

**Branches of
imperfect information:
logic, games, and computation**

Merlijn Sevenster

**Branches of
imperfect information:
logic, games, and computation**

ILLC Dissertation Series DS-2006-06



INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation

Universiteit van Amsterdam

Plantage Muidergracht 24

1018 TV Amsterdam

phone: +31-20-525 6051

fax: +31-20-525 5206

e-mail: illc@science.uva.nl

homepage: <http://www.illc.uva.nl/>

**Branches of
imperfect information:
logic, games, and computation**

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. mr. P. F. van der Heijden
ten overstaan van een door het college voor
promoties ingestelde commissie, in het openbaar
te verdedigen in de Aula der Universiteit
op woensdag 4 oktober 2006, te 12.00 uur

door

Merlijn Sevenster

geboren te 's Gravenhage.

Promotores: prof. dr. J. F. A. K. van Benthem
lector dr. P. van Emde Boas

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

This research was financially supported by the Netherlands Organization for Scientific Research (NWO), in the scope of the project *Imperfect Information Games; Models and Analysis*, 600.065.120.01N49.

Copyright © 2006 by Merlijn Sevenster

Cover design by David de Nood. The cover shows a fragment of the etch *Three trees* by Rembrandt, 1643.

Printed and bound by PrintPartners Ipskamp.

ISBN-10: 90-5776-157-2

ISBN-13: 978-90-5776-157-7

Contents

Acknowledgments	ix
1 Introduction	1
2 Prerequisites	9
2.1 Basic notation	9
2.2 Game theory	10
2.3 Logic	12
2.4 Computational complexity	18
2.5 Descriptive complexity	20
3 Fragments of IF logic	25
3.1 Introduction	25
3.2 Prerequisites	27
3.3 A proper rule book for IF games	39
3.4 Perfect recall and IF logic	42
3.5 Modal logic and IF logic	53
3.5.1 Uniformity interpretation for modal logic	53
3.5.2 The modal fragment of IF logic	57
3.5.3 $\mathbf{IFV}^{\mathbf{ML}}$ is undecidable	62
3.6 Concluding remarks	72
4 Partially ordered connectives	75
4.1 Introduction	75
4.2 GTS for partially ordered prefixes	78
4.3 Logics with partially ordered connectives	82
4.4 Related research	84
4.4.1 \mathbf{D} and complete problems	84
4.4.2 Partially ordered connectives coined	85

4.4.3	Normal form theorem for Henkin quantifiers	86
4.4.4	Finite model theory for IF logic	86
4.5	Translating \mathbf{D}_k into $\Sigma_{1,k}^1$	87
4.6	A characterization of \mathbf{D}_k	89
4.7	Applications of Theorem 4.6.7	97
4.7.1	Strict hierarchy result	97
4.7.2	On linear ordered structures $\mathbf{D} = \Sigma_1^1$	99
4.8	Ehrenfeucht-Fraïssé game for \mathbf{D}	101
4.8.1	Comparison game for \mathbf{FO}	102
4.8.2	Comparison game for $\mathbf{M}\Sigma_1^1$	103
4.8.3	Comparison games for \mathbf{D}	104
4.9	Non-expressibility result for \mathbf{D}	106
4.10	Descriptive complexity of $\mathcal{L}(\mathbf{D})$ and $\mathcal{L}(\mathbf{H})$	108
4.10.1	$\mathcal{L}(\mathbf{D})$ and $\mathcal{L}(\mathbf{H})$ capture $\mathbf{P}_{\parallel}^{\mathbf{NP}}$	108
4.10.2	Aftermath	111
4.11	Concluding remarks	113
5	Branching quantifiers	115
5.1	Introduction	115
5.2	Prerequisites	118
5.3	Strategic games and branching quantifiers	120
5.3.1	GTS for branching quantifiers	120
5.3.2	Contemplations on the strategic framework	125
5.4	Related research	127
5.4.1	Van Benthem’s semantic automata	127
5.4.2	Expression complexity of Hintikka sentences	129
5.5	Complexity of natural language quantifiers	129
5.6	Branching quantifiers and \mathbf{NP}	132
5.7	More complexity of quantifiers	134
5.7.1	Every... a different...	134
5.7.2	A few... all...	135
5.7.3	Disjoint halves	137
5.8	Concluding remarks	138
6	Scotland Yard	141
6.1	Introduction	141
6.2	Scotland Yard formalized	146
6.3	A perfect information Scotland Yard game	151
6.4	An effective equivalence	155
6.4.1	Scotland Yard and Scotland Yard-PI are isomorphic	155
6.4.2	Backwards induction algorithms	158
6.5	Scotland Yard is \mathbf{PSPACE} -complete	163
6.6	Ignorance is (computational) bliss	171

6.7 Concluding remarks	177
7 Conclusions	179
A The boring bits of Chapter 6	183
Index	199
List of Symbols	203
Samenvatting	205

Acknowledgments

Although some of this dissertation's details may slip my mind, I'm sure I'll carry with me forever the friendly and inspiring atmosphere at the ILLC. I thank everybody at the ILLC for creating this special atmosphere.

First and foremost, I wish to thank my promotores, Johan van Benthem and Peter van Emde Boas, for hiring me on the InIGMA project. I enjoyed doing research under their supervision, and I benefited greatly from the ways in which Johan and Peter complement one another.

I thank Johan for the pleasant conversations we've had and the numerous ideas he unfolded on his white board. I have a great admiration for Johan's never-ceasing energy to motivate and inspire fellow logicians, including myself. The lion's share of this dissertation's contents are inspired by Johan's thinking, directly or indirectly. It is through Johan that I got in touch with a number of the world's most eminent logicians, for which I thank him. Also, I thank Johan for taking me along to Stanford University as a visiting scholar. I treasure my memories of being part of this great American institute.

I thank Peter for being my daily supervisor, meaning that I could always drop by with questions of any kind. Invariably, Peter listened to my problems and the difficulties I encountered solving them. During these sessions, I was frequently impressed by Peter's ability to understand my problems instantaneously, despite the fact that they weren't clear to me and correspondingly ill-formulated. I got to know Peter as an unique, humorous, and above all, loyal person.

I thank Gabriel Sandu for arranging my enjoyable stay in Helsinki, and for the encouraging discussions we've had ever since. Also I'm greatly indebted to Gabriel for the detailed comments he gave on an earlier version of this dissertation. I thank him for being member of my assessment committee.

I also thank Krzysztof Apt, Harry Buhrman, Jeroen Groenendijk, and Jouko Väänänen for their willingness to assess my dissertation.

I thank Tero Tulenheimo for the good teamwork. Although our backgrounds are different, we worked on a variety of problems, and, importantly, in a very

stimulating atmosphere. During one of our numerous email conversations, I was thrilled when I realized that there were actually two human beings working on IF modal logic simultaneously.

I thank Sjoerd Druiven for his companionship during the beginning of the project that we embarked on together.

I thank Joost Joosten, Olivia Ladinig, Sieuwert van Otterloo, Yoav Seginer, and Neta Spiro, colleagues with whom I shared an office. I could always turn to you with technical problems and personal frustrations—thanks.

Special thanks are due to Aline K. Honingh, who has been listening with warm interest every time I would speak out my heart. I do hope our friendship will last and that we'll have many more *kwebbelingen*—*heel gezellig!*

I thank my Ogden Avenue roommates: Loes Olde Loohuis and Olivier Roy. Most likely, Loes is the only person occupied with balancing a spoon on the tip of her nose as I'm answering one of her own questions about logic. I thank Olivier for the action, the encouraging words, and the wonderful cooking. It was a pleasure discovering San Francisco with you guys.

I thank Marian Counihan, Ingrid van Loon, and Reut Tsarfaty with whom I worked in fine harmony on three issues of the ILLC magazine.

I thank the sandwich crew consisting of, amongst others, Ulle Endriss, Eric Pacuit, Leigh Smith, Jakub Szymanik, and Andreas Witzel.

I thank the following persons, for different reasons: Nick “Oh no” Bezhanishvili, Stefan Bold, Boudewijn de Bruin, Balder ten Cate, Francien Dechesne, Hartmut Fitz, Patrick Girard, René Goedman, Alistair Isaac, Theo Janssen, Tanja Kassenaar, Barteld Kooi, Clemens “Herr” Kupke, Fenrong Liu, Benedikt Löwe, Jessica Pogorzelski, Robert van Rooij, Tomasz Sadzik, Brian Semmes, Leen Torenvliet, Marjan Veldhuisen, Frank Veltman, and Jelle Zuidema.

I thank Jules Alberga for his good and professional care. Jules was *het geluk bij mijn knieongeluk*.

It is with the warmest of feelings that I thank all my friends. It was you who I missed the most in California. Special thanks are due to the Floraboys and Floragirls: you are *dubbele primaatjes*. Furthermore, I want to mention my oldest friends: Berend ter Borg and Niels Meerdink. I wish to express my greatest gratitude to Berend ter Borg for repairing my English in parts of this dissertation.

I thank my family for their support and care. Especially, I wish to thank with my whole heart my parents, my sister Dieuwke, my brother Bart, Leny, and Oma.

Amsterdam
August, 2006.

Merlijn Sevenster

Background. At the center of logic, computation theory, and game theory is an interest in information and in the ways information can be processed correctly, mechanically, and rationally. The three disciplines use their own frameworks and terminology, but as information is their primary topic of investigation, structures and concepts from one discipline are open to analysis by means of tools from the other disciplines. Some of these analyses were successful to the point of giving rise to areas of research of their own. For example, research by Lorenzen and Hintikka, Fagin (1974), and Berlekamp, Conway, and Guy (1982) shaped the areas of *game-theoretic semantics* (games and logic), *descriptive complexity theory* (logic and computation), and *combinatorial game theory* (computation and games), respectively.

