, for every $\mathbf{m} \in \text{Strat}^{\text{V}\wp}$ there is $i \in \llbracket 0, \dots, k \rrbracket$ such that $\text{PLAY}_G(\theta(\mathbf{m}), b) \models_{\text{LTL}} \psi_i$.

$(u\vec{d})$ Suppose the following.

$$\text{for all } \mathbf{m} \in \text{Strat}_{f_{\text{bind}}(u)}(\pi_{f_{\text{bind}}}(u))^{\text{VAR}\wp} \text{ there is } b_i \in f_{\text{bind}}(u) \text{ such that } \theta(\mathbf{m}) \models_{\text{SL}} b_i\psi_i \quad (5.3)$$

Let $p = \pi_{f_{\text{bind}}}(u)[|u| - 1]$. Consider the equivalence relation \sim on $f_{\text{bind}}(u)$ defined by $b \sim b'$ if $\delta(p, \{a \mapsto \vec{d}(b(a))\}) = \delta(p, \{a \mapsto \vec{d}(b'(a))\})$. We denote by $[b]$ the equivalence class of b .

Claim 5.7. *There is $b_{i_0} \in f_{\text{bind}}(u)$ such that for every $\mathbf{m} \in \text{Strat}_{[b_{i_0}]}(\text{path}(u\vec{d}, b_{i_0}))^{\text{VAR}\wp}$, there is $b_i \in [b_{i_0}]$ such that $\text{PLAY}_G(\theta(\mathbf{m}), b) \models_{\text{LTL}} \psi_i$*

Proof. We do a proof by contradiction. Suppose that

$$\begin{aligned} &\text{for every binding } b_{i_0} \in f_{\text{bind}}(u) \text{ there is } \mathbf{m}_{i_0} \in \text{Strat}_{[b_{i_0}]}(\text{path}(u\vec{d}, b_{i_0}))^{\text{VAR}\wp} \\ &\text{such that for every } b_i \in [b_{i_0}], \text{ we have } \text{PLAY}_G(\theta(\mathbf{m}_{i_0}), b_i) \not\models_{\text{LTL}} \psi_i \end{aligned} \quad (5.4)$$

Fix some i such that $b_i \in f_{\text{bind}}(u)$. Then, consider $\mathbf{m} \in \text{Strat}_{f_{\text{bind}}(u)}(\pi_{f_{\text{bind}}(u)})^{\text{VAR}_{\forall}(\wp)}$ defined by the following where $X \in \text{VAR}_{\forall}(\wp)$.

- when $\pi \in \text{Pref}(\pi_{f_{\text{bind}}(u)})$ or $\pi_{f_{\text{bind}}(u)} \notin \text{Pref}(\pi)$, we set $\mathbf{m}(X)(\pi) = \mathbf{m}_i(X)(\pi)$.
- when $\pi_{f_{\text{bind}}(u)} = \pi$, then we set $\mathbf{m}(X)(\pi) = \vec{d}(X)$.
- when $\pi_{f_{\text{bind}}(u)} \in \text{Pref}(\pi)$ and $\pi_{f_{\text{bind}}(u)} \neq \pi$, we have $\pi = \pi_{f_{\text{bind}}(u)} \cdot q \cdot \pi'$. Let b_{i_0} such that $\delta(p, \{a \mapsto \vec{d}(b_{i_0}(a))\}) = q$. Then, we set $\mathbf{m}(X)(\pi) = \mathbf{m}_{i_0}(X)(\pi)$.

For every binding $b_i \in f_{\text{bind}}(u)$, we have $\text{PLAY}_{\mathbf{G}}(\theta(\mathbf{m}), b_i) = \text{PLAY}_{\mathbf{G}}(\theta(\mathbf{m}_i), b_i)$. So, by Equation (5.4) $\text{PLAY}_{\mathbf{G}}(\theta(\mathbf{m}), b_i) \not\models_{\text{LTL}} \psi_i$, which is in contradiction with Equation (5.3). \square

Let b_{i_0} given by Claim 5.7. We define $f_{\text{bind}}(u\vec{d}) = [b_{i_0}]$.

1. As $[b_{i_0}] \subseteq f_{\text{bind}}(u)$, we have $f_{\text{bind}}(u\vec{d}) \subseteq f_{\text{bind}}(u)$
2. By definition of \sim , we have $\text{path}(u\vec{d}, b) = \text{path}(u\vec{d}, b')$ for every $b, b' \in [b_{i_0}]$
3. Claim 5.7 proves immediately this point.

\square

Lemma 5.7. *Given a quantifier prefix \wp , for every timeline map θ for this prefix, there is a Realizable meta map Δ for $\wp\exists x_{\odot}$ such that for every quantifier-free disjunctive formula $\phi = \wp \bigvee_{b \in B} b\psi_b$, if for every $\mathbf{m} \in \text{Strat}^{\text{V}_{\forall}}$, we have $\mathbf{G}, \theta(\mathbf{m}) \models_{\text{SL}} \phi$, then for every $R \in \text{Maps}_{\text{VAR}_{\forall}(\wp)}$, we have $\hat{\mathbf{G}}, \text{EXT}(\{\{\emptyset\}\}, \Delta(R)) \models^{\alpha} b_{id} \bigvee_{b \in B} ((\square b) \wedge \psi_B)$.*

Proof. We define Δ step by step. Given a PL map $R \in \text{Maps}_{\wp\forall x_{\odot}}(\text{VAR}_{\forall}(\wp))$, we fix an action $d \in \text{Ac}$ and we consider an assignment $\dot{\chi} \in \text{Asg}(\text{VAR}(\wp))$, a sequence of paths $(\mathbf{p}_n^{\dot{\chi}, R})_{n \in \mathbb{N}}$ in \mathbf{G} and of bindings $(b^n)_{n \in \mathbb{N}} \in \text{BND}(\phi)^{\mathbb{N}}$ and, for every variable index $i \in \llbracket 0, \dots, 1, m \rrbracket$, sequences of strategic maps $(\omega_n^{i, R})_{n \in \mathbb{N}} \in (\text{Strat}^{\text{V}_{\forall}})^{\mathbb{N}}$, of assignments $(\chi_n^{i, R})_{n \in \mathbb{N}} \in \text{Asg}(\text{VAR}(\wp))$, of actions $(a_n^i)_{n \in \mathbb{N}} \in (\text{Ac}^{\text{VAR}^{< x_i(\wp)}})^{\mathbb{N}}$ and of plans $(\rho_n^y)_{n \in \mathbb{N}} \in (\text{Plans}^{\text{VAR}^{< x_i(\wp)}})^{\mathbb{N}}$ defined co-inductively as follows, where π is a path in \mathbf{G} and $X \in \text{VAR}(\wp)$. All definitions are the same as for the proof of Lemma 5.5 but for b^n which is obtain thanks to Lemma 5.6.

- $b^n \in f_{\text{bind}}(\chi_n^{m, R}[: n - 1])$

We then define for $i \in I_{\exists}$, we set $\rho_i^{\dot{\chi}}[n] = a_n^i$ and then, we define $\Delta(\mathcal{R})(x_i)(\dot{\chi}) = \rho_i^{\dot{\chi}}$. Furthermore, we define $\Delta(\mathcal{R})(x_i)(\dot{\chi})[n] = b^n$.

The next claim can be proven in a similar fashion as for Lemma 5.5.

Claim 5.8. Δ is a Realizable meta map.

The next claim follows from Lemma 5.6.

Claim 5.9. *For every quantifier-free conjunctive formula $\phi = \wp \bigvee_{b \in B} b \psi_b$, if for every $\mathbf{m} \in \text{Strat}^{V_\forall}$, we have $\mathbf{G}, \theta(\mathbf{m}) \models_{SL} \phi$, then for every $R \in \text{Maps}_{\text{VAR}_\forall(\wp)}$, we have $\hat{\mathbf{G}}, \text{EXT}(\{\{\emptyset\}\}, \Delta(R)) \models^\alpha \bigvee_{i \in [0, \dots, k]} ((\Box b_i) \wedge \psi_i)$.*

□

Theorem 5.3. $\mathbf{G} \models_{SL} \Phi$ iff $\text{GTS}_{SL}(\mathbf{G}, \Phi) \models \text{GTS}_{SL}(\Phi)$, for all $SL[C/DG]$ sentences Φ .

The proof of Theorem 5.3 is very similar to the one for Theorem 5.4.

5.B Missing Details of Section 5.4

Given a CGS \mathbf{G} and a $PL[C/DG]$ sentence $\phi = \wp T \phi$, with binding set $B_\wp \stackrel{\text{def}}{=} \text{BND}(\wp)$, an action assignment $\vec{d} \in \text{Ac}^{\text{VAR}(\phi) \cup \{x_\otimes\}}$ is *V-incoherent* at an extended position $(p, B) \in \text{Pos} \times 2^{B_\wp}$ w.r.t ϕ , where $V \subset \text{Var}$, if there are two variables $x_1, x_2 \in V$ and two bindings $b_1, b_2 \in B$, with $x_1 \in \text{VAR}(b_1)$ and $x_2 \in \text{VAR}(b_2)$, such that $\vec{d}(x_1) \neq \vec{d}(x_2)$. We say that \vec{d} is *\exists -incoherent* (resp., *\forall -incoherent*) at (p, B) w.r.t ϕ if either

1. \vec{d} is \exists -ill-typed (resp., \forall -ill-typed) or
2. \vec{d} is well-typed and the leftmost set of variables V in T , such that \vec{d} is V -incoherent at \hat{p} w.r.t ϕ , occurs in a tying operator of type $\langle V \rangle$ (resp., $[V]$).

If \vec{d} is neither \exists -incoherent nor \forall -incoherent at (p, B) w.r.t ϕ , we say that \vec{d} is *coherent* at (p, B) w.r.t ϕ .

Construction 5.3. *Given a CGS $\mathbf{G} = \langle \text{Ag}, \text{Ac}, \text{Pos}, p_i, \delta, \lambda \rangle$ and a $PL[C/DG]$ sentence ϕ , with binding set $B_\wp \stackrel{\text{def}}{=} \text{BND}(\phi)$, let $\text{GTS}_{C/DG}(\mathbf{G}, \phi)$ be the CGS $\langle \hat{P}, \hat{p}_{\text{init}}, \hat{\text{Ag}}, \hat{\text{Ac}}, \hat{\delta}, \hat{\lambda} \rangle$ obtained as follows:*

1. *agents are the variables quantified in ϕ plus a distinguished agent for the Boolean connective, i.e., $\hat{\text{Ag}} \stackrel{\text{def}}{=} \text{VAR}(\phi) \cup \{x_\otimes\}$;*
2. *actions are extended with bindings, i.e., $\hat{\text{Ac}} \stackrel{\text{def}}{=} \text{Ac} \cup B_\wp$;*
3. *positions are pairs composed of original positions from \mathbf{G} and subsets of bindings of B_\wp , plus three distinguished sink positions $\hat{p}_\exists, \hat{p}_\forall$, and \hat{p}_\otimes , i.e., $\hat{P} \stackrel{\text{def}}{=} \{\hat{p}_\exists, \hat{p}_\forall, \hat{p}_\otimes\} \cup \text{Pos} \times 2^{B_\wp}$;*
4. *the initial position is the pair containing p_i and the full set B_\wp , i.e., $\hat{p}_{\text{init}} \stackrel{\text{def}}{=} (p_i, B_\wp)$;*

5. every position, but the distinguished ones, are labelled with the original labelling plus a set of fresh atomic propositions, one per active binding, i.e., $\hat{\lambda}(\hat{p}_\exists) \stackrel{\text{def}}{=} \{p_\exists\}$, $\hat{\lambda}(\hat{p}_\forall) \stackrel{\text{def}}{=} \{p_\forall\}$, $\hat{\lambda}(\hat{p}_\otimes) \stackrel{\text{def}}{=} \emptyset$, and $\hat{\lambda}((p, B)) \stackrel{\text{def}}{=} \lambda(p) \cup \{q_b \in AP \mid b \in B\}$;
6. the transition function $\hat{\delta}$ maps every position $(p, B) \in \hat{P}$ and action profile $\vec{d} \in \text{Ac}^{\text{VAR}(\phi) \cup \{x_\otimes\}}$ coherent at (p, B) w.r.t φ to position $(u, \mathcal{B}) \in \hat{P}$, where
- (a) u is the successor of p in \mathbf{G} according to the action profile $\vec{d} \circ \flat$ mapping each agent $a \in \text{Ag}$ to the action stipulated by \vec{d} for the variable $\flat(a)$, where $\flat = \vec{d}(x_\otimes)$ is the binding chosen by agent x_\otimes , and
- (b) $\mathcal{B} \subseteq B$ is the set of bindings indistinguishable from \flat at p w.r.t \vec{d} :

$$\hat{\delta}(\hat{p}, \vec{d}) \stackrel{\text{def}}{=} \begin{cases} \hat{p}_Q, & \text{if } \hat{p} = \hat{p}_Q \text{ or, } \hat{p} \neq \hat{p}_\otimes \text{ and } \vec{d} \text{ is } \overline{Q}\text{-incoherent at } \hat{p} \text{ w.r.t } \varphi; \\ \hat{p}_\otimes, & \text{if } \hat{p} = \hat{p}_\otimes \text{ or } \vec{d} \text{ is } \otimes\text{-ill-typed}; \\ (u, \mathcal{B}), & \text{otherwise, where } u = \delta(p, \vec{d} \circ \flat) \text{ and } \mathcal{B} = \{b' \in B \mid b' \equiv_{\vec{d}}^p \flat\}, \\ & \text{with } (p, B) = \hat{p} \text{ and } \flat = \vec{d}(x_\otimes). \end{cases}$$

with $Q \in \{\exists, \forall\}$

And regarding the formula, we set the following.

$$\text{GTS}_{\text{C/DG}}(\wp T \bigwedge_{b \in B_\varphi} b \psi_b) \stackrel{\text{def}}{=} \wp \forall x_\otimes b_{\text{id}} \left((\diamond p_\exists) \vee \left((\square \neg p_\forall) \wedge \bigwedge_{b \in B_\varphi} ((\square q_b) \rightarrow \psi_b) \right) \right)$$

$$\text{GTS}_{\text{C/DG}}(\wp T \bigvee_{b \in B_\varphi} b \psi_b) \stackrel{\text{def}}{=} \wp \exists x_\otimes b_{\text{id}} \left((\square \neg p_\forall) \wedge \left((\diamond p_\exists) \vee \bigvee_{b \in B_\varphi} ((\square q_b) \wedge \psi_b) \right) \right)$$

DEPENDENCY MATRICES

In the previous chapter, we have developed a framework that is suited for strategic reasoning that can generalize to many kind of dependencies. In this chapter, we propose to study the specific temporal dependencies that are delays, commitments and lookaheads. *Games with delayed control* [CFLMZ21] or *delay games* [KZ15; KZ17; TKH12] are two-player games where one player, that we will call the proponent, has to play with a given *delay* δ , i.e. has to choose its actions at time t only depending to the actions of their opponent up to $t - \delta$. In other word, the proponent has to *commit* δ actions. Games with delayed control and delay games differ in the information available for the other player: in a game with delayed control, the opponent cannot see the committed actions of the proponent while in a delay game, they can, granting them a *lookahead* on the proponent. Because of the different treatment of the opponent’s view on the game, games under delayed control are imperfect information games, and thus undetermined, when delay games are perfect information games and so, determined. See [FWZ23] for more in depth analysis of the differences between the two formalisms.

In this chapter, we propose a unifying theoretical framework to specify delay dependencies. To this aim, we define the notion of a *dependency matrix* D . The entry $D[a, b]$ is a generalized integer (i.e. in $\mathbb{Z} \cup \{-\infty, +\infty\}$) so that

“Player a ’s decision at time-step $t_{current}$ depends on
Player b ’s decisions up to time-step $t_{current} + D[a, b]$ ”.

The semantics of a dependency matrix relies on an involved machinery based on an imperfect-information multi-player game, called the *dependency game*. The positions of the game, called *configurations*, are all the possible partial labelings of the timeline by the players – and are thus in infinite number. The dynamics of the dependency game is involved because the dependencies may desynchronize players in their choices for labeling a time point.

Moreover, some matrices encode circular dependencies between players, leading to deadlocked situations where none of the players can progress anymore in the dependency

game, preventing them from completing their labeling of the timeline. Upon the study of this phenomenon, we introduce the class of *progressing* matrices, that guarantee that any play in the dependency game provides a full labeling of the timeline for each propositional variable. We establish an effective property that characterizes these matrices. Thus, we can ensure that plays are infinite and then can be qualified as winning or losing according to some linear-time formula, here an LTL formula.

We then study the problem called EWS (Existence of Winning Strategies) of deciding, given a dependency matrix, a coalition (subset of players) and an LTL formula, the existence of a joint strategy for the coalition such that any play brought about by this strategy satisfies the LTL formula. Importantly, the imperfect-information feature of the dependency game addresses two issues: first, this game is not determined in general, and second, winning strategies for the coalition need being *uniform*, a non-trivial notion in our rich setting since players may be desynchronized.

Although we prove that EWS is unsurprisingly undecidable, we exhibit the subclass of so-called *perfect information* dependency matrices for which EWS turns to a decidable problem. We first consider the perfect information property for matrices whose values range over \mathbb{Z} and show how EWS can be reduced to solving a two-player perfect-information parity game, yielding a 2-EXPTIME-*complete* complexity. We then extend the perfect-information property to arbitrary matrices and provide a decision procedure for EWS that generalizes the one for QPTL [SVW87], thus a non-elementary complexity.

To our knowledge, our proposal offers the first framework amenable for merging many, and yet remote, game settings such as concurrent or turn-based games [AHK02] , (two-player) delay games [KZ15; KZ17; TKH12] and games with delayed control [CFLMZ21], logic QPTL [SVW87], and Church Synthesis Problem (see the survey [Fin16]) – it can be shown that our framework also subsumes DQBF (Dependency Quantified Boolean Formulas) [PRA01].

The chapter is organized as follows. In Section 6.1, we define dependency matrices, and show that they embed several settings of games, we also present the formal machinery to define the dependency arena specified by a dependency matrix and address the problem EWS of the existence of winning strategies and show its undecidability in the general case. Then, in section 6.2, we design the decision procedure for EWS when restricted to a perfect-information matrix input.

6.1 A Multiplayer Delay Game

In this section, we propose the generic notion of dependency matrix and show that it subsumes several classic settings in games. We denote players by lowercase letters such as a, b, c , etc.

6.1.1 Definition

A dependency matrix specifies the mutual dependencies between players' decisions in a game where each player owns an atomic proposition and aims at filling the whole timeline with a valuation for it at each time point.

Definition 6.1. A dependency matrix (or simply a matrix) over a finite set \mathcal{P} of at least two players is a matrix $D = (D[a, b])_{a \neq b \in \mathcal{P}}$ whose values range over $\mathbb{Z} \cup \{-\infty, +\infty\}$.

In a dependency matrix $(D[a, b])_{a \neq b \in \mathcal{P}}$ over \mathcal{P} , the value $D[a, b]$ describes how Player a 's decisions depends on Player b 's: Player a 's decision for choosing the valuation at time point t depends on the ones chosen by Player b at all time points in the interval $[0, t + D[a, b]]$. As such, whenever $D[a, b] < 0$, Player a 's decision at time point t is independent of the decisions made by Player b up to some time point before t . In particular, if $D[a, b] = -\infty$, Player a 's decisions is independent of any of Player b 's.

On the contrary, when $D[a, b] \geq 0$, Player a 's decision at t depends on some Player b 's decisions up to some time point after t , so that Player a is not able to make their decision without this required information. In particular, if $D[a, b] = +\infty$, Player a 's decisions do depend on the decisions of Player b 's over all the timeline.

Because it is natural to consider that a player is aware of their own decisions so far, values on the diagonal $D[a, a]$ are irrelevant and are left undefined. As such, the matrix line $D[a, \cdot]$ specifies the dependencies of Player a with respect to all other players, and their ability to make a decision is constrained by all these dependencies. Unsurprisingly, some matrices may yield blocking situations for some players, an issue that we address later.

Beforehand, we illustrate how several settings in games can be captured with matrices.

Example 6.1 (Concurrent Game).

In a standard concurrent game (as in logics ATL , ATL^*), players have to concurrently choose a move. Thus the move of one player can only depend on the strict past of the history of moves. The corresponding matrix for 3 players is D_1 , where strict past is reflected by the value -1 .

$$D_1 = \begin{matrix} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} a \\ b \\ c \end{matrix} & \begin{pmatrix} \cdot & -1 & -1 \\ -1 & \cdot & -1 \\ -1 & -1 & \cdot \end{pmatrix} \end{matrix}$$

Example 6.2 (Round Robin Game).

In a Round Robin game, players play in turn: first Player a , then b , then c . The matrix is D_2 : decisions of Player a depend on the strict past, hence the values -1 in the a -row; decisions of b depend on the non-strict past of a (hence the value 0), and the strict past of c (hence the value -1); decisions of c depend on the non-strict past of the other players.

$$D_2 = \begin{matrix} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} a \\ b \\ c \end{matrix} & \begin{pmatrix} \cdot & -1 & -1 \\ 0 & \cdot & -1 \\ 0 & 0 & \cdot \end{pmatrix} \end{matrix}$$

Example 6.3 (QPTL).

In QPTL, dependencies stem from the order of the quantifiers: in the formula $\exists a \forall b \exists c \psi$ where ψ is an LTL-formula, Player a plays first on the full timeline and independently of the others. Then b only depends on what Player a did. Finally, c depends on what both Players a and b did. All this is reflected by matrix D_3 .

$$D_3 = \begin{matrix} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} a \\ b \\ c \end{matrix} & \begin{pmatrix} \cdot & -\infty & -\infty \\ +\infty & \cdot & -\infty \\ +\infty & +\infty & \cdot \end{pmatrix} \end{matrix}$$

Example 6.4 (Fixed Delay games).

A delay game is a two-player game, say between Player a and Player b . Player a must make a given finite number k of moves beforehand and then, Players a and b play in a turn based manner, hence maintaining the delay between them (see Klein et al. [KZ15]).¹ This setting is represented by the matrix D_5 .

$$D_5 = \begin{matrix} & \begin{matrix} a & b \end{matrix} \\ \begin{matrix} a \\ b \end{matrix} & \begin{pmatrix} \cdot & -k \\ k-1 & \cdot \end{pmatrix} \end{matrix}$$

Example 6.5 (Church Synthesis).

The Church Synthesis problem (see the survey [Fin16]) consists in responding to a stream of inputs by a stream of outputs, so that a given property holds. If Player a and Player b are in charge of the output, Player c and Player d are in charge of the input, the players dependencies are captured by matrix D_4 : output Players a and b only depend on the past values of the input players, and that input Players c and d see all values and have to respond on the spot.

$$D_4 = \begin{matrix} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{pmatrix} \cdot & -1 & -1 & -1 \\ -1 & \cdot & -1 & -1 \\ 0 & 0 & \cdot & -1 \\ 0 & 0 & -1 & \cdot \end{pmatrix} \end{matrix}$$

6.1.2 Strategies for Dependency Matrices

The Arena of a Matrix

For now, we restrict ourselves to only finite delays. We fix D with values in \mathbb{Z} and we describe the *dependency arena* of D . A position in the dependency arena is called a *configuration*, that is a word vector C that reflects the labeling over $\{\top, \perp\}$ chosen so far by each player of \mathcal{P} . The set of configurations is denoted by $\mathcal{C} := (\{\top, \perp\}^*)^{\mathcal{P}}$ and the initial configuration C^0 is the empty vector, namely $\varepsilon^{\mathcal{P}}$.

Example 6.6. Figure 6.1 shows the configuration C in which Player a played the word $C(a) = \top\top\perp\top\perp$, Player b played $C(b) = \perp\perp\top$, and Player c played $C(c) = \perp\perp\top\top$.

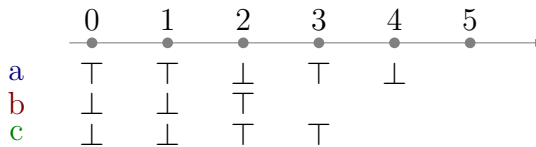
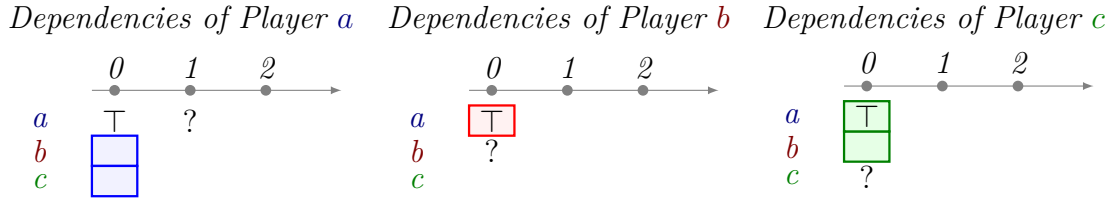


Figure 6.1 – A configuration C where Player a has chosen their labeling up to time point 4, Player b up to time point 2 and Player c up to 3.

1. In their setting, the delay is defined with a delay function that gives at each round the number of moves the input Player has to make. However, Klein et al. mainly studied the fixed delay setting where the function is set to 1 after the first round.

We define the dynamics of the game, namely which player can play/*progress*, in a given configuration and which moves are available to them. Here is an intuitive example of those dynamics with a Round Robin matrix.

Example 6.7. For the matrix D_2 of Example 6.2 and for the first round, only Player a can make a move, and the length of that moves is 1. Then, assuming Player a chooses \top , the dependencies for each player are depicted separately below: in each picture, the squares identify expected information for the considered player to make their decision about the question mark (?).



Observe that, here, neither Player a , nor Player c can make a move as the labeling of Player b at time point 0 is not set. On the contrary, Player b is able to progress.

A *move* of a Player a is a word $u_a \in \{\top, \perp\}^*$ that is to extend their labeling along the timeline. For Player a to progress in configuration C , all their dependencies must be fulfilled. This is formalized as follows: in order to make a move (necessarily starting at the time point $t = |C(a)|$), Player a needs to access the labeling of Player b up to time point $t + D[a, b]$, included. Therefore, the value $\alpha_{a,b}^C := |C(b)| - (D[a, b] + |C(a)|)$ characterizes the length of a move available to Player a with regard to their dependency on Player b only. Thus, the overall *progress value* of Player a , written α_a^C , takes into account all quantities $\alpha_{a,b}^C$ for $b \neq a$ in a conjunctive manner, leading to consider the most restrictive one. Formally:

$$\alpha_a^C \stackrel{\text{def}}{=} \max(0, \min_{b \neq a}(\alpha_{a,b}^C)) \quad (6.1)$$

As such, α_a^C is the maximum number of steps that Player a can perform in configuration C . Note that, if some $\alpha_{a,b}^C$ is non positive, then Player a is stuck in C because Player b has not yet provided the expected information.

Based on the progress value, a *legitimate move* for Player a in a configuration C is a word in $\{\top, \perp\}^{\alpha_a^C}$. Then, a *legitimate joint move* in C is a vector of words $\vec{u} \in (\{\top, \perp\}^*)^P$ such that, for every Player a , \vec{u}_a is a legitimate move for a . We can now define the *move*

function between configurations:

$$\Delta : \mathcal{C} \times (\{\top, \perp\}^*)^{\mathcal{P}} \rightarrow \mathcal{C}$$

$$(C, \vec{u}) \mapsto \begin{cases} C \cdot \vec{u} & \text{when } \vec{u} \text{ is a legitimate joint move in } C \\ \text{undefined} & \text{otherwise} \end{cases}$$

with \vec{u} a legitimate joint move. Remark that all players that can move have to play concurrently and greedily (i.e. their maximum number α_a^C of actions). A configuration C is said *reachable* if there is a finite sequence of legitimate joint moves that leads to C from the initial configuration C^0 .

Proposition 6.1. *Every reachable configuration, has a unique predecessor by Δ .*

Proof. Suppose that there are two reachable configurations C^1 and C^2 and two legitimate joint moves \vec{u}^1 and \vec{u}^2 such that $C^1 \cdot \vec{u}^1 = C^2 \cdot \vec{u}^2$. Then, consider a reachable configuration C' that is a prefix of both C^1 and C^2 . Suppose toward contradiction that there are legitimate joint moves $\vec{u}^{1'} \neq \vec{u}^{2'}$ with $C^{1'} = C' \cdot \vec{u}^{1'}$ prefix of C^1 and $C^{2'} = C' \cdot \vec{u}^{2'}$ prefix of C^2 . Since $\vec{u}^{1'} \neq \vec{u}^{2'}$, there is $t \in \mathbb{N}$ and $a \in \mathcal{P}$ such that $C^{1'}(a)[t] \neq C^{2'}(a)[t]$. By definition, $C^{i'}$ is a prefix of C^i for $i \in \{1, 2\}$. Hence $C^1(a)[t] \neq C^2(a)[t]$ which is in contradiction with $C^1 \cdot \vec{u}^1 = C^2 \cdot \vec{u}^2$. In conclusion, if $C^1 \cdot \vec{u}^1 = C^2 \cdot \vec{u}^2$, then $C^1 = C^2$ and $\vec{u}^1 = \vec{u}^2$. \square

By Proposition 6.1, the dependency arena of reachable configurations is a tree and every reachable configuration contains all moves since the start of the game.