Game-theoretic semantics for logics aim to give meaning to the logics' components in terms of players, goals, interaction, and what you have. The game-theoretic approach to meaning in logics is akin to the way natural language users express the proposition *No A is B* by *I'll give you one million dollars if you can find me an A such that B*. Although natural language users do not actually expect the hearer to start looking for an A that is B, research on game-theoretic semantics takes these games seriously as battlefields of "real" players.

Descriptive complexity bridges logic and computation in a very neat way. Conceptually, descriptive complexity departs from the insight that algorithms can verify the truth of an expression from a logical system in a given situation; and conversely, that the set of inputs that are accepted by an algorithm can be described by logical means. In order to illustrate the latter direction, consider all topographical maps (or equivalently, planar graphs) that have the property that one can color its countries with three colors in such a way that adjacent countries are colored differently. If this is the case, we say that the map is 3-colorable. It is easy to come up with a (naive) algorithm that decides for an arbitrary map whether it is 3-colorable. Descriptive complexity allows us to link this algorithm to a logical formula that describes 3-colorable maps. Interestingly, descriptive

complexity also allows to transfer the complexity of the algorithm to the formula.

Combinatorial game theory studies algorithmic approaches to play and analyze games. Well-known research in this field is concerned with the development of Chess computers. Also, combinatorial game theory studies the complexity of games. What the complexity of a game is, is of course determined by the complexity measures employed. A natural complexity measure for Sudoku, for instance, is the number of rules required to solve a puzzle.

It has been proven fruitful to transfer insights and problems from the one discipline to the other. For instance, let us suppose that we want to build an algorithm that can play a certain game. One approach to this problem, enabled by the interaction of the previous disciplines, is to isolate a logic in which all relevant statements about the game can be formalized. For instance, if the game has probabilistic features, one needs to incorporate probabilistic elements in the logical system that suffice to describe the game's features. Once an adequate logical language has been developed, we can use the machinery from the descriptive complexity toolkit, to obtain the desired algorithm.

Another example of the fields' interaction is where we are interested in the impact of a game's property on the game's complexity. Suppose one wishes to investigate whether two-player games are more complex than one-player games, such as Sudoku. One approach to this problem would be to fix several logics. Crucially, one has to see to it that the logics give rise, in the game-theoretic semantics sense, to one and two-player games. Then, we compare the complexity of the logics that give rise to one-player games with the complexity of the logics that give rise to two-player games. Here, the complexity of a logic can be measured by means of the tools from descriptive complexity.

This dissertation is positioned right on the interface of logic, games, and computation. Moreover, its results can be considered contributions to the disciplines of game-theoretic semantics, descriptive complexity, and combinatorial game theory. The theme of this dissertation is imperfect information in logic and games.

Imperfect information. Game theory concerns itself with the strategic interaction of rational agents in utility returning environments. In an early stage of the development of game theory, marked by the publication of (von Neumann and Morgenstern 1944), much attention was paid to situations where information about past moves is only partially available (Kuhn 1953). Games that facilitate strategic interaction in such contexts are called *imperfect information* games. In this thesis, imperfect information games are the principal object of investigation.¹ One of the reasons why games with imperfect information received so much attention is that interactive settings with partial information are omnipresent. Here

¹Games with partial information concerning the characteristics of the other agents, such as preferences, actions, and beliefs, are known as *incomplete information* games. For a definitive treatment of partial information games see (Harsanyi 1967–1968).

also the realm of parlor games comes to mind as one of the many areas in which imperfect information plays a key role. To appreciate the different possible origins of imperfect information in parlor games, I will set out a number of ways in which imperfect information concerning previous moves can be brought about, and mention some games in which they materialize—without aiming to be exhaustive.

- *Through rules:* One-shot games are games in which the players move in parallel, so that they are uninformed about the other players' actions when deciding on their own. So it is the *rules* regulating the behavior of the agents that cause them to be imperfectly informed. Prototypical games of this kind are Rock, Paper, Scissors, first price auctions, and the famous Prisoner's Dilemma, in which the prisoners decide on their testimony in their cell and privately inform the judge.
- *Through attributes:* In a wide range of games the fact that a move is made is commonly known, but the actual specifications of the move are hidden. Well-known examples of such games are Stratego, Kriegspiel, and Scotland Yard. The former two games are imperfect information variants of Chess. Scotland Yard is basically a cops and robbers game on a graph, during which the cops aim to enclose the robber by moving pawns on the board (a graph). The robber makes his moves covertly, jotting them down on a special notepad which is an *attribute* of the game, only now and then revealing his whereabouts to the cops.
- *Through cognitive boundaries:* These latter games would still be interesting if played by supernatural players, that is, players who have an infinite amount of time and paper to make their calculations. By contrast, the game of Memory hinges on the fact that humans have imperfect memory. Another amusing game of this kind is *Ik ga op vakantie en neem mee* (English: I go on holiday and take along), during every round of which a player recites all items that have been announced previously, and then adds another item. If a player cannot recall the contents of the suitcase he is out and loses.

The interface of logic, games, and computation has been extensively explored. Despite this and the fact that imperfect information arises in many natural contexts, structures with imperfect information have not received much attention from the logic and computation community when compared to structures with perfect information. Let me now give a succinct overview of research that studies imperfect information structures in logic, computation, and games.

Independence-friendly logic is an exception within the field of logic, as its semantic evaluation games can be regarded as imperfect information games, see (Sandu 1993; Hintikka 1996; Hintikka and Sandu 1997). The logical analysis of games with imperfect information commenced only recently, formalizing the

interaction between agents, knowledge, action, and preferences in dynamic environments. General frameworks (Fagin, Halpern, Moses, and Vardi 1995; Baltag, Moss, and Solecki 1998) have been proposed, but for some applications they remain undefined or give unsatisfactory accounts. Logical case studies of games with imperfect information have been performed, such as (van Ditmarsch 2000) which gives an in-depth account of Cluedo.

Publications on the computational analysis of games with imperfect information are rather scarce, cf. (van Emde Boas 2003), even on popular games such as Poker, cf. (Billings et al. 2002). *Interactive proof systems*, introduced in (Goldwasser et al. 1989), form an exception in computation theory. Some studies have been performed on the computational cost of imperfect information in games, and they are basically a bad news show. In (Jones 1978) it is shown that once one blindfolds one player, the complexity jumps from \mathbf{P} all the way to \mathbf{PSPACE} . Deciding whether a game-tree allows for a winning strategy for one player becomes intractable (\mathbf{NP} -hard) the very moment this player comes across imperfect information, see (Koller and Megiddo 1992). Furthermore, a Turing machine that has so-called *private states* is capable of recognizing undecidable languages in constant space, see (Reif 1984; Peterson, Azhar, and Reif 2001).

This thesis aims to study structures with imperfect information in the interface of logic, games, and computation.

Questions and motivations. This dissertation evolves around two questions.

Textbook introductions to logic's elementary concepts are usually presented with the help of games. The concept of quantifier and quantifier dependence are paradigmatic cases in point. In fact, from the many clear and intuitive reformulations of logical notions one is tempted to conclude that many logical notions are essentially game-theoretic. But despite the fact that game-theoretic characterizations are omnipresent in logic, we are still waiting for a unifying framework, i.e., a framework in which one can systematically study the consequences that changing a game's property has on the logical notion at hand. No such framework exists however. In this dissertation, modestly, I will not try to set up a general framework. Instead, I will focus on logics whose game-theoretic semantics rely on imperfect information games. Those games aim to be helpful items when setting up such a general framework. Thus, we arrive at the first question:

Question 1: Which games with imperfect information can be defined by logical means, and which reasonable sources can be seen to cause the imperfect information?

The first part of this question is motivated by the sheer absence of logics that rely on games with imperfect information. Although games are frequently used to give an interactive and goal-oriented perspective to logical concepts, still the

games employed in this manner are mostly games with perfect information. Insisting on imperfect information games may thus shed a new light on the interactive and goal-oriented content of logical concepts, and deepen our understanding of their nature.

It is well-conceivable that certain logics define game-theoretic games whose information flows are of a dubious kind, i.e., hard to realize. Therefore I will keep a keen eye on possible explanations of the game's imperfect information. In actual fact, the three aforementioned sources of imperfect information will be used to this effect.

Finally, I hold the view that it is not unlikely that a logical perspective on games with imperfect information can provide insights that are not offered by the current literature on game theory. Thus the focus on logic games may bear relevance to game theory at large. For instance, game-theoretic analyses of environments with imperfect information primarily aim to describe the most profitable strategic behavior, while treating the imperfect information as a given. In my view, a logical analysis may give a more informationally involved account of imperfect information environments than game-theoretic perspectives, explaining the *source* of the imperfect information.

Intuitively, some games are harder than others. We find this intuition confirmed in our daily newspaper in which puzzles are printed with increasing difficulty. Depending on the application and the research question, one has to select one's complexity measures and tools of analysis. Ultimately, interesting questions may be addressed that compare theoretical and cognitive complexity measures of a games. I.e., can computer-oriented complexity measures "predict" a game's cognitive complexity? This and similar grand questions motivate this dissertation's second main question.

Question 2: What are the computational costs of imperfect information in logic and combinatorial game theory?

In order to address Question 2, I will study the computational costs of algorithms that compute certain specific properties of games with imperfect information, and compare them to the computational costs of algorithms that perform the same task for the perfect information counterparts. I will mostly use the tools and notions offered by *complexity theory* to measure the computational costs of algorithms. Complexity theory (Garey and Johnson 1979; Papadimitriou 1994) provides measures of complexity that are well-known to be both mathematically adequate and relevant in everyday practice. In this manner, improving our understanding of the complexity of imperfect information games is not only of interest to computer scientists, but also to game-theorists and logicians working on these games.