Progressing Matrix

Observe that some dependency arenas have reachable configurations C where for every Player a , we have $\alpha_a^C = 0$. This happens because of cyclic dependencies. Here is an example.

Example 6.8.

Consider the matrix D_6 on the right. Observe that Player b cannot make any move because they needs Player a 's labeling at time point 1, but for Player a to label time point 1, they needs the label of Player b at time point 0. This dead-locked situation is depicted in Figure 6.2. However, observe that Player c can progress independently up to time point 3.

$$D_6 = \begin{matrix} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} a \\ b \\ c \end{matrix} & \begin{pmatrix} \cdot & -1 & -1 \\ +1 & \cdot & -1 \\ -4 & -4 & \cdot \end{pmatrix} \end{matrix}$$

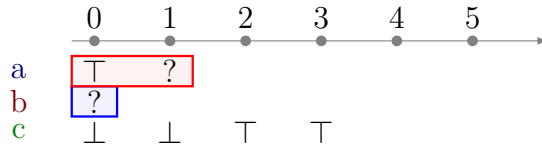
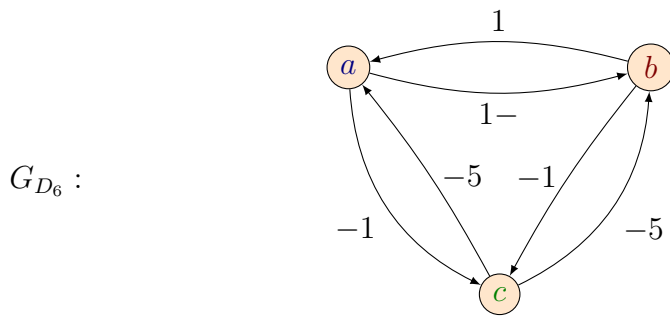


Figure 6.2 – Player *a* wants to know the moves of Player *b* and reciprocally.

Situations where some players eventually get stuck can be characterized by analyzing some graph: for the case of Example 6.8, the graph is depicted below, and interestingly, it contains the cycle (a, b, a) whose weight is 0, a non negative value. We will see in the next section, where the graph is formally defined, that such a cycle provides evidence that Players *a* and *b* eventually get stuck.



We aim at characterizing matrices where no player gets stuck because of cyclic dependencies, so that each play yields an LTL assignment in order to interpret the LTL winning condition. Such matrices are called *progressing* and can be identified by means of their adjacency weighted graph, here called the *dependency graph*.

Definition 6.2. Given a matrix $D = (D[a, b])_{a \neq b \in \mathcal{P}}$, the dependency graph of D is the weighted directed graph $G_D = (V, E, r)$ where:

- $V = \mathcal{P}$ is the set of vertices,
- $E = \{(a, b) | a \neq b\}$ is the set of edges,
- $r(a, b) = D[a, b]$ is the weight of the edge (a, b) .

The following proposition gives a characterization of progressing matrices.

Proposition 6.2. A matrix D is progressing if, and only if, its dependency graph G_D has no non-negative-weighted cycle.

The graph G_{D_6} of Example 6.8 has a non-negative 0-weight cycle (a, b, a) , so, matrix D_6 is not progressing. As a corollary, we have the following:

Theorem 6.1. *Deciding if a matrix is progressing is in PTIME.*

Proof. The size of the dependency graph is linear in the size of the matrix and finding a non-negative cycle is polynomial in the size of the graph. \square

From now on, unless stated otherwise, we only consider progressing matrices, that we keep calling “matrices” for simplicity. On the basis of such matrices, only infinite-horizon *plays* take place that consist in consecutive applications of the move function Δ in the dependency arena (see page 231):

Definition 6.3. *A play in the dependency arena associated to a matrix is an infinite sequence of configurations $(C^n)_{n \in \mathbb{N}}$ where C^0 is the empty configuration, and for every $n \in \mathbb{N}$, C^{n+1} is the successor of C^n .*

The interested reader can explore the dynamics of plays at:

<https://francoisschwarzentruber.github.io/fsttcs2022>

Notice that along a play $(C^n)_{n \in \mathbb{N}}$, C^{n+1} extends C^n , so that to the limit, the play naturally yields a temporal assignment χ of the timeline: for every $t \in \mathbb{N}$, and every Player a , we let $\chi(t)(a) \stackrel{\text{def}}{=} C^n(a)[t]$, for a sufficiently large integer n so that $C^n(a)[t]$ is defined.

Next, we focus on strategies and winning strategies in the meta arena, and we discuss how a strategy complies with a matrix.

Strategies with a Matrix

In this section, we fix a matrix D over \mathcal{P} , a coalition $\Gamma \subseteq \mathcal{P}$ and an LTL formula ψ . Classically, a strategy maps histories to moves. However, since in our setting, a configuration fully characterizes a history (Proposition 6.1), we can equivalently define strategies as mappings from configurations to moves. A *strategy* for Player $a \in \Gamma$ is a function $\sigma : \mathcal{C} \rightarrow \{\top, \perp\}^*$ where $\sigma(C)$ prescribes a legitimate move for Player a .

Furthermore, we define a *joint strategy* for the Γ as a function $\mathfrak{J} : \mathcal{C} \rightarrow (\{\top, \perp\}^*)^\Gamma$ such that $\mathfrak{J}(C)(a)$ is a legitimate move for Player $a \in \Gamma$. A joint strategy \mathfrak{J} provides a strategy \mathfrak{J}_a for each Player $a \in \Gamma$ defined by: $\mathfrak{J}_a(C) \stackrel{\text{def}}{=} \mathfrak{J}(C)(a)$.

Given a joint strategy \mathfrak{J} for a coalition Γ , a play $(C^n)_{n \in \mathbb{N}}$ is an *outcome* of \mathfrak{J} if for any $n \in \mathbb{N}$ and any Player $a \in \Gamma$, we have $C^{n+1}(a) = C^n(a) \cdot \mathfrak{J}_a(C^n)$. We denote

by $\text{OUT}_B((\mathfrak{J}))$ the set of outcomes of \mathfrak{J} . A joint strategy \mathfrak{J} is *winning* ψ whenever all assignments associated to the plays in $\text{OUT}_B((\mathfrak{J}))$ satisfy ψ .

However, in our dependency-based setting, a strategy is relevant only if it is *uniform*, in the sense that they only rely on the information available to the player. We illustrate this important feature in Example 6.9.

Example 6.9. Consider matrix D_7 and configurations C^1 and C^2 below.

$$D_7 = \begin{matrix} & a & b \\ \begin{matrix} a \\ b \end{matrix} & \begin{pmatrix} \cdot & -2 \\ -2 & \cdot \end{pmatrix} \end{matrix} \quad C^1 = \begin{matrix} a & 0 & 1 \\ b & \begin{matrix} \top \\ \perp \end{matrix} & \begin{matrix} \top \\ \perp \end{matrix} \end{matrix} \quad C^2 = \begin{matrix} a & 0 & 1 \\ b & \begin{matrix} \top \\ \perp \end{matrix} & \begin{matrix} \top \\ \top \end{matrix} \end{matrix}$$

When Player a in C^1 comes to label time point 2, and since at time point 2, they can only access Player b 's labeling up to time point 0, they cannot distinguish it from C^2 . However, once they labels time point 2, they are able to access Player b 's label at time point 1, and are allowed to take this information into account before choosing their label at time point 3. Now, according to the matrix D_7 , we have $\alpha_a^{C^1} = \alpha_a^{C^2} = 2$. Although Player a cannot distinguish between C^1 and C^2 , they are allowed to choose different 2-length moves that only differ in their second letters. Indeed, their choice at time point 2 has to be uniform (and therefore the same in both configurations C^1 and C^2). On the contrary, they may play differently for their choice at time point 3. For instance, the overall move in C^1 can be $\top\perp$ while it is $\top\top$ in C^2 .

We formalize the phenomenon described in Example 6.9 with equivalence relations between configurations, in a same vein as in Chapter 4. In the example, Player a has to choose a move u_0u_1 but C^1 and C^2 are indistinguishable for their when it comes to choosing the first letter u_0 . However, they can be distinguished for the choice of the second letter u_1 . Then, we need multiple relations, one for each letter of a move.

Formally, we introduce an equivalence relation between configurations parameterized by a scope k : two configurations C^1 and C^2 are k -indistinguishable for Player a , denoted $C^1 \stackrel{a,k}{\sim}_D C^2$, whenever $|C^1(a)| = |C^2(a)|$, and for every Player $b \neq a$ and every $t \leq |C^1(a)| + D[a, b] + k$, we have:

- either both $C^1(b)[t]$ and $C^2(b)[t]$ are undefined (meaning k is greater than the progress of b in both configurations), or
- $C^1(b)[t] = C^2(b)[t]$.

We resort to relations $\sim_D^{a^k}$ to formalize the notion of uniform strategies in our framework, where the parameter k is meant to range over $[0, \min(\alpha_a^{C^1}, \alpha_a^{C^2})[$.

Definition 6.4. *A strategy σ for Player a is D -uniform whenever for any two configurations C^1 and C^2 , and any natural number $k \leq \min(\alpha_a^{C^1}, \alpha_a^{C^2})$,*

$$C^1 \sim_D^{a^k} C^2 \text{ implies } \sigma(C^1)[:k] = \sigma(C^2)[:k].$$

Observe that $C^1 \sim_D^{a^{k+1}} C^2$ implies $C^1 \sim_D^{a^k} C^2$. We generalize Definition 6.4 to joint strategies in a natural way by requiring the uniform property for every individual strategy of the joint strategy. For the rest of the paper, all strategies are implicitly D -uniform.

Before addressing the central decision problem of the existence of a winning joint strategy, we extend our setting to allow matrices with infinite values.

Infinite-valued Matrix

Recall, by Definition 6.1, that a value $D[a, b] = -\infty$ indicates that Player a 's decision are independent from Player b 's. In particular, if the matrix line $D[a, \cdot]$ is all filled with value $-\infty$, Player a fills the whole timeline in the first round. On the contrary, $D[a, b] = +\infty$ forces Player a to wait until Player b has entirely filled the timeline.

A typical example of a matrix with infinite values is provided by the matrix D_3 of Example 6.3, and is reminiscent of what is expressed in the setting of the logic QPTL: in this example, a play takes place as follows. First, Player a chooses an a -assignment of the timeline. Second, since Player a is done and Player b is independent from Player c , Player b has all the required information for choosing the b -assignment over the timeline. Third and finally, Player c can proceed for the c -assignment, and the play ends.

For the cases that mix finite and infinite value, we consider first Example 6.10

Example 6.10. *Consider matrix D_8 below. In a play, Player b and Player c 's mutual dependencies enforce them to proceed in turn for choosing their respective labeling, while Player a cannot play until the other two have labeled the whole timeline.*

$$D_8 = \begin{matrix} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} a \\ b \\ c \end{matrix} & \begin{pmatrix} \cdot & +\infty & +\infty \\ -\infty & \cdot & -1 \\ -\infty & 0 & \cdot \end{pmatrix} \end{matrix}$$

As observed in Example 6.10, $(\mathbb{Z} \cup \{-\infty, +\infty\})$ -valued matrices may yield to “infinite” configurations (i.e. with possibly components in $\{\top, \perp\}^\omega$ instead of $\{\top, \perp\}^*$). As a result, a play may now be a finite sequence of (possibly infinite) sequences of (possibly infinite) configurations. We can adapt the notion of reachable configuration accordingly – since this is routine, we omit the precise definition here.

From now on, unless stated otherwise, we consider arbitrary matrices. Next, we address the decision problem EWS of the existence of a winning strategy.

6.1.3 Undecidability of Resolution

We consider the following central decision problem EWS of deciding the Existence of a Winning (uniform) Strategy for a coalition of players (EWS for short):

Theorem 6.2. *Let EWS be the following problem:*

Input: *A matrix D , a coalition Γ and an LTL-formula ψ .*

Output: *“Yes” if and only if there is a D -uniform joint strategy for Γ that wins ψ .*

EWS is undecidable.

This result is unsurprising, given the imperfect-information nature of our multi-player game.

Theorem 6.2 is proved by reducing the Tiling problem [Ber66] to EWS. Recall that the Tiling problem takes in input a finite set of square tiles and two binary connectivity relations over the tiles, that specify which pairs of tiles may be adjacent (*resp.* horizontally and vertically). The output is the answer to the question whether there exists a tiling of the plane, that is a mapping from \mathbb{N}^2 to the set of tiles such that any two of adjacent tiles respect the connectivity constraints. In a nutshell, our reduction involves four players τ_1, τ_2 (tilers) and c_1, c_2 (challengers): in a round, Challenger c_1 chooses a place (x, y) in \mathbb{N}^2 and privately communicates it to their tiler companion τ_1 by playing $\top^x \perp \top^y \perp^\omega$. Tiler τ_1 then responds by choosing a tile t independently of the choice of Challenger c_2 and plays $\top^t \perp^\omega$, and symmetrically for Players τ_2 and c_2 . This way, the two tilers play independently. The two different Challengers are used to test the binary relations by choosing adjacent places.

The following matrix encodes this situation.

$$D_{\text{Tiling}} := \begin{matrix} & c_1 & c_2 & \tau_1 & \tau_2 \\ \begin{matrix} c_1 \\ c_2 \\ \tau_1 \\ \tau_2 \end{matrix} & \begin{pmatrix} \cdot & -\infty & -\infty & -\infty \\ -\infty & \cdot & -\infty & -\infty \\ +\infty & -\infty & \cdot & -\infty \\ -\infty & +\infty & -\infty & \cdot \end{pmatrix} \end{matrix}.$$

In the next section, we present a decidable sub-case.

6.2 Resolving Dependency Matrices

A close inspection of the proof of Theorem 6.2 reveals that not being able to circumvent the amount of information hidden to players is a matter. We introduce the subclass of *perfect-information* matrices where every player always has full information about the current configuration before proceeding, yielding a dependency game that is turn-based with perfect information.

6.2.1 Definition and Properties

A dependency game is perfect-information as long as any two reachable configurations are not k -indistinguishable, for every k . Actually, not being 0-indistinguishable is sufficient, since k -indistinguishability are nested (see Section 6.1.2). Furthermore, for the dependency arena to be turn-based, we must guarantee that in each round, only one player can progress. This yields the following definition.

Definition 6.5. *A matrix D is perfect-information if for every reachable configuration C :*

- for every player a and reachable configuration $C' \neq C$, we have $C \not\sim_D^0 C'$ and
- there is exactly one player a such that $\alpha_a^C \geq 1$.

Remark that, by Definition 6.5, every move of a single player, from a reachable configuration that is not the initial one, is necessarily of length 1. Indeed, if a player could make a move of length strictly greater than 1, we could create another reachable configuration that would be 0-indistinguishable from the first one, a contradiction.

Lemma 6.1. *Let D be a perfect-information matrix, then $\alpha_a^C \leq 1$ for any non-initial reachable configuration C and every Player a .*

Proof. By contradiction, suppose that there is a non-initial reachable configuration C^1 with $\alpha_a^{C^1} \geq 2$ for some Player a . Because D is progressing and perfect-information, there is a unique Player b that can progress in $\Delta^{-1}(C^1)$. If $b = a$, we would not have $\alpha_a^{C^1} \geq 2$ because players play greedily. Then, we have $a \neq b$.

We now exhibit a reachable configuration $C^2 \neq C^1$ such that $C^1 \stackrel{a}{\sim}_D C^2$. Let u^1 be the joint move leading to C^1 , that is $\Delta(\Delta^{-1}(C^1), u^1) = C^1$. As exactly one player moves, $u^1(c) = \varepsilon$ for every Player $c \neq b$. We define the joint move u^2 as follows: for every Player $c \neq b$, let $u^2(c) = \varepsilon$ and $u^2(b) = \text{FLIPLAST}(u^1(b))$ where FLIPLAST flips the last letter of the word (mapping \top to \perp , and \perp to \top). For $C^2 = \Delta(\Delta^{-1}(C^1), u^2)$, we have $|C^1(a)| = |C^2(a)|$, and $C^1(c) = C^2(c)$ for Player $c \neq b$. We have $\alpha_{a,b}^{C^i} \geq 2$ for $i \in \{1, 2\}$. Then, by definition, $|C^i(b)| - (D[a, b] + |C^i(a)|) \geq 2$, whence $(D[a, b] + |C^i(a)|) \leq |C^i(b)| - 2$. Now, let $t \leq |C^1| + D[a, b]$. By transitivity, $t \leq |C^i(b)| - 2$. However, $C^1(b)[t] = C^2(b)[t]$ for $t \leq |C^1(b)| - 2$ since only the last letter of $C^1(b)$ differs from $C^2(b)$. Therefore, $C^1 \stackrel{a}{\sim}_D C^2$ which is a contradiction. \square

Corollary 6.1. *Let D be a perfect-information matrix, and C a non-initial reachable configuration, there is Player a with $\alpha_a^C = 1$ and for every other Player b , we have $\alpha_b^C = 0$.*

We can use this result to establish a characterization of perfect-information matrices. Let C be a reachable non-initial configuration and let Player a be the player that can progress in C . By Corollary 6.1, we have $\alpha_a^C = 1$ and $\alpha_b^C = 0$ for every Player $b \neq a$. We first make a claim:

Claim 6.1. $\alpha_{a,b}^C = 1$ and $\alpha_{b,a}^C \leq 0$

Using this claim, we obtain $|C(b)| - (D[a, b] + |C(a)|) = 1$ and $|C(a)| - (D[b, a] + |C(b)|) \leq 0$. Then, $D[a, b] + D[b, a] \geq -1$. Since the matrix is progressing, we have $D[a, b] + D[b, a] \leq -1$ and then $D[a, b] + D[b, a] = -1$. In fact, we show that this necessary condition is a precise characterization of the perfect-information matrices.

Theorem 6.3. *A matrix D is perfect-information if and only if for all players a and b , with $a \neq b$ we have $D[a, b] + D[b, a] = -1$.*

We now show the reciprocal, namely, if D satisfies $D[a, b] + D[b, a] = -1$, then D is perfect-information. The first step is to prove that, the associated dependency arena is turn based.

Lemma 6.2. *For D with $D[a, b] + D[b, a] = -1$ whenever $a \neq b$, there is at most one Player a with $\alpha_a^C \geq 1$ in every configuration C .*

Proof. By contradiction, suppose $\alpha_{a,b}^C \geq 1$ and $\alpha_{b,a}^C \geq 1$. Then, $\alpha_{a,b}^C + \alpha_{b,a}^C \geq 2$. Since $\alpha_{a,b}^C = |C(b)| - (D[a, b] + |C(a)|)$ and $\alpha_{b,a}^C = |C(a)| - (D[b, a] + |C(b)|)$, we obtain $D[a, b] + D[b, a] \leq -2$ which contradicts the assumption on D . \square

It is left to prove that any two different reachable configurations are not $\stackrel{a}{\sim}_D^0$ -equivalent for any a . We here just give an intuition of the proof by contradiction. Suppose that there are two different reachable configurations C^1 and C^2 such that $C^1 \stackrel{a}{\sim}_D^0 C^2$. We can assume without loss of generality that they are immediate successors of the same reachable configuration C . We compare the progress values of Player a with the one of the only player that can progress in C , and prove that the configurations C^1 and C^2 are equal.

6.2.2 A Parity Game to solve EWS

For perfect-information matrices, we establish a reduction from EWS to solving a parity game, thus attaining decidability (Theorem 6.4). Consider a perfect-information matrix D , a coalition Γ and a formula ψ . We define the parity game $G(D, \Gamma, \psi)$ where the coalition Γ has a winning D -uniform strategy if, and only if, Player 0 has a winning strategy in $G(D, \Gamma, \psi)$ against Player 1.

The parity game $G(D, \Gamma, \psi)$ is built up from the deterministic parity automaton \mathcal{A}_ψ for ψ [VW86a; Pit07]. Its plays simulate runs of automaton \mathcal{A}_ψ on the sequence of growing configurations along a play in the dependency arena. Positions in the parity games are pairs composed of states of \mathcal{A}_ψ and *buffers*: a *buffer* β is a word vector $(\beta_a)_{a \in \mathcal{P}}$ with at least one empty component. Formally, the set of buffers is:

$$\{(\beta_a)_{a \in \mathcal{P}} \in (\{\top, \perp\}^*)^{\mathcal{P}} \mid \beta_b = \varepsilon \text{ for some } b \in \mathcal{P}\}.$$

The *buffer of a configuration* is the “pending part” of the configuration, namely its greatest suffix that is a buffer.

Example 6.11. Consider the perfect-information matrix D_9 below where a reachable configuration C and its buffer are depicted.

$$D_9 = \begin{matrix} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} a \\ b \\ c \end{matrix} & \begin{pmatrix} \cdot & 2 & 3 \\ -3 & \cdot & -1 \\ -4 & 0 & \cdot \end{pmatrix} \end{matrix} \quad C = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} a \\ b \\ c \end{matrix} & \begin{matrix} \top & \top & \perp & \top & \perp \\ \top & \perp & \perp & \top & \top \\ \perp & \perp & \top & \top & \perp \end{matrix} \end{matrix}$$

We say that a buffer is *reachable* if it is the buffer of some reachable configuration. We can show that in a reachable configuration, the single player that can progress only depends on the buffer β of this configuration, and we write a_β this player.

We denote by B the set of reachable buffers, by B_\exists the set of buffers β in B where $a_\beta \in \Gamma$, and we let $B \stackrel{\text{def}}{=} B \setminus B_\exists$. Although our matrix is perfect-information, we remark that, by Lemma 6.1, for the particular case of the empty buffer β^0 (of the initial configuration), Player a_{β^0} might be playing a long move. Moreover, it can be shown that the reachable buffers are finitely many¹ and that their number is exponential in the values of the matrix (this is because the longest component of a reachable buffer is given by the biggest absolute value in the matrix).

We now informally describe the parity game with its two players Player 0 and Player 1. As said, a position in the parity arena is pair (q, β) composed of a state q of the automaton and a buffer β . Position (q, β) belongs to Player 0 whenever $\beta \in B_\exists$, otherwise $\beta \in B_\forall$ and it belongs to Player 1.

In a given position (q, β) , only a_β progresses by choosing a move $u \in \{\top, \perp\}$. We consider the word vector obtained by concatenating u to a_β 's component in buffer β , that we write $\beta +_{a_\beta} u$ in the following.

If $\beta +_{a_\beta} u$ is still a buffer we update the position to $(q, \beta +_{a_\beta} u)$. Note that this is always the case for the initial buffer β^0 . Otherwise, the first letter of word vector $\beta +_{a_\beta} u$ is all filled with labels on every component, and thus can be read by automaton \mathcal{A}_ψ . We then update the position to the new current state of \mathcal{A}_ψ and to the buffer obtained by removing the first letter of $\beta +_{a_\beta} u$ (which is a buffer as β is not initial).

Formally, the parity game is the following.

1. Actually the set of reachable configurations is a regular language that can be recognized by a word automaton with buffers as states.

Definition 6.6. Given a perfect-information matrix D , a coalition of players Γ and a deterministic parity automaton $\mathcal{A}_\psi = (Q, q_0, \Sigma, \delta, \text{par})$ with $\Sigma = \{\top, \perp\}^P$, we define the parity game $G(D, \Gamma, \psi) = \langle P_0, P_1, p_0, \rightarrow, \text{par}_G \rangle$ where:

- $P_0 = Q \times B_\exists$ is the set of positions for Player 0,
- $P_1 = Q \times B_\forall$ is the set of positions for Player 1,
- $p_0 = (q_0, \beta^0)$ is the initial position,
- $(q, \beta) \rightarrow (q', \beta')$ when there is a legitimate move u for a_β in the dependency arena such that:
 1. either $\beta +_{a_\beta} u$ is a buffer and $q = q'$ and $\beta' = \beta +_{a_\beta} u$.
 2. or $q' = \delta(q, \beta +_{a_\beta} u[0])$ and $\beta' = \beta +_{a_\beta} u[1 :]$ and a_β is the only player such that $\beta_{a_\beta} = \varepsilon$;
- $\text{par}_G(q, \beta) = \text{par}(q)$, that is the priority of a position (q, β) is the priority of the state q in the automaton \mathcal{A}_ψ .

Note that the number of positions in the parity game is the product of the number of states in the automaton and the number of buffers, and that the game has the same priorities as the automaton.

Proposition 6.3. (For a perfect-information matrix D) $\langle D, \Gamma, \psi \rangle$ is a positive instance of EWS if, and only if, Player 0 has a winning strategy in $G(D, \Gamma, \psi)$.

Proposition 6.3 gives us an upper bound complexity of EWS by the following algorithm.

1. Compute the deterministic parity automaton \mathcal{A}_ψ (accepting the models of ψ);
2. Compute the parity game $G(D, \Gamma, \psi)$;
3. Solve $G(D, \Gamma, \psi)$.

In the following, the *size* of the matrix D is the quantity $|D| = \sum_{a \neq b} |D[a, b]|$. Observe that we can build \mathcal{A}_ψ by using the Vardi-Wolper construction [VW86a] with the Safra-like translation from Büchi to parity acceptance condition [Pit07], so that parity game of Step 2 has $O(2^{2^{|\psi|}} \times 2^{2^{|D|}})$ positions and $O(2^{|\psi|})$ priorities, hence a 2-EXPTIME decision procedure for EWS.

The next subsection, we show that this algorithm is essentially optimal by a reduction of the Church Synthesis problem.

6.2.3 Reduction from the Church Synthesis problem

For the lower bound, we reduce the Church Synthesis for LTL properties [PR89; Fin16]. Our Example 6.5 illustrates the reduction.

Definition 6.7. *Given a coalition Γ , a Church matrix is a matrix D where for any two players $a \neq b$, we have:*

$$D[a, b] = \begin{cases} 0 & \text{if } a \notin \Gamma \text{ and } b \in \Gamma \\ -1 & \text{otherwise} \end{cases}$$

In essence, for Church matrices, players have the same knowledge about the current configuration, allowing them to foresee their allies moves.

Observe that Church matrices may not be perfect-information, since the moves of every player in a team (coalition or opponents) are concurrent. Nonetheless, we can “transform” any Church matrix D into a linear sized perfect-information Round Robin matrix D' (see Example 6.2 and Definition 6.8) such that $\langle D, \Gamma, \psi \rangle$ is a positive instance of EWS if, and only if, $\langle D', \Gamma, \psi \rangle$ is a positive instance of EWS.

Definition 6.8. *A Round Robin matrix is a matrix D such that there exists a total order \preceq over \mathcal{P} , where*

$$D[a, b] = \begin{cases} -1 & \text{if } a \preceq b \\ 0 & \text{otherwise} \end{cases}$$

The total order \preceq describes the order in which players will play (the player that is minimal for \preceq plays first). Remark that $D[a, b] + D[b, a] = -1$ for any two players $a \neq b$ then every Round Robin matrix is perfect-information (by Theorem 6.3). Given a Church matrix, we can choose order \preceq so that all players in the coalition play before their opponents.

By summing up, we polynomially reduce² a Church synthesis problem to a EWS problem for a Church matrix and that is in turn linearly reduced to a EWS problem for a Round Robin matrix. From this latter reductions, we can state the following.

Theorem 6.4. *EWS for perfect-information matrices is 2-EXPTIME-complete in the size of the LTL formula.*

². The size of the Church matrix is quadratic in the number of propositions of the Church synthesis problem

In the next subsection, we extend the class of perfect-information matrices to allow some matrices with infinite values, while keeping the decidability of EWS for the resulting superclass. In particular, QPTL matrices (see Example 6.3 on page 228) falls into this class.

6.2.4 Perfect-Information Matrices with Possibly Infinite Values

The proof of Proposition 6.3 can be extended to QPTL formulas instead of LTL formulas. The following example illustrates a procedure for matrices with infinite values that yields a generalization of the perfect-information property (see Definition 6.9).

Example 6.12. *Consider the following matrices where the latter is perfect-information:*

$$D_8 = \begin{matrix} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} a \\ b \\ c \end{matrix} & \begin{pmatrix} \cdot & +\infty & +\infty \\ -\infty & \cdot & -1 \\ -\infty & 0 & \cdot \end{pmatrix} \end{matrix} \quad \text{and} \quad D'_8 = \begin{matrix} & \begin{matrix} b & c \end{matrix} \\ \begin{matrix} b \\ c \end{matrix} & \begin{pmatrix} \cdot & -1 \\ 0 & \cdot \end{pmatrix} \end{matrix}$$

According to D_8 , Player a depends on the whole labeling of Player b and Player c . Given an LTL formula ψ and say coalition $\{a, b\}$, we can answer the EWS on instance $\langle D_8, \{a, b\}, \psi \rangle$ as follows: we can first answer EWS for $\langle D'_8, \{b\}, \exists a \psi \rangle$ (since D'_8 is perfect-information). If no, then return no for $\langle D_8, \{a, b\}, \psi \rangle$. Otherwise, each outcome of the winning strategy for Player 0 in $\langle D'_8, \{b\}, \exists a \psi \rangle$ reflects a play ρ in $\langle D'_8, \{b\}, \exists a \psi \rangle$. From play ρ , exhibit a unique accepting run in $\mathcal{A}_{\exists a \psi}$. By tracing back this run inside \mathcal{A}_ψ , reconstruct Player a 's response to the play ρ .

The procedure employed in Example 6.12 applies to arbitrary matrices as long as they fulfill the Definition 6.9.