The computational costs of imperfect information were explored in general frameworks in the aforementioned (Jones 1978; Reif 1984; Koller and Megiddo

1992; Peterson et al. 2001), but these publications leave Question 2 entirely unanswered. Although they invariably report negative results—in a slogan: “imperfect information increases complexity”—, they cannot tell whether imperfect information has a negative computational impact on *specific* games. For it may well turn out that the computational results reported in the previous publications are due to pathological cases.

Methodology. To let this dissertation carry relevance not only for those interested in games with imperfect information, but also for those interested in logic, games, and computation in general, I selected my topics of investigation from all three disciplines. In particular, the topics addressed in this dissertation are taken from the philosophy of mathematics (Hintikka 1996), generalized quantifier theory (Henkin 1961), the semantics of natural language (Barwise 1979), and the realm of parlor games.

An advantage of working on topics from this range of areas is their high level of interaction. Thus, the objects at stake can be analyzed by means of more or less the same formal machinery. A case in point is the machinery developed in descriptive complexity theory that offers a unifying perspective on logic and computation. Another advantage is that the perfect information variants of the topics considered have been studied rather intensively and so one can meaningfully compare the computational impact of imperfect information.

Let me scan the fields in which the objects of investigation are situated:

- Sandu and Hintikka (Sandu 1993; Hintikka 1996; Hintikka and Sandu 1997) show that Independence-friendly logic (abbreviated “IF logic”) can be associated with semantic evaluation games with imperfect information. Readers familiar with semantic evaluation games for first-order logic will acknowledge that those have perfect information. From a game-theoretic perspective, IF logic can be seen to loosen this assumption by allowing for hiding parts of past moves.
- From a logical viewpoint, the idea underlying IF logic generalizes ideas from *partially ordered quantification theory*. Henkin (1961) introduced partially ordered quantifiers more or less as a mathematical exercise. The ideas in (Henkin 1961) gave birth to an extensive model-theoretic theory on the topic with extensions in many directions. Gottlob et al. (1995, pg. 67) describe partially ordered quantifiers as “important in both model theory and theoretical linguistics.”
- In the theory of natural language semantics, Hintikka (1974) and Barwise (1979) argued, amongst others, that some natural language sentences cannot be accounted for by “traditional” logical means. They argue that the formal apparatus of *branching quantifiers* should be included in the semanticist’s toolbox to give certain sentences their correct logical form.

- Although the existence of games with imperfect information is acknowledged in one of the paradigmatic publications in combinatorial game theory (Berlekamp et al. 1982, pg. 16-7), they have not been studied intensively. This state of affairs is somewhat unjust, because imperfect information makes many games tick, even some very popular ones. Combinatorial game theory studies the algorithmic aspects of games, i.e., ways of mechanically computing properties of games. The header combinatorial game theory may be somewhat confusing, in that it seems to imply that the field is a branch of game theory, whereas in actual fact it lies right on the interface of computation and game theory.²

Structure. In Chapter 2, I collect preliminary definitions and seminal results from logic, complexity theory, and game theory. Even a superficial glance at the structure of this chapter shows the cohesion of the fields. The reader may use this chapter as a reference throughout this dissertation.

In Chapter 3, I study two fragments of Independence-friendly logic which are motivated from a game-theoretic and computational perspective, respectively. IF logic’s semantics can be studied through imperfect information games and I show that the imperfect information traditionally associated with IF logic can be explained by attributes such as envelopes. In the interest of Question 1, I import the received game-theoretic notion of *perfect recall* into the IF framework and study the impact it has on the complexity of the system. The result of this enterprise is that the restrictions perfect recall imposes on imperfect information games defined by IF logic decrease the complexity, serving Question 2. Also, I study which imperfect information games are described by independence-friendly modal logics, hooking up with current research on this topic.

In Chapter 4, I take up the study of *partially ordered connectives* as defined in (Blass and Gurevich 1986; Sandu and Väänänen 1992), that can be seen as variants of Henkin’s (1961) partially ordered quantifiers. As I pointed out before, Henkin quantifiers are precursors of IF logic. I will show that, just like IF logic, their semantics can be given in terms of semantic games with imperfect information. However, unlike IF games, the imperfect information in games for logics with partially ordered quantifiers and connectives can be explained by cognitive bounds. I will further show that modifying the game-theoretic parameter of *absentmindedness* in this framework gives rise to generalized quantifiers which were studied independently. The lion’s share of this chapter is devoted to the descriptive complexity of logics with partially ordered connectives in line with

²For those familiar with branching notation: A more perspicuous name for combinatorial game theory would thus be

$$\left(\begin{array}{c} \text{combinatorics} \\ \text{game theory} \end{array} \right)$$

indicating that game theory does not have combinatorics in its scope, nor vice versa.

Question 2.

In Chapter 5, the principal object of investigation are branching quantifiers, as one finds them in theoretical linguistics and quantification theory. In the interest of Question 1, I give a game-theoretic interpretation of these quantifiers in the framework of *strategic game theory*. Several researchers (Blass and Gurevich 1986; van Benthem 2004; Ajtai 2005) have suggested the interesting mathematical structure this framework has in reserve for logic theory, but to the best of my knowledge the presented results are the first in this respect. The analysis shows that the imperfect information in games for branching quantifiers can be explained by an appeal to rules, just as the Prisoner's dilemma. In view of Question 2, I develop a theory of the computational complexity of natural language quantifiers, in order to compare their complexity to that of branching quantifiers. It turns out that branching quantifiers are intractable (**NP**-hard), whereas the previously studied natural language quantifiers are computationally much more well-behaved.

In Chapter 6, pursuing an answer to Question 2 will be the focus of attention as I analyze the parlor game of Scotland Yard. This game is commonly known and has amused game players for over two decades. The imperfect information in the game is introduced by a special attribute—a *move board*. The important thing is that Scotland Yard features a natural form of imperfect information that can be formalized and generalized in a straightforward manner. The formalization suits the aims of this dissertation perfectly to the point that it also allows for analyses of Scotland Yard as a perfect information game. Quite surprisingly, it will be shown that the imperfect information does *not* increase complexity.

Chapter 7 concludes the dissertation.

A bibliography, an index, a list of symbols, and a summary (in Dutch) are found at the back of this volume.

Origin of the material. The thinking in parts of Chapter 3 has influenced (Tulenheimo and Sevenster 2006). Chapter 4 is based on the joint paper (Sevenster and Tulenheimo 2006) and on (Sevenster 2006b). An extension of (Sevenster and Tulenheimo 2006) will be submitted for publication to the *Journal of Logic and Computation*. Chapter 6 is based on the research report (Sevenster 2006a). The material from the other chapters has not been published.

Chapter 2

Prerequisites

This chapter introduces notation and basic terminology from the addressed disciplines. The order in which the material is presented is largely determined by its mutual dependence—it is not to be taken as a ranking of importance.

2.1 Basic notation

Let X and Y be sets. If X contains no objects, it is the empty set \emptyset . $\|X\|$ denotes the *cardinality* of X and $\wp(X)$ denotes the *power set* of X . The operations \cup , \cap , \subseteq , and $-$ on sets are defined as usual. The set $X \times Y$ denotes the *cartesian product* of X and Y : $\{\langle x, y \rangle \mid x \in X, y \in Y\}$. If k is an integer, X^k denotes

$$\underbrace{X \times \dots \times X}_k.$$

Y^X denotes the space of functions of type $X \rightarrow Y$.

Let $f : X \rightarrow Y$ be a function. Then, “considering f as a set” means to regard it as the mathematical object $\{\langle x, f(x) \rangle \mid x \in X\}$ to which any set-theoretic operation applies. $X = \text{dom}(f)$ is called f 's *domain*, whereas $Y = \text{rng}(f)$ is called f 's *range*.

Let the set of natural numbers be denoted by \mathbb{N} , and the set of reals by \mathbb{R} . By postulation, 0 is in neither.

Let f and g be functions from \mathbb{N} to \mathbb{N} . Then, I say f is of the order of g , if there are integers c and n_0 , such that for all $n \geq n_0$, $f(n) \leq c \cdot g(n)$. If f is of the order of g I may also write $f(n) = \mathcal{O}(g(n))$.

Strings of similar objects a_1, \dots, a_n are abbreviated by \vec{a} , but this will be highlighted every time confusions threatens.

2.2 Game theory

In terminology and notation I stay close to (Osborne and Rubinstein 1994). Crucial game-theoretic concept in the present dissertation is *extensive game with imperfect information*. One may consider this an extension of the notion of *extensive game with perfect information*, due to von Neumann and Morgenstern (1944).

Extensive games with perfect information. An extensive game with perfect information G is a tuple $\langle N, H, P, \langle U_i \rangle_{i \in N} \rangle$, where

- N is the set of *players*. Referring to the number of players, G is called an $\|N\|$ -*player game*.
- H is the set of *histories* that satisfies the following two conditions:
 - The empty sequence ε is a history in H , called the *initial history*.
 - If $hh' = \langle a_1, \dots, a_m, a_{m+1}, \dots, a_n \rangle$ is a history in H , then the string $h = \langle a_1, \dots, a_m \rangle$ is a history in H —called an *initial prefix* of hh' .

In case h is an initial prefix of hh' and h' is non-empty, then h is called a *proper initial prefix* of hh' . If $ha = \langle a_1, \dots, a_m, a \rangle$ is a history in H and a is a single component, then a is called an *action* that extends h . Further, ha is called an *immediate successor* of h . $A(h)$ denotes all actions that extend h . Let Z be the subset of H whose histories cannot be extended. Z is called the *set of terminal histories*.

If $h = \langle a_1, \dots, a_n \rangle$, then $\ell(h) = n$ denotes the *length* of h . Throughout this dissertation all histories are finite. Define $\ell(\varepsilon) = 0$.

- P is the *player function*, that assigns to every non-terminal history h a player $P(h)$. Formally, P is a function of type $(H - Z) \rightarrow N$. I say that a history h *belongs to* $P(h)$.

G is called *finite*, if H is finite. G is said to be of *finite horizon* if every history in H has a finite length $\ell(h)$. All games in this dissertation have finite horizon.

- U_i is player i 's *utility function*. In general, one may regard it as a function from the set of terminal histories Z into the reals. However, in the majority of this dissertation's applications the range of the utility functions will be restricted to $\{-1, 1\}$. Games with utility functions of the above kind are called *win-loss games*. Intuitively, $U_i(h) = 1$ indicates that player i has won in history h . I shall also write *win* and *lose* for 1 and -1 , respectively. Call G a *zero-sum game* if for every history h in G , $\sum_{i \in N} U_i(h) = 0$.

Let G be an extensive game with perfect information as above, that is furthermore two-player and win-loss. Then, a function S is called a *strategy* for player $i \in N$ in G , if it maps every history h belonging to i , to an action in $A(h)$. Let S be a strategy for player i in G . Then, call a history h in *accordance* with S , if for every proper initial prefix $h' = \langle a_1, \dots, a_n \rangle$ of h , such that $P(h') = i$, it is the case that $\langle a_1, \dots, a_n, S(h') \rangle$ is also an initial prefix of h . Further still, if for every history h that is in accordance with S it is the case that $U_i(h) = \text{win}$, then S is called a *winning strategy* for i in G . G is called *determined*, if any of its players has a winning strategy in G .

The following seminal result is often referred to as the *Gale-Stewart Theorem*:

2.2.1. THEOREM (GALE AND STEWART (1953)). *Let G be a two-player, zero-sum, extensive game with perfect information, that is of finite horizon. Then, G is determined.*

In some respects, a strategy S is a baroque object, as it may assign an action to histories that themselves are not in accordance with S . This observation leads to the following definition. Let G be an extensive game with perfect information and let S be a strategy in G for player i . For now, consider S as a set. Then, a function $T \subseteq S$ is called a *plan of action* for player i in G (based on S), if its range is the set of histories from H that are in accordance with S . For a discussion about strategies and plans of action see (Osborne and Rubinstein 1994, pg. 103-4).

The notion of winning strategy is naturally transferred to plans of actions. In fact, when it comes to winning, the two notions are interchangeable: Let T be a plan of action for player i in G based on S . Then, T is a winning plan of action in G iff S is a winning strategy in G .

Let G be a two-player, finite, extensive game with perfect information, that is win-loss. The *backward induction algorithm*, due to Zermelo (1913), decides whether player i has a winning plan of action. The algorithm takes G as input and goes about as follows:

- Label all terminal histories h in G with $U_i(h)$.
- Until the initial history ε has been labeled, consider every unlabeled, non-terminal history h in G and do as follows:
 - If $P(h) = i$ and there is an immediate successor history h' of h labeled *win*, then label h with *win*.
 - If $P(h) \neq i$ and every immediate successor history h' of h is labeled *lose*, then label h with *lose*.

An easy inductive argument shows that the backwards induction algorithm labels ε with *win* iff player i has a winning plan of action in G .

Extensive games with imperfect information. Extensive games with imperfect information extend extensive games with perfect information in that they are five-tuples $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, \langle U_i \rangle_{i \in N} \rangle$, carrying *information sets* \mathcal{I}_i for every player $i \in N$. The other notions are similar to the ones defined for extensive games with perfect information. An information set $\mathcal{I}_i = \{I_1, \dots, I_n\}$ is a partition of the set of histories belonging to player i , that meets the *action consistency requirement*: if h and h' sit in $I \in \mathcal{I}_i$, then $A(h) = A(h')$. Every partition in an information set is called an *information partition*. If $I \in \mathcal{I}_i$, player i is said to *own* I and \mathcal{I}_i , or they are said to *belong* to i . Intuitively, an extensive game with imperfect information models the situation in which player i knows that some history $h \in I \in \mathcal{I}_i$ has happened, but she is unable to tell h apart from the other histories in I . The requirement that all histories in an information partition can be extended by the same actions—the action consistency requirement—captures the idea that otherwise the player owning the information partition could deduce information about the actual history from the actions available. If $I \in \mathcal{I}_i$, write $A(I)$ to denote $A(h)$, for an arbitrary $h \in I$.

Let $G = \langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, \langle U_i \rangle_{i \in N} \rangle$ be an extensive game with imperfect information. Then, a function S is called a *strategy* for player i in G , if it maps every information partition $I \in \mathcal{I}_i$ belonging to i onto an action in $A(I)$.

In the context of win-loss games, a strategy S for player i is called *winning in G* if every terminal history h that is in accordance with S yields $U_i(h) = \text{win}$. The notion of determinedness of G is inherited analogously. But a Gale-Stewart Theorem cannot be proved for games with imperfect information. A case in point is the game of Rock, paper, scissors in which neither player has a winning strategy. This game is thus called *undetermined*.

G is said to have the *von Neumann-Morgenstern property* if for every information partition in G it is the case that all of its histories have the same length. Games that violate the von Neumann-Morgenstern property often involve *absent-minded* agents.

2.3 Logic

Syntax. Let $\text{VAR} = \{x, y, z, \dots\}$ be the countably infinite set of *variables*. Let $\text{IND} = \{i, j, \dots\}$ be the countably infinite set of *indices*. Let $\text{R-VAR}_n = \{X, Y, \dots\}$ be the countably infinite set of *relation variables* with arity n and let $\text{R-VAR} = \bigcup_n \text{R-VAR}_n$. To stress that an object from VAR is not a relation variable, I may call it a *first-order variable*. Let $\text{F-VAR}_n = \{f, g, \dots\}$ be the countably infinite set of *function variables* with arity n and let $\text{F-VAR} = \bigcup_n \text{F-VAR}_n$.

A *vocabulary* τ is a finite set of *relation symbols* P, R, \dots , that rigidly contains the equality symbol $=$. In fact, if I want to specify a vocabulary by its contents, I will omit $=$. Thus if the vocabulary τ does not contain relation symbols other than the equality symbol, I write \emptyset to refer to τ .

It will be convenient to assume that with every vocabulary τ there is a set of *tokens* associated with it, such that there is a bijection R that maps every $t \in \text{Token}(\tau)$ onto the relation symbol $R_t \in \tau$.

Relation symbols come with a natural number, their *arity*. If R is a unary relation symbol, R is also called a *predicate symbol*. The relation symbol $=$ is binary.

The *formulae of second-order logic in the vocabulary τ* , denoted $\mathbf{SO}(\tau)$, are the strings generated by applying the following formation rules a finite number of times:

- (T1) All first-order variables are terms.
- (T2) If f is an n -ary function variable and t_1, \dots, t_n are terms, then $f(t_1, \dots, t_n)$ is a term.
- (F1) If R is an n -ary relation symbol in τ and t_1, \dots, t_n are terms, then the string $R(t_1, \dots, t_n)$ is a formula.
- (S1) If $X \in \text{R-VAR}_n$ and t_1, \dots, t_n are terms, then $X(t_1, \dots, t_n)$ is a formula.
- (F2) If Φ is a formula, then $\neg\Phi$ is a formula.
- (F3) If Φ and Ψ are formulae, then $\Phi \vee \Psi$ is a formula.
- (F4) If Φ is a formula and $x \in \text{VAR}$, then $\exists x \Phi$ is a formula.
- (S2) If Φ is a formula and $X \in \text{R-VAR}$, then $\exists X \Phi$ is a formula.
- (S3) If Φ is a formula and $g \in \text{F-VAR}$, then $\exists g \Phi$ is a formula.

The *formulae of first-order logic in the vocabulary τ* , denoted $\mathbf{FO}(\tau)$, are generated by applying the rules (T1) and (F1)-(F4) a finite number of times.

In Chapter 3 an extension of first-order logic, denoted $\mathbf{FOV}(\tau)$, is considered whose strings are generated by applying a finite number of times the rules (T1), (F1)-(F4) plus (F5) and (F6):

- (F5) If t_1, \dots, t_n are terms and $i \in \text{IND}$, then $R_i(t_1, \dots, t_n)$ is a formula.
- (F6) If Φ is a formula, $i \in \text{IND}$, and I is a subset of $\text{Token}(\tau)$, then $\bigvee_{i \in I} \Phi$ is a formula.

If Φ is a second-order formula in the vocabulary τ , then it is also called a $\mathbf{SO}(\tau)$ -*formula*. Φ is called a \mathbf{SO} -*formula*, if it is a $\mathbf{SO}(\tau)$ -*formula* for some vocabulary τ . This convention pertains to all logics discussed in this dissertation.

All formulae produced by (F1), (S1), and (F5) are called *atoms*.

Throughout this dissertation I will use the following shorthand notation:

$$\begin{aligned}\Phi \wedge \Psi & \text{ for } \neg(\neg\Phi \vee \neg\Psi) \\ \Phi \rightarrow \Psi & \text{ for } \neg\Phi \vee \Psi \\ \forall x \Phi & \text{ for } \neg\exists x\neg\Phi \\ \bigwedge_{i \in I} \Phi & \text{ for } \neg\bigvee_{i \in I} \neg\Phi.\end{aligned}$$

The objects $\exists x$ and $\forall x$ are *first-order quantifiers*; $\exists X$ and $\forall X$ are *second-order quantifiers*; $\bigvee_{i \in I}$ and $\bigwedge_{i \in I}$ are *restricted quantifiers*. If $\exists \dots$ ($\forall \dots$) is a quantifier, it is an *existential* (*universal*) *quantifier*; $\bigvee_{i \in I}$ is a *disjunctive quantifier* and $\bigwedge_{i \in I}$ is a *conjunctive quantifier*.