Definition 6.9. *An arbitrary matrix D is perfect-information if for any $a \neq b$:*

1. $D[a, b] \in \mathbb{Z}$ implies $D[a, b] + D[b, a] = -1$;
2. $D[a, b] \in \{-\infty, +\infty\}$ implies $D[a, b] = -D[b, a]$;
3. $D[a, b] = +\infty$ implies $D[a, c] \in \{-\infty, +\infty\}$, for all $c \neq a$.

Observe that the procedure is in fact tower-exponential in the number of players with $+\infty$ dependencies. Moreover, since the validity problem for QPTL [SVW87] reduces to EWS for arbitrary perfect-information matrices, we have the following.

Theorem 6.5. *EWS is non-elementary for arbitrary perfect-information matrices.*

6.3 Conclusion

We presented the expressive framework of dependency matrices that can capture several game settings such as concurrent and turn-based games [AHK02], (two-player) delay games [KZ15; KZ17; TKH12], logic QPTL [SVW87], and Church Synthesis Problem [Fin16].

We proved that the existence of a winning strategy for a coalition to achieve an LTL formula (EWS) is undecidable for arbitrary matrices.

We then exhibited the subclass of perfect-information bounded-value matrices for which the problem EWS is 2-EXPTIME-*complete* in the size of the formula.

Finally, we extended the class of perfect-information matrices with a narrow use of infinite dependencies allowing to re-use known techniques of automata projection for QPTL. For these matrices, EWS becomes non-elementary. Still our complexity analysis of EWS needs being refined regarding the matrix parameter: we do not know yet the lower bound complexity when the LTL formula is fixed.

A first track to continue this work concerns EWS for the whole class of bounded-value matrices. We conjecture it is decidable, since, for a bounded-value matrix, each k -indistinguishable equivalence class of a reachable configurations has a bounded size.

A second track regards our transformation of Church matrices into perfect-information Round Robin ones. We believe that our approach can generalize to a class of bounded-values matrices enlarging the one of Church matrices.

6.A Proofs of Section 6.1

Given a matrix D , we say that a player a is *eventually blocked* if there is a natural $k \in \mathbb{N}$ such that for all reachable configurations C , we have $|C(a)| \leq k$. First we prove a lemma on non-negative cycles. Intuitively, this lemma helps to find a player that can never progress in a non-negative cycle.

Given two vertices c_i and c_j of a non-negative cycle $c = (c_0, \dots, c_{|c|})$ (with $c_0 = c_{|c|}$), we denote by w_i the label $r(c_i, c_{i+1})$ and $W_{i,j}$ the circular sum of the labels between c_i and c_j :

- If $i < j$, then $W_{i,j} = w_i + \dots + w_{j-1}$;
- else, $W_{i,j} = W_{i,|c|} + W_{0,j}$.

Remark that for every index i , $W_{i,i}$ is the sum of all labels of the cycle. So, for a non-negative cycle, $W_{i,i} \geq 0$. Furthermore, we denote by W_i^* the minimal sum $W_i^* = \min_j(W_{i,j})$.

Lemma 6.6. *Given a matrix D with a non-negative cycle $c = (c_0, \dots, c_{|c|})$, for all player c_i in the cycle, for all reachable configuration C , we have the following.*

$$|C(c_i)| \leq \max(0, -W_i^*)$$

Proof. We do the proof by induction on reachable configuration.

(base case) In the initial configuration C^0 , the property is obvious.

(inductive case) Consider a configuration $C' = \Delta(C, \vec{u})$ for some joint move \vec{u} and a reachable configuration C such that $|C(c_i)| \leq \max(0, W_i^*)$ for every $c_i \in c$. Consider $i \in \llbracket 0, \dots, |c| \rrbracket$.

$$\begin{aligned} |C'(c_i)| &= |C(c_i)| + \alpha_{c_i}^C \\ &= |C(c_i)| + \max(0, \min_{a \neq c_i}(\alpha_{c_i, a}^C)) \\ &\leq |C(c_i)| + \max(0, \alpha_{c_i, c_{i+1}}^C) \\ &\leq |C(c_i)| + \max(0, |C(c_{i+1})| - D[c_i, c_{i+1}] - |C(c_i)|) \\ &\leq \max(|C(c_i)|, |C(c_{i+1})| - w_i) \\ &\leq \max(\max(0, -W_i^*), \max(-w_i, -W_{i+1}^* - w_i)) \end{aligned}$$

The last inequality is obtained thanks to the inductive hypothesis. We now prove that $-W_i^* \geq \max(-w_i, -W_{i+1}^* - w_i)$. Since $w_i = W_{i, i+1}$, we have $-W_i^* \geq -w_i$. Let $j_0 \in \llbracket 0, \dots, |c| \rrbracket$ such that $W_{i+1}^* = W_{i+1, j_0}$. We now do a case study on j_0 .

— if $j_0 = i + 1$, then, because c is non-negative, $-W_{i+1}^* - w_i \leq -w_i$.

— if $j_0 \neq i + 1$, then, $-W_{i+1}^* - w_i = -W_{i,j_0} \leq -W_i^*$

Then we have $|C'(c_i)| \leq \max(0, -W_i^*)$.

We have proven the property by induction. □

We now state a lemma to prove the other way around: that non-negative cycles are necessary for a matrix not to be progressing.

Lemma 6.7. *Given a bounded matrix D , if there is a player that is eventually blocked, then every player is eventually blocked.*

Proof. Consider a bounded matrix D where there is a player a and a natural number $k \in \mathbb{Z}$ such that for every reachable configuration C , we have $|C(a)| \leq k$. We prove by induction on reachable configurations that for each Player b , we have $|C(a)| \leq \max(0, k - D[b, a])$.

(base case) In the initial configuration C^0 , the property is obvious.

(inductive case) Consider a configuration $C' = \Delta(C, \vec{u})$ for some joint move \vec{u} and a configuration C such that $|C(a)| \leq k - D[b, a]$ for every b . Then, for every Player b we have the following.

$$\begin{aligned}
 |C'(b)| &= |C(b)| + \alpha_b^C \\
 &= |C(b)| + \max(0, \min_{c \neq b}(\alpha_{b,c}^C)) \\
 &\leq |C(b)| + \max(0, \alpha_{b,a}^C) \\
 &\leq |C(b)| + \max(0, |C(a)| - D[b, a] - |C(b)|) \\
 &\leq \max(|C(b)|, |C(a)| - D[b, a]) \\
 &\leq \max(\max(0, k - D[b, a]), k - D[b, a]) \\
 &\leq \max(0, k - D[b, a])
 \end{aligned}$$

We have proven the property by induction. □

Now, when all players are blocked, we use the next well known result of graph theory to find our non-negative cycle.

Lemma 6.8. *Given a directed graph G with no self loop, if every vertex is the source of an edge, then, there is a cycle in the graph.*

Now, we can give a proof for Proposition 6.2.

Proposition 6.2. *A matrix D is progressing if, and only if, its dependency graph G_D has no non-negative-weighted cycle.*

Proof. Given a matrix D with a non-negative cycle, then, by Lemma 6.6, we immediately have that every player in the cycle is eventually blocked.

In a second time, consider a matrix D that is not progressing. Then, by Lemma 6.7, for every player a , there is an integer k_a such that for every reachable configuration C , we have $|C(a)| \leq k_a$. Consider a configuration C such that every player is blocked. Then, by immediate contradiction, for every player a , there is a player $q_a \neq a$ such that $\alpha_{a,q_a}^C \leq 0$ (otherwise, there would be a player that can progress). The graph $G = \langle V, E \rangle$ with the vertices $V = \mathcal{P}$ are the players of the matrix and the edges are defined as $E = \{(a, q_a) \mid a \in \mathcal{P}\}$. Since, G is a directed graph with no self loop, by Lemma 6.8, there is a cycle $c = (c_0, \dots, c_{|c|})$ in the graph thus, for every $i \in \llbracket 0, \dots, |c| - 1 \rrbracket$, we have $c_{i+1} = q_{c_i}$ and $c_{|c|} = c_0$. We have the following.

$$\begin{aligned} \sum_{i=0}^{|c|-1} \alpha_{c_i, c_{i+1}}^C &= \sum_{i=0}^{|c|-1} |C(c_{i+1})| - D[c_i; c_{i+1}] - |C(c_i)| \\ &= - \sum_{i=0}^{|c|-1} D[c_i; c_{i+1}] \end{aligned}$$

Since $\alpha_{c_i, c_{i+1}}^C \leq 0$ for every $i \in \llbracket 0, \dots, |c| \rrbracket$, we have proven that c is a non-negative cycle. \square

6.B Proofs of Section 6.2

We now address the proof of Theorem 6.3. The first direction states that a perfect-information matrix D satisfies that for every different Players a and b , we have $D[a, b] + D[b, a] = -1$ and is presented in Section 6.2. We just need to prove Claim 6.1. Recall that C is a reachable non-initial configuration and Player a is the player that can progress in C . By Corollary 6.1, we have $\alpha_a^C = 1$ and $\alpha_b^C = 0$ for every player $b \neq a$. We consider a player $b \neq a$.

Claim 6.1. $\alpha_{a,b}^C = 1$ and $\alpha_{b,a}^C \leq 0$

Proof. First we prove that $\alpha_{b,a}^C \leq 0$. Remark that, in C , Player a has two legitimate moves: \top and \perp . Let $C^1 = \Delta(C, (\top)_a)$ and $C^2 = \Delta(C, (\perp)_a)$. Note that C^1 and C^2 are reachable. Toward contradiction, assume that $\alpha_{b,a}^C > 0$. We prove the contradiction $C^1 \stackrel{b}{\sim}_D^0 C^2$. We have $\alpha_{b,a}^C = |C(a)| - (D[b, a] + |C(b)|) > 0$. Then $|C(a)| > D[b, a] + |C(b)|$. Therefore, for

all $t \leq D[b, a] + |C(b)|_0$, we have $t < |C(a)|$. And because $C^1(a)[t] = C(a)[t] = C^2(a)[t]$, we conclude that $C^1 \stackrel{b_0}{\sim}_D C^2$.

Then we prove that $\alpha_{a,b}^C = 1$. By definition, $\alpha_{a,b}^C \geq 1$. Toward contradiction, suppose $\alpha_{a,b}^C > 1$. Let C' be the configuration defined by $C'(c) = C(c)$ for all $c \neq b$ and $C'(b) = \text{FLIPLAST}(C(b))$. We have that C' is reachable and, by the same kind of reasoning than previous point, we have $C \stackrel{a_0}{\sim}_D C'$, which is in contradiction with the assumption on D . \square

Let us prove the other direction, namely that a matrix D is perfect-information if $D[a, b] + D[b, a] = -1$ for every pair of different Players a and b . Lemma 6.2 states that the meta game of such a matrix is turn based. Now, we need to prove that two different reachable configurations are not 0-indistinguishable. To do so, we first show two results on 0-indistinguishable relations. These results allow us to consider the "first time" at which two configurations diverge while being 0-indistinguishable.

Lemma 6.9. *Given a matrix D and two reachable configurations C and C' , with $(C^k)_{k \leq n}$ and $(C'^k)_{k \leq m}$ such that $C^n = C$ and $C'^m = C'$. If $n \geq m$ then, for every Player a we have $|C(a)| \geq |C'(a)|$.*

Proof. The proof is done by induction on $n - m$.

(base case) If $n = m$, we do an induction on n .

(base case) If $n = m = 0$, then $C = C' = C^0$, the property is immediate.

(induction) Suppose the property holds for some $n, m \in \mathbb{N}$ with $n = m$. Consider C and C' reachable. There are $(C^k)_{k \leq n+1}$ and $(C'^k)_{k \leq m+1}$ with $C^{n+1} = C$ and $C'^{m+1} = C'$. By inductive hypothesis, we have $|C^n(a)| = |C'^m(a)|$ for every Player a . Then, $\alpha_a^{C^n} = \alpha_a^{C'^m}$ and so, $|C(a)| = |C'(a)|$.

(induction) Suppose the property holds for some $n, m \in \mathbb{N}$ with $n \geq m$. Consider C and C' reachable: there are $(C^k)_{k \leq n+1}$ and $(C'^k)_{k \leq m}$ with $C^{n+1} = C$ and $C'^m = C'$. By inductive hypothesis, for every Player a $|C^n(a)| \geq |C'^m(a)|$ and because $|C^{n+1}(a)| \geq |C^n(a)|$, we have $|C^{n+1}(a)| \geq |C'^m(a)|$.

We have proved the property by induction. \square

Lemma 6.10. *Given a dependency matrix D , and two reachable configurations C and C' that are non-initial, if $C \stackrel{a_0}{\sim}_D C'$, then, one of the following holds.*

1. $\Delta^{-1}(C) \stackrel{a_0}{\sim}_D C'$

2. $C \stackrel{a_0}{\sim}_D \Delta^{-1}(C')$
3. $\Delta^{-1}(C) \stackrel{a_0}{\sim}_D \Delta^{-1}(C')$

Proof. Consider two reachable configurations C and C' such that $C \stackrel{a_0}{\sim}_D C'$. By definition, $|C(a)| = |C'(a)|$, and for every Player $b \neq a$, every $t \leq |C(a)| + D[a, b]$, we have $C(b)[t] = C'(b)[t]$. Since C and C' are reachable, there are two sequences of successive configurations $(C^k)_{k \leq n}$ and $(C'^k)_{k \leq m}$ such that $C^n = C$ and $C'^m = C'$. By symmetry we can assume $n \geq m$. We do a case study.

If $n = m$, then $\Delta^{-1}(C)$ and $\Delta^{-1}(C')$ are also reachable and their sequences have the same length. By Lemma 6.9 for every player b , we have that $|\Delta^{-1}(C)(b)| = |\Delta^{-1}(C')(b)|$. Immediately, $|\Delta^{-1}(C)(a)| = |\Delta^{-1}(C')(a)|$ and because $C \stackrel{a_0}{\sim}_D C'$, for every $t \leq |\Delta^{-1}(C)(a)| + D[a, b] \leq |C(a)| + D[a, b]$, we have $\Delta^{-1}(C)(b)[t] = C(b)[t] = C'(b)[t] = \Delta^{-1}(C')(b)[t]$. Hence, $\Delta^{-1}(C) \stackrel{a_0}{\sim}_D \Delta^{-1}(C')$

If $n > m$ then $\Delta^{-1}(C)$ is reachable with a sequence of length $n - 1 \geq m$ as $\Delta^{-1}(C) = C^{n-1}$ then, by Lemma 6.9, for every player b , we have $|\Delta^{-1}(C)(b)| \geq |C'^m(b)|$. And, because $C \stackrel{a_0}{\sim}_D C'$, we have $|C(a)| = |C'(a)|$, then $|\Delta^{-1}(C)(a)| = |C'(a)|$. Furthermore, for every $t \leq |\Delta^{-1}(C)(a)| + D[a, b]$, if both $\Delta^{-1}(C)(b)[t]$ and $C'(b)[t]$ are defined, we have $\Delta^{-1}(C)(b)[t] = C(b)[t] = C'(b)[t]$. Finally, we prove that $C(b)[t]$ is defined if and only if $C'(b)[t]$ is defined. By Lemma 6.9, we already have that $|\Delta^{-1}(C)(b)| \geq |C'^m(b)|$, and if $\Delta^{-1}(C)(b)[t]$ is defined, so is $C(b)[t]$, and by hypothesis, so is $C'^m(b)[t]$.