Throughout this dissertation I will write capital letters Φ, Ψ, \dots to denote second-order formulae and lowercase letters ϕ, ψ, \dots to denote first-order formulae. It should be borne in mind that the definitions for second-order logic introduced shortly pertain to **FO**, since it is a syntactical fragment. I take it for granted that the reader can transfer all terminology to **FOV**.

Let $\Sigma_n^1(\tau)$ denote the set of **SO**(τ)-formulae of the form

$$\exists X_1 \dots Q_m X_m \phi$$

whose string of second-order quantifiers $\exists X_1 \dots Q_m$ consists of n consecutive blocks, where in each block all quantifiers are of the same type and adjacent blocks contain quantifiers of different type. The set $\Pi_n^1(\tau)$ is defined analogously, but here the first block is universal. The language Σ_1^1 will be referred to as *existential, second-order logic*.

The set $Free(\Phi)$ of *free variables* in a second-order formula Φ is defined by:

$$\begin{aligned}Free(\Phi) & = \text{the set of all variables in } \Phi, \quad \text{for atomic } \Phi \\ Free(\neg\Phi) & = Free(\Phi) \\ Free(\Phi \vee \Psi) & = Free(\Phi) \cup Free(\Psi) \\ Free(\exists x \Phi) & = Free(\Phi) - \{x\} \\ Free(\exists X \Phi) & = Free(\Phi) - \{X\} \\ Free(\exists f \Phi) & = Free(\Phi) - \{f\}.\end{aligned}$$

In order to indicate that the variables x_1, \dots, x_n are free in Φ , I may write $\Phi(x_1, \dots, x_n)$. If $Free(\Phi) = \emptyset$, then Φ is called a *sentence*.

The set $Sub(\Phi)$ of *subformulae* of a second-order formula Φ is defined by:

$$\begin{aligned}Sub(\Phi) & = \{\Phi\}, \quad \text{for atomic } \Phi \\ Sub(\neg\Phi) & = \{\neg\Phi\} \cup Sub(\Phi) \\ Sub(\Phi \vee \Psi) & = \{\Phi \vee \Psi\} \cup Sub(\Phi) \cup Sub(\Psi) \\ Sub(\exists x \Phi) & = \{\exists x \Phi\} \cup Sub(\Phi) \\ Sub(\exists X \Phi) & = \{\exists X \Phi\} \cup Sub(\Phi) \\ Sub(\exists f \Phi) & = \{\exists f \Phi\} \cup Sub(\Phi).\end{aligned}$$

A second-order formula Φ is in *negation normal form*, if for every $\neg\Psi \in \text{Sub}(\Phi)$ it is the case that Ψ is an atom.

Semantics. Let $\tau = \{R_1, \dots, R_k\}$ be a vocabulary. Then, a τ -structure \mathfrak{A} is a tuple

$$\langle A, R_1^{\mathfrak{A}}, \dots, R_k^{\mathfrak{A}} \rangle$$

where A is a non-empty set, called \mathfrak{A} 's *universe*, and $R_i^{\mathfrak{A}} \subseteq A^{a_i}$, for each a_i -ary relation symbol R_i in τ . The set $R_i^{\mathfrak{A}}$ is called the *interpretation* of R_i on \mathfrak{A} . On every structure \mathfrak{A} , the equality symbol $=$ is interpreted as the identity relation on A : $\{\langle a, a \rangle \mid a \in A\}$.

If the relation (truth value) between a formula Φ and a structure \mathfrak{A} is at stake, then by stating that \mathfrak{A} is *suitable* I ensure that \mathfrak{A} interprets all of Φ 's relation symbols.

Let \mathfrak{A} be a τ -structure. Consider maps of the following types:

$$\text{VAR} \quad \text{to} \quad A \tag{2.1}$$

$$\text{R-VAR}_n \quad \text{to} \quad \wp(A^n) \tag{2.2}$$

$$\text{F-VAR}_n \quad \text{to} \quad A^{(A^n)} \tag{2.3}$$

$$\text{IND} \quad \text{to} \quad \text{Token}(\tau). \tag{2.4}$$

A function α is a **SO**(τ) *assignment in* \mathfrak{A} if it is a many-sorted function of type (2.1)-(2.3); it is a **FO**(τ) *assignment in* \mathfrak{A} if it is a function of type (2.1); and it is a **FOV**(τ) *assignment in* \mathfrak{A} if it is a many-sorted function of type (2.1) and (2.4). The nature of an assignment will be clear from the context, for which reason I mostly omit to mention what kind of assignment it actually is.

Let α be an assignment in \mathfrak{A} , let $x \in \text{VAR}$, and let a be an object in the universe of \mathfrak{A} . Then, $[\alpha.x/a]$ denotes the assignment in \mathfrak{A} that agrees with α on every variable, except for the variable x to which it assigns the object a . Changes in α with respect to variables from R-VAR and F-VAR and indices from IND are defined analogously.

If only the object a that is assigned by α to x is of interest, I may write $[x/a]$ rather than α . If the object a at stake is immaterial, I may even write $[x^{\mathfrak{A}}]$, to indicate that in this assignment some object $x^{\mathfrak{A}}$ from A is assigned to x .

Let τ be a vocabulary, let \mathfrak{A} be a τ -structure, and let α be an assignment in \mathfrak{A} . Define the *satisfaction relation* for second-order logic of vocabulary τ , as

follows:

$$\begin{aligned}
\mathfrak{A} \models (t_1 = t_2)[\alpha] & \text{ iff } \alpha(t_1) = \alpha(t_2) \\
\mathfrak{A} \models R(t_1, \dots, t_n)[\alpha] & \text{ iff } \langle \alpha(t_1), \dots, \alpha(t_n) \rangle \in R^{\mathfrak{A}} \\
\mathfrak{A} \models X(t_1, \dots, t_n)[\alpha] & \text{ iff } \langle \alpha(t_1), \dots, \alpha(t_n) \rangle \in \alpha(X) \\
\mathfrak{A} \models \neg\Phi[\alpha] & \text{ iff not } \mathfrak{A} \models \Phi \\
\mathfrak{A} \models (\Phi \vee \Psi)[\alpha] & \text{ iff } \mathfrak{A} \models \Phi[\alpha] \text{ or } \mathfrak{A} \models \Psi[\alpha] \\
\mathfrak{A} \models (\exists x \Phi)[\alpha] & \text{ iff } \mathfrak{A} \models \Phi[\alpha.x/a], \text{ for some } a \in A, x \in \text{VAR} \\
\mathfrak{A} \models (\exists X \Phi)[\alpha] & \text{ iff } \mathfrak{A} \models \Phi[\alpha.X/B], \text{ for some } B \subseteq A^n, X \in \text{R-VAR}_n \\
\mathfrak{A} \models (\exists f \Phi)[\alpha] & \text{ iff } \mathfrak{A} \models \Phi[\alpha.f/g], \text{ for some } g \in A^{A^n}, f \in \text{F-VAR}_n,
\end{aligned}$$

where $\alpha(f(t_1, \dots, t_n))$ is inductively defined as $\alpha(f)(\alpha(t_1), \dots, \alpha(t_n))$.

Extend the satisfaction relation for second-order logic to apply to $\mathbf{FOV}(\tau)$, in the following manner:

$$\begin{aligned}
\mathfrak{A} \models R_i(t_1, \dots, t_n)[\alpha] & \text{ iff } \langle \alpha(t_1), \dots, \alpha(t_n) \rangle \in R_{\alpha(i)}^{\mathfrak{A}} \\
\mathfrak{A} \models \bigvee_{i \in I} \Phi[\alpha] & \text{ iff } \mathfrak{A} \models \Phi[\alpha.i/t], \text{ for some } t \in \text{Token}(\tau).
\end{aligned}$$

2.3.1. EXAMPLE. The formal treatment of restricted quantifiers is somewhat unusual, for which reason I give a small example. To this end, consider the vocabulary $\tau = \{R_a, R_b\}$, for which $\text{Token}(\tau) = \{a, b\}$.

Consider the \mathbf{FOV} -formula

$$\phi(x_1) = \bigwedge_{i \in \{a, b\}} \exists x_2 R_i(x_1, x_2).$$

Suppose that for some τ -structure \mathfrak{A} and assignment α it is the case that

$$\mathfrak{A} \models \phi(x_1)[\alpha]. \tag{2.5}$$

Spelling out the truth definition yields that (2.5) iff for every $t \in \{a, b\}$

$$\mathfrak{A} \models \exists x_2 R_i(x_1, x_2)[\alpha.i/t],$$

which is equivalent to it being the case that for every $t \in \{a, b\}$ there exists an $c \in A$, such that $\langle \alpha(x_1), c \rangle \in R_t^{\mathfrak{A}}$. More colloquially, $\mathfrak{A} \models \phi(x_1)[\alpha]$ says that $\alpha(x_1)$ has an R_a successor and an R_b successor: $\mathfrak{A} \models (\exists x_2 R_a(x_1, x_2)) \wedge (\exists x_2 R_b(x_1, x_2))$. \square

If $\mathfrak{A} \models \Phi[\alpha]$, I say that Φ is *true* on \mathfrak{A} under α . If $\mathfrak{A} \not\models \Phi[\alpha]$, I say that Φ is *false* on \mathfrak{A} under α . When it comes to the truth or falsity of a sentence on a structure under an assignment, the assignment is immaterial and shall be omitted.

Let \mathbf{L} and \mathbf{L}' be logical languages for which the satisfaction relation \models is properly defined with respect to models and assignments, as well as the notion of formula. Let Φ be an \mathbf{L} -sentence and let Ψ be an \mathbf{L}' -sentence.

Then, Φ and Ψ are *equivalent*, if for every suitable structure \mathfrak{A} and assignment α in \mathfrak{A} , it is the case that

$$\mathfrak{A} \models \Phi[\alpha] \quad \text{iff} \quad \mathfrak{A} \models \Psi[\alpha].$$

It is well-known that every first-order formula has an equivalent formula in negation normal form.

I write $\mathbf{L} \leq \mathbf{L}'$ to indicate that for every \mathbf{L} -formula Φ , there is an \mathbf{L}' -formula Ψ in the same vocabulary such that Φ and Ψ are equivalent. If $\mathbf{L} \leq \mathbf{L}'$, I say that \mathbf{L} is *translatable* into \mathbf{L}' . Further, $\mathbf{L} = \mathbf{L}'$ is shorthand for $\mathbf{L} \leq \mathbf{L}'$ and $\mathbf{L} \geq \mathbf{L}'$; and $\mathbf{L} < \mathbf{L}'$ abbreviates $\mathbf{L} \leq \mathbf{L}'$ but not $\mathbf{L} = \mathbf{L}'$.

Game-theoretic semantics for FO. While evaluating the truth of a first-order formula ϕ on \mathfrak{A} under α one often finds oneself imagining playing a game against an opponent with opposite ends. In this game the turn taking is regulated by the logical constants in ϕ and the winning conditions are set by \mathfrak{A} . The common practice of regarding verification as game playing has been given a formal underpinning using tools from game theory, yielding so-called *semantic evaluation games*.

Let ϕ be a first-order τ -formula in negation normal form, let \mathfrak{A} be a τ -structure, and let α be an assignment in \mathfrak{A} . Then, the *semantic evaluation game of ϕ on \mathfrak{A} under α* is a game between two players, Abelard and Eloise, starting from the position $\langle \phi[\alpha], \mathfrak{A} \rangle$ governed by the following rules:

- In $\langle (\phi_0 \vee \phi_1)[\alpha], \mathfrak{A} \rangle$ Eloise chooses $i \in \{0, 1\}$; the game proceeds as $\langle \phi_i[\alpha], \mathfrak{A} \rangle$.
- In $\langle (\phi_0 \wedge \phi_1)[\alpha], \mathfrak{A} \rangle$ Abelard chooses $i \in \{0, 1\}$; the game proceeds as $\langle \phi_i[\alpha], \mathfrak{A} \rangle$.
- In $\langle \exists x \phi[\alpha], \mathfrak{A} \rangle$ Eloise chooses $a \in A$; the game proceeds as $\langle \phi[\alpha.x/a], \mathfrak{A} \rangle$.
- In $\langle \forall x \phi[\alpha], \mathfrak{A} \rangle$ Abelard chooses $a \in A$; the game proceeds as $\langle \phi[\alpha.x/a], \mathfrak{A} \rangle$.
- $\langle R(x_1, \dots, x_n)[\alpha], \mathfrak{A} \rangle$ marks the end of the game. Eloise wins if the tuple $\langle \alpha(x_1), \dots, \alpha(x_n) \rangle$ sits in $R^{\mathfrak{A}}$; otherwise, Abelard wins.
- $\langle \neg R(x_1, \dots, x_n)[\alpha], \mathfrak{A} \rangle$ is similar to the previous rule, with the winning conditions swapped.

These game rules give rise to the game-theoretic object of *extensive game of perfect information Sem-game^{FO}*($\phi[\alpha], \mathfrak{A}$). I omit a detailed specification but it is readily obtained from the extensive game for Independence-friendly logic, defined

in Definition 3.2.7 of Section 3.2. In the game-theoretic framework one can set up a full-blown semantics for first-order logic, by putting ϕ *true under game-theoretic semantics on \mathfrak{A} under α* iff Eloise has a winning strategy in $Sem\text{-}game^{\mathbf{FO}}(\phi[\alpha], \mathfrak{A})$. An easy inductive argument shows that game-theoretic semantics is just a reformulation of the satisfaction relation \models laid out above.

Henceforth, when I speak of the semantic games of a first-order formula ϕ , I refer to the class of all extensive games for ϕ . Likewise, the semantic games for first-order logic refers to the union of the semantic games of all first-order formulae. This convention pertains to all logics discussed in this dissertation.

2.4 Computational complexity

The definitions are mostly adopted from (Papadimitriou 1994), that has been a source of inspiration throughout my studies.

Turing machines and complexity classes. The basic device of computation in this dissertation will be the *Turing machine*, introduced in (Turing 1936). Most of the particulars of Turing machines are not of direct interest to the current thesis, for which reason I omit them; but do see (van Emde Boas 1990).

Let Σ be an alphabet, that is, a finite set of letters. Let $L \subseteq \Sigma^*$ be a set of finite strings in Σ . Call L a *language* or a (*decision*) *problem*. A string $x \in \Sigma^*$ is called an *instance* of L . Say that a deterministic Turing machine M *decides* L , if for every instance x of L , the computation path of M on x halts on the accepting state if $x \in L$; and halts in the rejecting state if $x \notin L$. Let $\mathbf{TIME}(f(n))$ be the class of languages L for which there exists a deterministic Turing machine that decides L in at most $f(n)$ time steps, that is, for every instance x of L , the computation path of M on x is shorter than $f(n)$, where $n = \|x\|$ is the length of x . The object $\mathbf{TIME}(f(n))$ is called a *complexity class*. A non-deterministic Turing machine M *decides* L , if for every instance x of L there is a branch in the computation tree of M on x that ends in the accepting state if $x \in L$; and there is no such branch, if $x \notin L$. Let $\mathbf{NTIME}(f(n))$ be the class of languages L for which there exists a non-deterministic Turing machine M that decides L in $f(n)$, that is, for every instance x of L all branches in the computation tree of M on x are shorter than $f(n)$, where $n = \|x\|$.

Space bounded complexity classes and their respective non-deterministic counterparts are defined analogously.

The complexity class most prominent in this thesis are listed below:

$$\begin{aligned}
\mathbf{L} &= \bigcup_{c \in \mathbb{N}} \mathbf{SPACE}(c \log n) \\
\mathbf{NL} &= \bigcup_{c \in \mathbb{N}} \mathbf{NSPACE}(c \log n) \\
\mathbf{P} &= \bigcup_{c \in \mathbb{N}} \mathbf{TIME}(n^c) \\
\mathbf{NP} &= \bigcup_{c \in \mathbb{N}} \mathbf{NTIME}(n^c) \\
\mathbf{PSPACE} &= \bigcup_{c \in \mathbb{N}} \mathbf{SPACE}(n^c) \\
\mathbf{NPSPACE} &= \bigcup_{c \in \mathbb{N}} \mathbf{NSPACE}(n^c) \\
\mathbf{EXPTIME} &= \bigcup_{c \in \mathbb{N}} \mathbf{TIME}(c^n) \\
\mathbf{NEXPTIME} &= \bigcup_{c \in \mathbb{N}} \mathbf{NTIME}(c^n).
\end{aligned}$$

If $L \in \mathbf{NP}$, one typically says that L is *decidable (solvable, computable) in non-deterministic polynomial time* and likewise for the other complexity classes.

For any pair of complexity classes presented above, it is fairly straightforward to show that the lower one includes the upper one. However, only little is known about the strictness of these inclusions. For one thing, it is known by a brute force argument that \mathbf{L} is strictly included in \mathbf{PSPACE} , but the inclusions in between those complexity classes are unknown to be strict. Whether \mathbf{P} is strictly included in \mathbf{NP} is the one million dollar $\mathbf{P} = \mathbf{NP}$ question. The importance of this question is hard to overestimate, since the problems in \mathbf{NP} are usually taken to be *intractable* or *not efficiently solvable* problems, cf. (Garey and Johnson 1979). By contrast, most problems in \mathbf{P} are conceived of as *tractable* or *efficiently solvable*.

It is known that every language decidable in polynomial space on a non-deterministic Turing machine, is also decided by a polynomial space, deterministic Turing machine.

2.4.1. THEOREM (SAVITCH (1970)). $\mathbf{PSPACE} = \mathbf{NPSPACE}$.

Let $L \subseteq \Sigma^*$ be a language. Then, $\bar{L} = \Sigma^* - L$ denotes the *complement of L* . Let \mathbf{C} be a complexity class, then \mathbf{coC} denotes the class $\{\bar{L} \mid L \in \mathbf{C}\}$, which is called \mathbf{C} 's *complement*. It is readily observed that every deterministic complexity class is equivalent to its complement. As regards non-deterministic classes, the following result is known.

2.4.2. THEOREM (SZELEPCSÉNYI (1987) AND IMMERMANN (1988)).
 $\mathbf{NL} = \mathbf{coNL}$.

It is unknown whether $\mathbf{NP} = \mathbf{coNP}$.

Reductions and complete problems. The idea of one problem being harder than another is formalized by the notion of *reduction*. The kind of reduction I use is also known as *many-one* or *Karp* reduction. Let L and L' be two problems and let \mathbf{C} be a complexity. Then, L is \mathbf{C} -reducible to L' , if there is a \mathbf{C} -computable function R from strings to strings, such that for all instances x of L the following holds: $x \in L$ iff $R(x) \in L'$. R is called a \mathbf{C} -computable reduction from L to L' . Throughout this dissertation, I will only use \mathbf{P} -computable reductions, for which reason I will omit to mention \mathbf{C} . One may insist on weaker reductions. For instance, Papadimitriou (1994, Definition 8.1) uses \mathbf{L} -computable reductions. For the purposes of this thesis the weaker notions of reductions are not of great interest for which reason I did not try to strengthen my results in this respect.