We have proved the property. □

We now prove the theorem.

Theorem 6.3. *A matrix D is perfect-information if and only if for all players a and b , with $a \neq b$ we have $D[a, b] + D[b, a] = -1$.*

Proof. The first direction is presented in Section 6.2. For the other direction, consider D such that $D[a, b] + D[b, a] = -1$. By Lemma 6.2, there is only one player that can progress in any reachable configuration. Is left to prove that for any two reachable configurations C and C' with $C \neq C'$, for every player a , we have $C \not\stackrel{a_0}{\sim}_D C'$. The proof is done by contradiction.

Suppose that there are two different reachable configurations C^1 and C^2 with $C^1 \stackrel{a_0}{\sim}_D C^2$. Then, by Lemma 6.10, we can assume that $\Delta^{-1}(C^1) = \Delta^{-1}(C^2) = C$. Let b the player that can progress in C . Then, for all $c \neq b$, we have $C(c) = C^1(c) = C^2(c)$

and, for all $t < |C(b)|$, we have $C(b)[t] = C^1(b)[t] = C^2(b)[t]$. Because $C^1 \neq C^2$, then we necessarily have the following.

$$C^1(b)[t_0] \neq C^2(b)[t_0] \text{ for some } t_0 \in \{|C(b)| - \alpha_b^C, |C(b)| - 1\} \quad (6.2)$$

Since $\alpha_{b,a}^C \geq \alpha_b^C$, we have $|C(a)| - (D[b, a] + |C(b)|) \geq \alpha_b^C$. As Player a does not progress in C , we obtain $|C^1(a)| - (D[b, a] + |C(b)|) \geq \alpha_b^C$. By hypothesis $D[b, a] = -1 - D[a, b]$, we have $|C^1(a)| + 1 + D[a, b] - |C(b)| \geq \alpha_b^C$ and because $|\Delta^{-1}(C^1)(b)| + \alpha_b^C = |C^1(b)|$, we have:

$$|C^1(a)| + D[a, b] \geq |C^1(b)| - 1 \quad (6.3)$$

Finally, since $C^1 \stackrel{a_0}{\sim}_D C^2$, we have that $C^1(b)[t] = C^2(b)[t]$ for every $t \leq |C^1(a)| + D[a, b]$. In particular, thanks to Equation (6.3), we can take $t = t_0$, and we have $C^1(b)[t_0] = C^2(b)[t_0]$, in contradiction with Equation (6.2). \square

6.C Reduction from Church matrices to Round Robin matrices

We now prove the claim that Church matrices can be reduced to Round Robin matrices.

From a Church matrix D for a coalition Γ , we define a Round Robin matrix $\text{Rob}(D)$ as follows. We take an arbitrary order \preceq on the players such that for every Player $a \in \Gamma$ and every Player $b \notin \Gamma$, it holds $a \preceq b$. $\text{Rob}(D)$ is the Round Robin matrix for this order.

Lemma 6.11. *Given an LTL formula ψ and a Church matrix D , there is a winning joint D -uniform strategy iff there is a winning joint strategy $\text{Rob}(D)$.*

Proof. Suppose that there is a winning joint strategy \mathfrak{J} for Γ that is $\text{Rob}(D)$ -uniform. Only the moves of players of the coalition can make \mathfrak{J} non- D -uniform. But, given the joint strategy for the whole coalition, we cannot reach two configurations that are equal on everything except a labeling of a player of the coalition because our strategies are deterministic. Thus, in practice, \mathfrak{J} is D -uniform.

Conversely, consider two configurations C^1 and C^2 such that $C^1 \stackrel{a}{\sim}_{\text{Rob}(D)} C^2$ for some $a \in \Gamma$. We denote by k the length of $C^1(a)$. Then for every $b \in \mathcal{P}$, if $a \preceq b$, we have $C^1(b)[: k - 1] = C^2(b)[: k - 1]$ and otherwise, $C^1(b)[: k] = C^2(b)[: k]$. Because we chose

the order so that $a \preceq b$ for every $a \in \Gamma$ and $b \notin \Gamma$, we have that $C^1 \stackrel{a}{\sim}_D C^2$. Thus every strategy D -uniform is Rob(D)-uniform. \square

6.D Proof of Theorem 6.5

Theorem 6.5. *EWS is non-elementary for arbitrary perfect-information matrices.*

Proof. Given a dependency matrix D , we decompose the set of Players \mathcal{P} as follows.

$$\begin{aligned} \mathcal{P}_\infty &:= \{a \in \mathcal{P} \mid \text{There is } b \in \mathcal{P} \text{ such that } D[a, b] = +\infty\} \\ \mathcal{P}_Z &:= \{a \in \mathcal{P} \mid \text{For all } b \in \mathcal{P} \text{ it holds } D[a, b] < +\infty\} \end{aligned}$$

We reason by induction on the size of \mathcal{P}_∞ . The base case is $|\mathcal{P}_\infty| = 0$. In this case, we can apply Theorem 6.4.

Suppose that we can decide the EWS problem for matrices with $|\mathcal{P}_\infty| = n$ for some $n \in \mathbb{N}$. We now prove that we can decide the problem for matrices with $|\mathcal{P}_\infty| = n + 1$. Consider a matrix D such that $|\mathcal{P}_\infty| = n + 1$, a coalition Γ and a QPTL formula φ . We define an order \preceq on \mathcal{P}_∞ that is given as follows. $a \preceq b$ iff for all $c \in \mathcal{P}$, if $D[b, c] = +\infty$, then $D[a, c] = +\infty$. Intuitively, $a \preceq b$ means that Player a is to play after Player b . Consider Player a , the smallest player for this order. For all players b different than a we have $D[a, b] = +\infty$. We now construct a new instance of the problem by projecting out Player a .

If $(a \in \Gamma)$ we state $\Gamma' = \Gamma \setminus \{a\}$ and $\varphi' = \exists a. \varphi$. By inductive hypothesis, we can decide whether there is a joint strategy winning for the entry $\langle D', \Gamma', \varphi' \rangle$. Let us prove that $\langle D', \Gamma', \varphi' \rangle$ is a positive instance iff $\langle D, \Gamma, \varphi \rangle$ is a positive instance. If there is a joint strategy \mathfrak{J}' for the coalition winning for the entry $\langle D', \Gamma', \varphi' \rangle$, then, for every play $(C_1^n)_n, \dots, (C_k^n)_n$ in the meta game of D' , the assignment χ which is the limit of that play satisfy $\langle D', \Gamma', \varphi' \rangle$. Therefore, there is an infinite word u_χ such that $\chi[a \mapsto u_\chi]$ satisfies φ' . Then we define the joint strategy \mathfrak{J} as follows. For every Player $b \neq a$, we set $\mathfrak{J}_b = \mathfrak{J}'_b$. For Player a , consider a configuration C such that $\alpha_a^C > 0$. Because of the dependencies of Player a , it holds that $|C(b)| = +\infty$ for every Player $b \neq a$. Let χ be the temporal assignment on $\mathcal{P} \setminus \{a\}$ defined by C . We set $\mathfrak{J}_a(C) = u_\chi$. This joint strategy is winning. The converse follows the same idea: if there is \mathfrak{J} winning for the entry (D, Γ, φ) then, the joint strategy $\mathfrak{J}' = \mathfrak{J}|_{\mathcal{P} \setminus \{a\}}$ is winning for the entry (D', Γ', φ') .

If $(a \notin \Gamma)$ we state $\varphi' = \forall a. \varphi$. We then decide the instance $\langle D', \Gamma, \varphi' \rangle$. Joint strategies for the coalition translate naturally between the two instances. If \mathfrak{J}' is winning for $\langle D', \Gamma, \varphi' \rangle$, then every play in the outcome of \mathfrak{J}' satisfies $\forall a \varphi$. Then, by defining $\mathfrak{J}(C) = \mathfrak{J}'(C \upharpoonright_{\mathcal{P} \setminus a})$ we define a winning strategy for $\langle D, \Gamma, \varphi \rangle$. The converse is similar.

□

CONCLUSION AND FUTURE WORK

In this very last section, we take an opportunity to recap our contribution and address what we believe are relevant research directions to be undertaken to strengthen the metatheory of Hyperteam semantics.

We recall that the original target of our work was to fix the *realizability* problem in *strategy logic* (SL). In the original semantics of SL, alternating quantifiers are treated in a classical way: the newly quantified strategy depends on the entirety of the previously quantified ones. This dependency is far too coarse to guarantee the finer dependency underlying realizability. The overall challenge is to master dependency specifications, and we ended up with *hyperteams*, a new mathematical structure that yields subtle ways of handling many sorts of dependencies.

Hyperteam semantics We recall that a hyperteam is a set of sets of assignments: this two-level structure inherits the concept of *team* introduced by [Hod97b] and widely considered in classical logic (see for instance [MSS11; Vää07]).

What a hyperteam provides, as opposed to a team, is the ability to represent the viewpoints of both existential and universal players. Indeed, hyperteams can be dualized, a crucial operation to obtain *compositional semantics* of logical statements, and that opens the door to inductive reasoning. Tightly coupled with the dualization, we defined cross-referenced hyperteam semantics to handle quantifiers and Boolean connectives, for many logical languages: first-order logic (Chapter 3), quantified temporal logic (Chapter 4), and plan logic (Chapter 5).

Along with hyperteam semantics, we associated a more traditional semantics that resembles second-order logic-like semantics, where quantifiers range over *response functions*. Note that this association is somehow agnostic of a particular underlying logical language.

Hyperteam semantics should not be confused with *multiteam* semantics [DHKMV18], which uses multisets of assignments instead of mere sets of assignments. This shift is useful for generalizing the application of team semantics to quantitative considerations. Another closely related framework is the one of *polyteam semantics* [HKV20], where a set of teams is used instead of a single team. Although a polyteam formally resembles

a hyperteam, the usage is entirely different, as no dualization operation is defined for polyteams. In fact, their goal is to *extend* the applicability of team semantics to address *embedded dependencies*.

Open problem on hyperteam semantics. Independently of any logical language, one may legitimately wonder whether or not operations on hyperteam semantics can be defined to avoid the use of the dual semantics. Without any dependency, a *cylindrification* operation (see Section 1.2.1) should be sufficient. We envision a way to extend this operation to handle dependencies leading to what we call a *constrained cylindrification*, and show it is equivalent to the sequence of operations dualization-extension-dualization used in our hyperteam semantics (Section 2.3).

We now keep on with the logical languages hyperteam semantics has been applied to.

ADIF. We considered simple dependencies in the same vein as *dependence/independence-friendly logic*, where teams have already proven useful for defining compositional semantics to reason *with* imperfect information. With hyperteam semantics, we introduced *alternating dependence/independence-friendly logic* (ADIF), which allows for meaningful constraints on both existential and universal quantifiers. We proved that ADIF is equipotent with SO and Team Logic and defined a game-theoretic semantics for ADIF sentences in prenex form.

Open problem for ADIF. The prenex assumption on ADIF formulas seems hard to alleviate, and we do not know yet if every formula enjoys a prenex normal form (i.e. whether it is equivalent to one in prenex form). We conjecture it is not the case, as our attempt to conduct an inductive approach fails because of a weird interplay between the semantics of Boolean operators and the dependency specifications. To fix this, we defined *restrained operators*, in essence Boolean operators subject to the same dependency constraints as quantified variables. It appeared that the constrained cylindrification operation, mentioned earlier, could be the missing piece to conclude, further motivating the need to prove the constrained cylindrification adequacy with respect to our original hyperteam semantics.

GFG-QPTL For the quantified propositional temporal logic (QPTL), we enforced quantified (propositional) variables to remain independent of the future of the computation. The resulting logic, *good-for-game QPTL* (GFG-QPTL), already displays a flavor

of strategic reasoning since quantifications over a proposition amounts to quantifying over a strategy. We proved that, in GFG-QPTL, quantifiers can commute by adapting their dependency constraints, making every formula equivalent to one with only a single alternation of quantifiers.

Future work on temporal logic. Other temporal logics can be considered and may lead to interesting results. For example, one could explore branching-time logics, such as *quantified computation tree logic* [LM14], or those with fix-point operators, such as *quantified μ -calculus* [Pin07].

Realizable strategic reasoning in PL Instead of directly applying the hyperteam semantics approach to strategy logic (SL) that quantifies over strategies, we leveraged the insights gained from GFG-QPTL and chose to focus on *plans*. While a strategy is a branching-time object (namely, a function that maps a history in the game to an action), a plan is a linear-time object (namely, a sequence of actions). The branching-time nature of strategies is recovered over plans through hyperteam semantics together with the tying operator we introduced. Recall that this operator filters out plans that cannot form part of a common strategy. This approach led us to defining *plan logic* (PL). We developed an associated game-theoretic semantics for PL, which turns out to be central for translating the conjunctive and disjunctive fragments of SL with timeline semantics (an *ad hoc* semantics for enforcing realizability [GBM20]) into PL. This game semantics is precious to solving the model-checking problem for PL with an 2-EXPTIME procedure.

Future work on PL. The translation of SL into PL applies only to the conjunctive and disjunctive fragments of SL. However, Gardy et al. [GBM20] managed to enforce realizability with a 2-EXPTIME model checking procedure for a slightly larger fragment of SL, named SL[EG]. Also they established that SL[EG] is the largest fragment one can expect. We conjecture that we have a way to translate the full SL[EG] fragment into PL.

Dependency matrices. We addressed a multiplayer extension of delay games by means of a dependency matrix that contains all the (possibly infinite) delay specifications between any two players. We introduced the dependency arena, where players must play when their dependencies are fulfilled, and we captured the uniformity property that a matrix imposes on the players' strategies. Even when considering only LTL formulas to

specify winning plays, solving the game defined by a matrix is undecidable. We exhibited a restriction on matrices that ensures a perfect-information game, sufficient to recover decidability.

Future work on dependency matrices. First, we conjecture that the game can be solved when all the considered delays are finite, even for imperfect information matrices. Second, one could design a logic with hyperteam semantics where strategy quantifiers are parameterized by a row of the dependency matrix. A full study of the obtained formalism seems worthwhile as the first logic to reason about “quantitative” dependencies.

About the idea of dependency. Since many more settings can benefit from using dependency that underlies complex system behavior in practice. We believe Hyperteam semantics to be a promising tool that can adapt to reasoning about more general modal logic domains like knowledge, belief, etc.

BIBLIOGRAPHY

- [ABM19] E. Acar, M. Benerecetti, and F. Mogavero, « Satisfiability in Strategy Logic Can Be Easier than Model Checking », *in: AAI'19*, AAAI Press, 2019, pp. 2638–2645.
- [AHK02] R. Alur, T.A. Henzinger, and O. Kupferman, « Alternating-Time Temporal Logic », *in: JACM* 49.5 (2002), pp. 672–713.
- [AHK97] R. Alur, T.A. Henzinger, and O. Kupferman, « Alternating-Time Temporal Logic », *in: FOCS'97*, IEEECS, 1997, pp. 100–109.
- [BBDM23a] D. Bellier et al., « Alternating (In)Dependence-Friendly Logic », *in: APAL* 174.10 (2023), 103315:1–58.
- [BBDM23b] D. Bellier et al., « Good-for-Game QPTL: An Alternating Hodges Semantics », *in: TOCL* 24.1 (2023), 4:1–57.
- [Bee+24] Maurice H. ter Beek et al., « Formal Methods in Industry », *in: Form. Asp. Comput.* (2024), Just Accepted, ISSN: 0934-5043, DOI: 10.1145/3689374, URL: <https://doi.org/10.1145/3689374>.
- [Ber66] Robert Berger, *The undecidability of the domino problem*, 66, American Mathematical Soc., 1966.
- [BG86] A. Blass and Y. Gurevich, « Henkin Quantifiers and Complete Problems », *in: APAL* 32.1 (1986), pp. 1–16.
- [BGM15] P. Bouyer, P. Gardy, and N. Markey, « Weighted Strategy Logic with Boolean Goals Over One-Counter Games », *in: FSTTCS'15*, LIPIcs 45, Leibniz-Zentrum fuer Informatik, 2015, pp. 69–83.
- [BHM09] Armin Biere, Marijn Heule, and Hans van Maaren, *Handbook of satisfiability*, vol. 185, IOS press, 2009.
- [BK08] Christel Baier and Joost-Pieter Katoen, *Principles of model checking*, MIT press, 2008.

-
- [BL19] U. Boker and K. Lehtinen, « Good for Games Automata: From Nondeterminism to Alternation », *in: CONCUR'19*, vol. 140, LIPIcs 140, Leibniz-Zentrum fuer Informatik, 2019, 19:1–19.
- [BMM13] M. Benerecetti, F. Mogavero, and A. Murano, « Substructure Temporal Logic », *in: LICS'13*, IEEECS, 2013, pp. 368–377.
- [BMM15] M. Benerecetti, F. Mogavero, and A. Murano, « Reasoning About Substructures and Games », *in: TOCL 16.3* (2015), 25:1–46.
- [BPS22] Dylan Bellier, Sophie Pinchinat, and François Schwarzentruher, « Dependency Matrices for Multiplayer Strategic Dependencies », *in: 42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2022)*, Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2022.
- [Bra13] J. Bradfield, « Team Building in Dependence », *in: CSL'13*, LIPIcs 23, Leibniz-Zentrum fuer Informatik, 2013, pp. 116–128.
- [Bus98a] S.R. Buss, « An Introduction to Proof Theory », *in: Handbook of Proof Theory*, Elsevier, 1998, pp. 1–78.
- [Bus98b] S.R. Buss, *Handbook of Proof Theory*, Studies in Logic and the Foundations of Mathematics 137, Elsevier, 1998.
- [CFHH19] N. Coenen et al., « The Hierarchy of Hyperlogics », *in: LICS'19*, IEEECS, 2019, pp. 1–13.
- [CFKMRS14] M.R. Clarkson et al., « Temporal Logics for Hyperproperties », *in: POST'14*, LNCS 8414, Springer, 2014, pp. 265–284.
- [CFLMZ21] Mingshuai Chen et al., « Indecision and delays are the parents of failure—taming them algorithmically by synthesizing delay-resilient control », *in: Acta Informatica 58.5* (2021), pp. 497–528.
- [CHP10] K. Chatterjee, T.A. Henzinger, and N. Piterman, « Strategy Logic », *in: IC 208.6* (2010), pp. 677–693.
- [Chu56] A. Church, *Introduction to Mathematical Logic*, Princeton Mathematical Series 17, Princeton University Press, 1956.
- [CS10] Michael R Clarkson and Fred B Schneider, « Hyperproperties », *in: Journal of Computer Security 18.6* (2010), pp. 1157–1210.

-
- [Dam99] D. Dams, « Flat Fragments of CTL and CTL*: Separating the Expressive and Distinguishing Powers », *in: LJIGPL 7.1* (1999), pp. 55–78.
- [DGL16a] Stéphane Demri, Valentin Goranko, and Martin Lange, *Temporal logics in computer science: finite-state systems*, vol. 58, Cambridge University Press, 2016.
- [DGL16b] Stéphane Demri, Valentin Goranko, and Martin Lange, *Temporal Logics in Computer Science: Finite-State Systems*, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2016.
- [DHKMOV18] Arnaud Durand et al., « Approximation and dependence via multiteam semantics », *in: Annals of Mathematics and Artificial Intelligence* 83 (2018), pp. 297–320.
- [EC82] E Allen Emerson and Edmund M Clarke, « Using branching time temporal logic to synthesize synchronization skeletons », *in: Science of Computer programming 2.3* (1982), pp. 241–266.
- [EG22] S. Enqvist and V. Goranko, « The Temporal Logic of Coalitional Goal Assignments in Concurrent Multiplayer Games », *in: TOCL 23.4* (2022), 21:1–58.
- [EH86] E Allen Emerson and Joseph Y Halpern, « “Sometimes” and “not never” revisited: on branching versus linear time temporal logic », *in: Journal of the ACM (JACM)* 33.1 (1986), pp. 151–178.
- [EJ88] E.A. Emerson and C.S. Jutla, « The Complexity of Tree Automata and Logics of Programs (Extended Abstract) », *in: FOCS’88, IEEECS*, 1988, pp. 328–337.
- [EJ91] E.A. Emerson and C.S. Jutla, « Tree Automata, muCalculus, and Determinacy », *in: FOCS’91, IEEECS*, 1991, pp. 368–377.
- [EJS93] E.A. Emerson, C.S. Jutla, and A.P. Sistla, « On Model-Checking for Fragments of muCalculus », *in: CAV’93, LNCS 697, Springer*, 1993, pp. 385–396.
- [End70] H.B. Enderton, « Finite Partially-Ordered Quantifiers », *in: MLQ 16.8* (1970), pp. 393–397.
- [FG17] P. Fritz and J. Goodman, « Counting Impossibles », *in: Mind* 126.504 (2017), pp. 1063–1108.

-
- [FH16] B. Finkbeiner and C. Hahn, « Deciding Hyperproperties », *in: CONCUR'16*, LIPIcs 59, Leibniz-Zentrum fuer Informatik, 2016, 13:1–14.
- [Fin16] Bernd Finkbeiner, « Synthesis of Reactive Systems. », *in: Dependable Software Systems Engineering* 45 (2016), pp. 72–98.
- [Fre01] T. French, « Decidability of Quantified Propositional Branching Time Logics », *in: AAI'01*, LNCS 2256, Springer, 2001, pp. 165–176.
- [FRS15] B. Finkbeiner, M.N. Rabe, and C. Sánchez, « Algorithms for Model Checking HyperLTL and HyperCTL* », *in: CAV'15*, LNCS 9206, Springer, 2015, pp. 30–48.
- [FS10] B. Finkbeiner and S. Schewe, « Coordination Logic », *in: CSL'10*, LNCS 6247, Springer, 2010, pp. 305–319.
- [FWZ23] M. Fränzle, S. Winter, and M. Zimmermann, « Strategies Resilient to Delay: Games under Delayed Control vs. Delay Games. », *in: GandALF* 390 (2023), pp. 220–235.
- [FZ16] B. Finkbeiner and M. Zimmermann, « The First-Order Logic of Hyperproperties », *in: STACS'17*, LIPIcs 66, Leibniz-Zentrum fuer Informatik, 2016, 30:1–14.
- [Gal12] P. Galliani, « Inclusion and Exclusion Dependencies in Team Semantics - On Some Logics of Imperfect Information », *in: APAL* 163.1 (2012), pp. 68–84.
- [GBM18] P. Gardy, P. Bouyer, and N. Markey, « Dependences in Strategy Logic », *in: STACS'18*, LIPIcs 96, Leibniz-Zentrum fuer Informatik, 2018, 34:1–15.
- [GBM20] P. Gardy, P. Bouyer, and N. Markey, « Dependences in Strategy Logic », *in: TCS* 64.3 (2020), pp. 467–507.
- [GHW13] J. Gutierrez, P. Harrenstein, and M. Wooldridge, « Iterated Boolean Games », *in: IJCAI'13, IJCAI' & AAI', 2013*, pp. 932–938.
- [GHW15] J. Gutierrez, P. Harrenstein, and M. Wooldridge, « Iterated Boolean Games », *in: IC* 242 (2015), pp. 53–79.
- [Grä+05] E. Grädel et al., *Finite Model Theory and Its Applications*, Texts in Theoretical Computer Science, Springer, 2005.

-
- [GS53] D. Gale and F.M. Stewart, « Infinite Games with Perfect Information », *in: Contributions to the Theory of Games (vol. II)*. Vol. 28, Princeton University Press, 1953, pp. 245–266.
- [GTW02] E. Grädel, W. Thomas, and T. Wilke, *Automata, Logics, and Infinite Games: A Guide to Current Research*, LNCS 2500, Springer, 2002.
- [GV13] E. Grädel and J.A. Väänänen, « Dependence and Independence », *in: SL 101.2* (2013), pp. 399–410.
- [GV14] V. Goranko and S. Vester, « Optimal Decision Procedures for Satisfiability in Fragments of Alternating-time Temporal Logics », *in: AIML'14*, College Publications, 2014, pp. 234–253.
- [HA28] D. Hilbert and W. Ackermann, *Grundzüge der Theoretischen Logik*, Die Grundlehren der Mathematischen Wissenschaften, Springer, 1928.
- [Har79] D. Harel, « Characterizing Second Order Logic with First Order Quantifiers », *in: MLQ 25.25-29* (1979), pp. 419–422.
- [Hei67] J. van Heijenoort, *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*, Harvard University Press, 1967.
- [Hel89] L. Hella, « Definability Hierarchies of Generalized Quantifiers », *in: APAL 43.3* (1989), pp. 235–271.
- [Hen61] L. Henkin, « Some Remarks on Infinitely Long Formulas », *in: IM'61*, Pergamon Press, 1961, pp. 167–183.
- [Hin96] J. Hintikka, *The Principles of Mathematics Revisited*, CUP, 1996.
- [HKV20] Miika Hannula, Juha Kontinen, and Jonni Virtema, « Polyteam semantics », *in: Journal of Logic and Computation 30.8* (2020), pp. 1541–1566.
- [Hod97a] Wilfrid Hodges, « Compositional semantics for a language of imperfect information », *in: Logic Journal of the IGPL 5.4* (1997), pp. 539–563.
- [Hod97b] Wilfrid Hodges, « Some strange quantifiers », *in: Structures in logic and computer science* (1997), pp. 51–65.
- [HP06] T.A. Henzinger and N. Piterman, « Solving Games Without Determinization », *in: CSL'06*, LNCS 4207, Springer, 2006, pp. 395–410.
- [HR76] J. Hintikka and V. Rantala, « A New Approach to Infinitary Languages », *in: AML 10.1* (1976), pp. 95–115.

-
- [HS89] J. Hintikka and G. Sandu, « Informational Independence as a Semantical Phenomenon », *in: ICLMPS'89*, Elsevier, 1989, pp. 571–589.
- [HS97] J. Hintikka and G. Sandu, « Game-Theoretical Semantics », *in: Handbook of Logic and Language*, North-Holland & Elsevier, 1997, pp. 361–410.
- [HV94] H. Heikkilä and J.A. Väänänen, « Reflection of Long Game Formulas », *in: MLQ* 40.3 (1994), pp. 381–392.
- [Imm81] N. Immerman, « Number of Quantifiers is Better Than Number of Tape Cells », *in: JCSS* 22.3 (1981), pp. 384–406.
- [Imm99] N. Immerman, *Descriptive Complexity*, Graduate Texts in Computer Science, Springer, 1999.
- [Kai11] L. Kaiser, *Logic and Games on Automatic Structures - Playing with Quantifiers and Decompositions*, LNCS 6810, Springer, 2011.
- [KM95] M. Krynicki and M. Mostowski, « Henkin Quantifiers », *in: Quantifiers: Logics, Models and Computation, Volume I*, Springer, 1995, pp. 193–262.
- [KMOV17] Andreas Krebs et al., « Team semantics for the specification and verification of hyperproperties », *in: arXiv preprint arXiv:1709.08510* (2017).
- [KN01] B. Khoussainov and A. Nerode, *Automata Theory and Its Applications*, Birkhauser, 2001.
- [KN09] J. Kontinen and V. Nurmi, « Team Logic and Second-Order Logic », *in: WOLLIC'09*, LNCS 5514, Springer, 2009, pp. 230–241.
- [Kni02] John C Knight, « Safety critical systems: challenges and directions », *in: Proceedings of the 24th international conference on software engineering*, 2002, pp. 547–550.
- [Kol85] P.G. Kolaitis, « Game Quantification », *in: Handbook of Model-Theoretic Logics*, Springer, 1985, pp. 365–421.
- [Kuh50] H.W. Kuhn, « Extensive Games », *in: PNAS* 36.1 (1950), pp. 570–576.
- [Kuh53] H.W. Kuhn, « Extensive Games and the Problem of Information », *in: Contributions to the Theory of Games (vol. II)*. Vol. 28, Princeton University Press, 1953, pp. 193–216.
- [KV09] J. Kontinen and J.A. Väänänen, « On Definability in Dependence Logic », *in: JLLI* 18.3 (2009), pp. 317–332.