Let \mathbf{C} be a complexity class and let L be a language. Then, L is called *complete* for \mathbf{C} , if $L \in \mathbf{C}$ and every language $L' \in \mathbf{C}$ is reducible to L . If L is complete for \mathbf{C} (or \mathbf{C} -complete), L can be thought of as a prototypical language of \mathbf{C} in that it is amongst the hardest languages in \mathbf{C} . Fascinatingly, many languages bearing real-life interest turn to be complete for one complexity class or another.

The first problem shown complete for \mathbf{NP} , was the satisfiability problem of propositional logic. Formally, the problem SAT contains all propositional formulae that are satisfiable, that is, for which there exists a truth assignment over its proposition letters that renders the formula true. A variant of SAT will be used in Chapter 6.

2.4.3. THEOREM (COOK (1971)). SAT is \mathbf{NP} -complete.

2.5 Descriptive complexity

Finite model theory. Let τ be a vocabulary as before. A *property* Π over a class \mathcal{K} of τ -structures is a function assigning a truth value $\Pi(\mathfrak{A}) \in \{true, false\}$ to every structure \mathfrak{A} from \mathcal{K} . Equivalently, I may consider a property Π over \mathcal{K} as the class of all structures \mathfrak{A} from \mathcal{K} , such that $\Pi(\mathfrak{A}) = true$. If Π is a property over \mathcal{K} , then its complement $\bar{\Pi}$ is the property over \mathcal{K} , such that Π and $\bar{\Pi}$ disagree on every structure from \mathcal{K} .

In finite model theory, the class $\mathcal{F}(\tau)$ of finite τ -structures and its subclasses play a key role.

Let τ be a vocabulary with one binary relation symbol (other than the equality symbol), call it R . Then, every τ -structure is called a *directed graph* or *digraph*. If the τ -structure $\mathfrak{A} = \langle A, R^{\mathfrak{A}} \rangle$ has it that $R^{\mathfrak{A}}$ is symmetric, \mathfrak{A} is called a *graph*.

A recurring graph-property is n -COLORABILITY, where n -COLORABILITY(\mathfrak{G}) = *true* iff there is a function $f : G \rightarrow \{1, \dots, n\}$, such that if two vertices v, v' in G are joined by an edge, then $f(v) \neq f(v')$. Such a function f is called an n -coloring.

If a vocabulary τ contains the binary relation symbol $<$, every τ -structure interprets $<$ as a *linear order* on A , by postulation. That is, $<^{\mathfrak{A}}$ satisfies the following conditions for every $a, b, c \in A$:

- not $a <^{\mathfrak{A}} a$;
- $a <^{\mathfrak{A}} b$ or $b <^{\mathfrak{A}} a$ or $a = b$; and
- if $a <^{\mathfrak{A}} b$ and $b <^{\mathfrak{A}} c$, then $a <^{\mathfrak{A}} c$.

The symbol $<$ will be called the *linear order symbol* and any τ -structure is called a *linear ordered structure*.

Let \mathbf{L} be a logical language over the vocabulary τ for which the satisfaction relation \models is defined, as well as the notion of sentence. Let \mathcal{K} be a class of τ -structures. Then, a τ -sentence Φ in $\mathbf{L}(\tau)$ *expresses* a property Π over \mathcal{K} , if for every \mathfrak{A} from \mathcal{K} it is the case that

$$\mathfrak{A} \models \Phi \quad \text{iff} \quad \Pi(\mathfrak{A}) = \text{true}.$$

If there is a sentence in $\mathbf{L}(\tau)$ that expresses the property Π over \mathcal{K} , then Π is said to be *expressible* in $\mathbf{L}(\tau)$ over \mathcal{K} . If Π is expressible in $\mathbf{L}(\tau)$ over $\mathcal{F}(\tau)$ and it is clear from the context that I am working on finite structures, I will omit the phrase “over $\mathcal{F}(\tau)$ ”.

Let \mathcal{K} be a class of τ -structures. Let $\mathbf{L}(\tau)$ and $\mathbf{L}'(\tau)$ be logical languages for which the satisfaction relation \models is properly defined, as well as the notion of sentence. I say that $\mathbf{L}(\tau)$ has *less expressive power* (not strictly) on \mathcal{K} than $\mathbf{L}'(\tau)$, if for every property Π over \mathcal{K} , it is the case that if Π is expressible in $\mathbf{L}(\tau)$, then Π is expressible in $\mathbf{L}'(\tau)$; symbolically $\mathbf{L} \leq_{\mathcal{K}} \mathbf{L}'$. The shorthand notation $=_{\mathcal{K}}$ and $<_{\mathcal{K}}$ is defined in terms of $\leq_{\mathcal{K}}$ in the standard way. Again, if \mathcal{K} is the class of all finite τ -structures, I will omit to mention it.

Logics capturing complexity classes. Let τ be a vocabulary and let Π be a property over \mathcal{K} . Then, Π can be treated as a decision problem. In fact, it can be analyzed using the tools from complexity theory, after postulating an encoding of Π 's structures. A natural way of encoding a structure \mathfrak{A} is by listing A and by storing the interpretation of symbols in τ by listing the truth values on all tuples of the objects from A . See (Immerman 1999, Definition 2.1) for a detailed account. To the ends of this dissertation it suffices to observe that for every vocabulary τ the canonical encoding of a τ -structure \mathfrak{A} is $\mathcal{O}(\|A\|^c)$, for some constant c depending on τ .

Then, Π can be regarded as a language or a decision problem in the computational sense of the word, introduced in Section 2.4. Let \mathbf{C} be a complexity class and let $\mathbf{L}(\tau)$ be a logical language for which the satisfaction relation \models is defined, as well as the notion of sentence. Let \mathcal{K} be a class of finite τ -structures. Say that $\mathbf{L}(\tau)$ *captures* at least \mathbf{C} on \mathcal{K} , if each \mathbf{C} -decidable property over \mathcal{K} is expressible by $\mathbf{L}(\tau)$.

Every $\mathbf{L}(\tau)$ -sentence Φ and class of finite structures \mathcal{K} constitute the property

$$\{\mathfrak{A} \in \mathcal{K} \mid \mathfrak{A} \models \Phi\}. \quad (2.6)$$

Let \mathbf{C} be a complexity class. Then, the *expression complexity* of Φ on \mathcal{K} is in \mathbf{C} if (2.6) is decidable in \mathbf{C} . If (2.6) happens to be complete for \mathbf{C} , then I say that the expression complexity of Φ on \mathcal{K} is \mathbf{C} -complete. The *expression complexity* of $\mathbf{L}(\tau)$ on \mathcal{K} is in \mathbf{C} , if the expression complexity of every sentence in $\mathbf{L}(\tau)$ on \mathcal{K} is in \mathbf{C} . The expression complexity of $\mathbf{L}(\tau)$ on \mathcal{K} is \mathbf{C} -complete, if the expression complexity of every $\mathbf{L}(\tau)$ -sentence on \mathcal{K} is in \mathbf{C} and there is a sentence in $\mathbf{L}(\tau)$ that has a \mathbf{C} -complete expression complexity on \mathcal{K} .

If $\mathbf{L}(\tau)$ captures at least \mathbf{C} on \mathcal{K} and if the expression complexity of $\mathbf{L}(\tau)$ on \mathcal{K} is in \mathbf{C} , then I say that $\mathbf{L}(\tau)$ *captures* \mathbf{C} on \mathcal{K} .

If any of the previous definitions holds with respect to the class of all finite τ -structures $\mathcal{F}(\tau)$, for arbitrary τ , I omit to mention both $\mathcal{F}(\tau)$ and τ .

The following result started up the field of descriptive complexity and is commonly known as *Fagin's Theorem*.

2.5.1. THEOREM (FAGIN (1974)). *On finite graphs, Σ_1^1 captures NP.*

2.5.2. EXAMPLE. For future reference and concreteness, let me go through a small example. Let \mathcal{G} be the class of all finite graphs. Consider the Σ_1^1 -formula

$$\Xi = \exists X_1 \exists X_2 \exists X_3 \forall x_1 \forall x_2 (\Phi \wedge \Phi'),$$

where

$$\begin{aligned} \Phi &= \left(\bigvee_{i \in \{1,2,3\}} X_i(x_1) \right) \wedge \left(\bigwedge_{i \in \{1,2,3\}} \bigwedge_{j \in \{1,2,3\} - \{i\}} \neg (X_i(x_1) \wedge X_j(x_1)) \right) \\ \Phi' &= \left(\bigwedge_{i \in \{1,2,3\}} (X_i(x_1) \wedge X_i(x_2) \rightarrow \neg R(x_1, x_2)) \right). \end{aligned}$$

Now, consider the decision problem

$$\{\mathfrak{G} \in \mathcal{G} \mid \mathfrak{G} \models \Xi\}, \quad (2.7)$$

and observe that (2.7) is true on all graphs \mathfrak{G} for which there is a 3-coloring. This means that the decision problem constituted by Ξ on \mathcal{G} coincides with

3-COLORABILITY. Thus, Ξ expresses 3-COLORABILITY on \mathcal{G} . Since the decision problem n -COLORABILITY is complete for \mathbf{NP} , for every $n \geq 3$, so is the expression complexity of Ξ on finite graphs. Clearly, $\neg\Xi$ expresses $\overline{3\text{-COLORABILITY}}$, and therefore has a \mathbf{coNP} -complete expression complexity on finite graphs. \square

Having observed Fagin's Theorem, it is readily observed that Π_1^1 captures \mathbf{coNP} on graphs. In fact, the vast majority of complexity classes has been characterized in terms of logics, be it adaptations of first-order or second-order logic. These characterizations attract interest because if two logics are shown to capture the complexity class \mathbf{C} and \mathbf{D} on the same class of structures \mathcal{K} , an argument showing that their expressive power is not equal on \mathcal{K} , implies that $\mathbf{C} \neq_{\mathcal{K}} \mathbf{D}$. Fascinatingly however, as yet it is not known what language \mathbf{L} captures \mathbf{P} on graphs (that are not linearly ordered), such a language would be a steppingstone to prove that $\mathbf{L} \neq \Sigma_1^1$, and consequently that $\mathbf{P} \neq \mathbf{NP}$.