-
- [KVBSV13] Timothy Kam et al., *Synthesis of finite state machines: functional optimization*, Springer Science & Business Media, 2013.
- [KVW00] O. Kupferman, M.Y. Vardi, and P. Wolper, « An Automata Theoretic Approach to Branching-Time Model Checking », *in: JACM* 47.2 (2000), pp. 312–360.
- [KVW01] Orna Kupferman, Moshe Y Vardi, and Pierre Wolper, « Module checking », *in: Information and Computation* 164.2 (2001), pp. 322–344.
- [KZ15] F. Klein and M. Zimmermann, « What are strategies in delay games? Borel determinacy for games with lookahead », *in: arXiv preprint arXiv:1504.02627* (2015).
- [KZ17] F. Klein and M. Zimmermann, « How Much Lookahead is Needed to Win Infinite Games? », *in: Logical Methods in Computer Science* 12 (2017).
- [LM14] F. Laroussinie and N. Markey, « Quantified CTL: Expressiveness and Complexity », *in: LMCS* 10.4 (2014), pp. 1–45.
- [Mar75] A.D. Martin, « Borel Determinacy », *in: AM* 102.2 (1975), pp. 363–371.
- [Mar85] A.D. Martin, « A Purely Inductive Proof of Borel Determinacy », *in: RT'82, SPM'42, AMS and ASL, 1985*, pp. 303–308.
- [Maz04] Laurent Mazaré, « Using unification for opacity properties », *in: Proceedings of the 4th IFIP WG1 7* (2004), pp. 165–176.
- [MK20] Ankur Maurya and Divya Kumar, « Reliability of safety-critical systems: A state-of-the-art review », *in: Quality and Reliability Engineering International* 36.7 (2020), pp. 2547–2568.
- [MMPV14] Fabio Mogavero et al., « Reasoning about strategies: On the model-checking problem », *in: ACM Transactions on Computational Logic (TOCL)* 15.4 (2014), p. 34.
- [MMPV17] F. Mogavero et al., « Reasoning About Strategies: On the Satisfiability Problem », *in: LMCS* 13.1:9 (2017), pp. 1–37.
- [Mos84] A.W. Mostowski, « Regular Expressions for Infinite Trees and a Standard Form of Automata », *in: SCT'84, LNCS* 208, Springer, 1984, pp. 157–168.
- [Mos91] A.W. Mostowski, *Games with Forbidden Positions*, tech. rep., University of Gdańsk, Gdańsk, Poland, 1991.

-
- [MSS11] A.L. Mann, G. Sandu, and M. Sevenster, *Independence-Friendly Logic - A Game-Theoretic Approach*, CUP, 2011.
- [Neu28] J. von Neumann, « Zur Theorie der Gesellschaftsspiele », *in: MA 100.1* (1928), pp. 295–320.
- [NM44] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, 1944.
- [Pin07] Sophie Pinchinat, « A Generic Constructive Solution for Concurrent Games with Expressive Constraints on Strategies », *in: Automated Technology for Verification and Analysis, 5th International Symposium, ATVA 2007, Tokyo, Japan, October 22-25, 2007, Proceedings*, ed. by Kedar S. Namjoshi et al., vol. 4762, Lecture Notes in Computer Science, Springer, 2007, pp. 253–267, DOI: 10.1007/978-3-540-75596-8_19, URL: https://doi.org/10.1007/978-3-540-75596-8_19.
- [Pit06] N. Piterman, « From Nondeterministic Buchi and Streett Automata to Deterministic Parity Automata », *in: LICS'06, IEEECS, 2006*, pp. 255–264.
- [Pit07] Nir Piterman, « From nondeterministic Büchi and Streett automata to deterministic parity automata », *in: Logical Methods in Computer Science* 3 (2007).
- [Pnu77] Amir Pnueli, « The temporal logic of programs », *in: 18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, IEEE, 1977, pp. 46–57.
- [PP04] D. Perrin and J. Pin, *Infinite Words*, Pure and Applied Mathematics, Elsevier, 2004.
- [PR89] A. Pnueli and R. Rosner, « On the Synthesis of a Reactive Module », *in: POPL'89*, ACM, 1989, pp. 179–190.
- [PRA01] Gary Peterson, John Reif, and Salman Azhar, « Lower bounds for multi-player noncooperative games of incomplete information », *in: Computers & Mathematics with Applications* 41.7-8 (2001), pp. 957–992.
- [SH01] G. Sandu and J. Hintikka, « Aspects of Compositionality », *in: JLLI 10.1* (2001), pp. 49–61.

-
- [Sha91] S. Shapiro, *Foundations Without Foundationalism: A Case for Second-Order Logic*, Oxford University Press, 1991.
- [Sin94] Munindar P Singh, *Multiagent systems*, Springer, 1994.
- [Sis83] A.P. Sistla, « Theoretical Issues in the Design and Verification of Distributed Systems », PhD thesis, Harvard University, Cambridge, MA USA, 1983.
- [Sto76] L.J. Stockmeyer, « The Polynomial-Time Hierarchy », *in: TCS 3.1* (1976), pp. 1–22.
- [SVW87] A Prasad Sistla, Moshe Y Vardi, and Pierre Wolper, « The complementation problem for Büchi automata with applications to temporal logic », *in: Theoretical Computer Science 49.2-3* (1987), pp. 217–237.
- [Tar36] A. Tarski, « The Concept of Truth in Formalized Languages », *in: Logic, Semantics, Metamathematics*, Oxford University Press, 1936, pp. 152–278.
- [Tar44] A. Tarski, « The Semantic Conception of Truth: and the Foundations of Semantics », *in: PPR 4.3* (1944), pp. 341–376.
- [TKH12] Wolfgang Thomas, Lukasz Kaiser, and Michael Holtmann, « Degrees of Lookahead in Regular Infinite Games », *in: Logical Methods in Computer Science 8* (2012).
- [Vää07] J.A. Väänänen, *Dependence Logic: A New Approach to Independence Friendly Logic*, vol. 70, London Mathematical Society Student Texts, CUP, 2007.
- [VW86a] M.Y. Vardi and P. Wolper, « An Automata-Theoretic Approach to Automatic Program Verification », *in: LICS'86, IEEECS*, 1986, pp. 332–344.
- [VW86b] M.Y. Vardi and P. Wolper, « Automata-Theoretic Techniques for Modal Logics of Programs », *in: JCSS 32.2* (1986), pp. 183–221.
- [Wal70] W.J. Walkoe, « Finite Partially-Ordered Quantification », *in: JSL 35.4* (1970), pp. 535–555.
- [Wol55] P. Wolfe, « The Strict Determinateness of Certain Infinite Games », *in: PJM 5* (1955), pp. 841–847.

-
- [Zie98] W. Zielonka, « Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees », *in: TCS 200.1-2* (1998), pp. 135–183.

INDEX

- ADF, 83
- ADIF, 81, 82
 - Semantic relation, 85
 - Syntax, 82
- AIF, 83
- CGS, 184
 - Action, 184
 - Action profile, 184
 - Agent, 184
 - Initial position, 184
 - Label function, 184
 - Position, 184
 - Transition, 184
- DIF, 30
 - DFL, 29
 - IFL, 30
- FO Structure, 25
- LTL, 127
 - LTL assignment, 127
- QPTL, 128
- SL
 - Boolean goal, 185
 - Conjunctive goal, 186
 - Disjunctive goal, 186
 - One goal, 186
 - Semantics, 185
 - Timeline semantics, 185
- SO, 96
- Adequacy, 49
- Alternating dependence-friendly logic, 83
- Alternating Dependence/Independence-Friendly Logic, 81, 82
- Alternating independence logic, 83
- Alternation flag, 37
- Arity, 24
- Assignment, 25
- Bucket of Response Functions, 106
- Commitment, 225
- Concurrent game structure, 184
- Conjunction, 23
- Delay, 225
- Dependence/Independence-friendly logic, *see* DIF
- Dependency, 29
- Dependency context, 52
 - Acyclic, 52
- Dependency game, 225
 - Arena, 229
 - Configuration, 225
- Dependency matrix
 - Graph, 232
 - Progressing, 232
- Dependency matrix, 225
 - Perfect information, 237

Progress, 230
 Progressing, 226
 Disjunction, 23
 Splitjunction, 30
 Filtering, 191
 First-Order Logic, 23
 FO Structure
 Domain, 25
 Formula
 FO formula, 24
 Flat, 186
 Pseudo sentence, 83
 Sentence, 24
 Free variable, 82
 Generalized extension, 48
 Goal formula, 185
 Hyperteam, 38
 Choice function, 40
 Dualisation, 40
 Empty hyperteam, 38
 Extension, 40, 42
 Null hyperteam, 38
 partition, 40
 Proper hyperteam, 38
 Semantic relation, 44
 Trivial hyperteam, 38
 Implication, 23
 Leak of information, 89
 Linear-Time Temporal Logic, 127
 Lookahead, 225
 Meta semantics, 51
 Negation, 23
 Negation normal form, 45
 Operator agent, 195
 Plan, 183
 Predicate, 23, 24
 Prenex normal form, 91
 Proposition, 126
 Quantified Propositional Temporal
 Logic, 128
 Quantifier, 23
 Existential, 23
 Quantifier prefix, 48
 Universal, 23
 Realizable
 Realizable quantification, 129
 Response function, 28, 42
 Uniform, 29
 Second-order logic, 96
 Signature, 24
 Support variable, 82
 Symbols
 Function symbols, 24
 Predicate symbols, 24
 Team, 28
 Cylindrification, 28
 Extension, 28
 Partitionning, 28
 Term, 23, 24
 Tying, 187
 Bindings tied in, 189
 Variable, 23
 Free variables, 26

Titre : Raisonnement Stratégique avec Dépendances : Logique d'Hyperteam, Stratégies Réalisables, Matrices de Dépendance

Mot clés : Raisonnement Stratégique, Logique pour Dépendance, Logique Temporelle

Résumé : Strategy Logique (SL, en abrégé) a été introduite par Chatterjee, Henzinger et Piterman comme un outil pour raisonner sur les stratégies. Ce formalisme puissant peut exprimer des notions stratégiques complexes telles que l'existence d'équilibre de Nash dans un jeu ou l'existence d'une stratégie défensive, etc. Cependant, cette expressivité vient à un prix, car le problème de vérification associé a une complexité non élémentaire et le problème de satisfaisabilité est indécidable. Une analyse soignée montre que cette haute complexité est liée à la nature monolithique des quantificateurs stratégiques dans SL, où les stratégies sont des citoyens de premier rang de la logique. En conséquence, dans une formule, une stratégie quantifiée dépend de toutes les stratégies quantifiées avant elle, dans leur intégralité, ce qui inclut leurs décisions futures. Cette caractéristique monolithique des quantificateurs SL empêche les stratégies considérées de ne dépendre que de l'histoire du jeu en cours et peut mener à des solutions avec des stratégies irréalisables, c'est-à-dire qui ne peuvent pas être mises en œuvre. Il est donc raisonnable d'investiguer de nouvelles sémantiques ou logiques qui garantissent la réalisabilité des stratégies et réduisent les complexités de problèmes de décision.

Dans notre travail, nous nous sommes inspirés des logiques dotées de sémantiques d'équipe, qui se basent sur une équipe (un ensemble d'assignation) au lieu d'une seule assignation. De manière formelle, une équipe satisfait une formule atomique si chaque assignation dans l'équipe satisfait la dite formule. Les équipes sont des objets pratiques pour représenter toutes les situations possibles –

de sorte que les sémantiques d'équipe capturent l'imperfection de l'information. Elles permettent également d'exprimer des contraintes sur les valeurs des variables quantifiées. Par exemple, la propriété "la valeur d'une variable x dépend uniquement de la valeur d'une autre variable y " est caractérisée par les équipes dans lesquelles chaque deux assignations qui donnent la même valeur à y , donnent également la même valeur à x . Cependant, les logiques avec sémantiques d'équipe sont souvent indéterminées (il existe des formules ni vraies ni fausses) car les contraintes de dépendance s'appliquent uniquement aux variables qui ont le même type de quantificateur (soit existentiel ou universel).

L'idée centrale de notre travail est de remplacer les sémantiques d'équipe par des sémantiques hyperteam, où une hyperteam est un ensemble d'équipes. Cette approche permet d'exprimer la dépendance de manière compositionnelle avec un traitement symétrique des quantificateurs existentiels et universels. Pour commencer, nous avons appliqué cette approche à plusieurs logiques : tout d'abord, aux langages du premier ordre en concevant une logique Alternating Dependence/Independence Friendly (ADIF), puis à la Logique Temporelle Propositionnelle Quantifiée (QPTL). Ensuite, nous avons adapté les sémantiques hyperteam pour le raisonnement stratégique en considérant des plans (séquences d'actions), des objets qui sont beaucoup plus simples que les stratégies, comme des citoyens de premier rang de la logique.

Enfin, à part nos contributions sur les sémantiques hyperteam pour diverses logiques, nous développons un nouveau genre de jeux avec des retards entre les actions des joueurs,

qui généralise les Jeux de Retard et qui offre une autre façon d'exprimer les dépendances entre les stratégies dans les jeux à plusieurs joueurs.

Title: Strategic Reasoning with Dependencies: Hyperteam Logics, Realizable Strategies, Dependency Matrices

Keywords: Strategic reasoning, Logic for Dependencies, Temporal logic

Abstract: Strategy Logic (SL for short) has been introduced by Chatterjee, Henzinger and Piterman as a logic to reason about strategies. This powerful setting can express complex strategic notions like the existence of a Nash equilibrium in a game or the existence of a defensive strategy, etc. However, this expressiveness comes with a price as the associated model-checking problem has a non-elementary complexity and the satisfiability problem is undecidable. A careful analysis relates this high complexity with the monolithic nature of strategy quantifiers in SL, where strategies are the first class citizen of the logic. Consequently, in a formula, a quantified strategy not surprisingly depends on all strategy quantified before, but on their entirety which include counter-factuals and future decisions. This monolithic feature of SL-quantifiers prevents from enforcing strategies to depend only .g. on the history of the current play and may lead to solutions with unrealizable strategies, i.e. that cannot be implemented. It is therefore reasonable to investigate new semantics/new logics that guarantee the realizability of strategies and ideally that reduce decision problem complexities.

In our work, we took inspiration from logics with so-called team semantics, that rely on a team (a set of assignments) instead of a single assignment. Formally, a team satisfies an atomic formula if every assignment in the team satisfies the said formula. Teams are convenient objects to represent all possible situations – so that the team semantics captures imperfect information. They also enable

to express constraints on the quantified variable values. For instance, property "the value of a variable x depends only on the value of another variable y " is characterized by teams in which every two assignments giving the same value to y , also give the same value to x . However, logics with team semantics generally lead to undeterminacy (there are formulas neither true nor false) because dependency constraints apply only to variables with the same quantifier-type (either existential or universal).

The core idea of our work is to replace team semantics by hyperteam semantics, where a hyperteam is a set of teams. This approach enables to express dependence in a compositional manner with symmetric treatment of existential and universal quantifiers. For a start, We successfully apply the approach to several logical settings: first, to first-order-like languages by designing Alternating Dependence/Independence Friendly Logic (ADIF), and second to Quantified Propositional Temporal Logic (QPTL). Next, we adapt hyperteam semantics to strategic reasoning by considering plans (sequences of actions), objects that are much simpler than strategies, as first-class citizens of the logic.

Aside our contributions on hyperteam semantics for various logics, we develop a setting with a new kind of games with delays between player actions, that generalizes Delay Games, and that offers another way to express dependencies between strategies in multi-player games.