Let me close this section on a warning note. In the literature on computation and logic several measures of complexity are studied. Expression complexity is one of them. Another notion is the *model checking complexity* of a logic \mathbf{L} . In this framework, a fixed finite structure \mathfrak{A} is seen to give rise to the set

$$\{\Phi \mid \mathfrak{A} \models \Phi\}, \quad (2.8)$$

where Φ ranges over the \mathbf{L} -sentences. The model checking complexity of \mathbf{L} is then said to be in the complexity class \mathbf{C} , if for all suitable structures the (encoding of the) set (2.8) is \mathbf{C} -computable.

The reader is urged to bear in mind that the expression complexity and model checking complexity are worlds apart. In the expression complexity framework, the size of the input is the length of the encoding of structures, whereas in the model checking framework it is the length of the encoding of the sentences. The computational differences between the two notions can be dramatic. For instance, Σ_1^1 's expression complexity is \mathbf{NP} -complete (Fagin's Theorem), but its model checking complexity is $\mathbf{NEXPTIME}$.

In (Vardi 1982) the distinction between expression complexity and model checking complexity is clearly described, in terms of *expression* and *data* complexity. In (Gottlob, Leone, and Veith 1995) a systematic theory is pursued on the relation between a logic's expression complexity and its model checking complexity.

Chapter 3

Fragments of IF logic

From its conception Independence-friendly logic has been associated with semantic information games with imperfect information. In this chapter I will flesh out this connection, firstly by defining a natural set of game rules that can account for the imperfect information. To this end I make use of envelopes and teams of Eloises. I continue by studying the expressive power of the fragment of IF logic that defines games with *perfect recall*. Inspired by current research on independence-friendliness and modal logics, I isolate my own IF modal language and compare its computational behavior with full IF logic and modal logic. Further research along this way may result in IF modal languages that exercise “pleasant” computational behavior.

3.1 Introduction

Ever since (Henkin 1961) there is a tradition in logic to associate certain logical languages with imperfect information. The logical languages at hand aim to break away from the strong restrictions of quantifier dependence that the syntax of first-order (or modal) logic imposes. For example, in the first-order formula

$$\forall x_1 \exists y_1 \forall x_2 \exists y_2 R(x_1, y_1, x_2, y_2) \tag{3.1}$$

the quantifier $\exists y_2$ appears syntactically to the left of the quantifiers $\forall x_1$ and $\forall x_2$. Hence, $\exists y_2$ falls within the scope of $\forall x_1$ and $\forall x_2$. On closer inspection, it emerges that in first-order logic there is no way to rearrange the quantifiers in such a way that $\exists y_i$ falls only within the scope of $\forall x_i$, for $i \in \{1, 2\}$. Henkin defined a special construct that allows for such *dependence schemes*:

$$\left(\begin{array}{cc} \forall x_1 & \exists y_1 \\ \forall x_2 & \exists y_2 \end{array} \right) R(x_1, y_1, x_2, y_2).$$

The above constructs are usually called *partially ordered quantifiers*, and their study is taken up in Chapter 4 of this dissertation.

On a more abstract level, Hintikka and Sandu distinguish scopes of two kinds. The first is the *priority scope*, which is a syntactically determined relation between quantifiers. The second is the notion of *binding scope* which is a relation between quantifiers and variables. In first-order logic the two notions are inseparable. For instance, if $Qx \phi$ is a first-order formula of which $Q'y \psi$ is a sub-formula, then the variable y is within the scope of the quantifier Qx no matter what. Hintikka and Sandu (1997, pg. 366) argue that this requirement “is seen to be unmotivated by any deeper theoretical reasons and hence dispensable.” To repair “Frege’s fallacy”, they introduce the / device in first-order logic that may allow y to be outside the binding scope of Qx . If the / is indeed employed in this way, y is said to be *independent* of Qx . The language that incorporates these syntactic indications was coined *Independence-friendly logic*, or IF logic for short.¹ For instance, to indicate that in $\forall x_1 \exists y_1 \forall x_2 \exists y_2 \phi$ the variable y_2 be independent of $\forall x_1$ is thus achieved in the following IF-formula:

$$\forall x_1 \exists y_1 \forall x_2 (\exists y_2 / \{x_1\}) R(x_1, y_1, x_2, y_2). \quad (3.2)$$

In (Sandu 1993; Hintikka 1996; Hintikka and Sandu 1997) various natural language applications are supplied for IF logic. These applications typically include the so-called ϵ - δ definition of continuity and *Hintikka sentences*, addressed in Chapter 5 of this dissertation. Hintikka (2002b) also speculates on applications in quantum mechanics.

Since its conception IF logic has been provided with semantics in terms of *semantic games with imperfect information*. The partiality of information—i.e., imperfect information—present in these games can be seen to reflect the partial ordering of the quantifiers. Hintikka and Sandu’s move from first-order logic to IF logic can thus also be appreciated as a generalization of first-order logic so as to incorporate imperfect information. The question what the influx of imperfect information in semantic games for IF logic is exactly cannot be answered easily, but was mentioned as a topic of interesting future research in (van Benthem lecture notes) and (Hodges 2006). Part of this research will be taken up in the present chapter.

A lively branch of research in the logic community is the pursuit of fragments of particular logics that enjoy “more pleasant” properties than the original language. Many of these properties, such as the issues of satisfiability and expression complexity, are computationally motivated. In this chapter I will take an interest in fragments of IF logic. The first fragment is motivated by the game-theoretic interest in games with *perfect recall*, as I will consider the fragment of IF logic whose formulae define games with perfect recall. Note that games with perfect recall still have imperfect information, only the information flow is constrained

¹Independence-friendly logic is sometimes also referred to as *Information-friendly logic*, *Hyperclassical logic* (Hintikka 2002a), and *IF first-order logic* (Hintikka 1996). See (Feferman 2006) for a critical treatment of the first-order aspects of IF logic.

to meet certain criteria that will be explained below. The second fragment is inspired by ongoing research on modal logics. Modal logics are an excellent example of the quest for computationally well-behaved fragments of first-order logic. The hope is that certain independence-friendly modal languages have “attractive” computational properties. Here, I will remind the reader that modal logics were first studied in analytic philosophy, but eventually inspired Andréka, van Benthem, and Németi (1998) to define the so-called *Guarded Fragment*. The Guarded Fragment is an expressively strong, though decidable, fragment of first-order logic.

In Section 3.2, I give an overview of the literature on IF logic in order to make this chapter reasonably self-contained.

In Section 3.3, I revisit the issue of articulating a set of natural game rules that gives rise to the extensive games with imperfect information with respect to which IF formulae are evaluated. I will show that, combining hints from the available literature, the imperfect information in the semantic games for IF logic can be explained by thinking of teams of players that put moves in envelopes.

In Section 3.4, I study the fragment of IF logic whose semantic games—possibly with imperfect information—have perfect recall. It turns out that perfect recall has a strong limiting effect on the resulting logic’s expressive power, that is, the perfect recall fragment of IF logic coincides with first-order logic.

In Section 3.5, I dwell on the issue of what may be an independence-friendly modal logic. In the literature several proposals have been put forward, but pairwise all of them are different systems (syntactically, semantically, or both). I will isolate a modal fragment of IF logic, which constitutes the main result of this section. This language will be seen to give rise to game-theoretically proper games, unlike earlier proposed systems. I prove that the IF modal language has a Π_1^0 -hard satisfiability problem, i.e., it is undecidable.

Section 3.6 closes the chapter with concluding remarks.

3.2 Prerequisites

In this section, I give an overview of IF logic. More elaborate overviews are (Hintikka 1996; Hintikka and Sandu 1997; Dechesne 2005).

Syntax. I give an exposition of the syntax of IF logic shortly, but let me first dwell on what independence-friendliness boils down to on a more abstract level.

3.2.1. DEFINITION. Let τ be a vocabulary. Then, $\mathbf{FOV}'(\tau)$ is the fragment of $\mathbf{FOV}(\tau)$ all of whose formulae ϕ are in negation normal form and have it that all its variables and indices are quantified at most once. Let $\mathbf{FO}'(\tau)$ be defined from $\mathbf{FO}(\tau)$ in a similar way.

Let all conventions for $\mathbf{FOV}(\tau)$ be inherited by $\mathbf{FOV}'(\tau)$.

It is straightforward to show that $\mathbf{FOV}' = \mathbf{FOV}$, so restricting oneself to \mathbf{FOV}' does not imply a decrease in expressive power.

Let ϕ be a \mathbf{FOV}' -formula. Let C_ϕ denote the set of uniquely identified boolean connectives, quantifiers, and restricted quantifiers in ϕ . Call all objects in C_ϕ *operators*. As for the unique identification of the operators, recall that in ϕ every variable is quantified at most once. So every quantifier or restricted quantifier in ϕ is uniquely determined by its variable or index. To distinguish connectives, assume they appear indexed in C_ϕ .

Let ϕ be a first-order formula and let $Qx\psi$ be a subformula of ϕ , for $Q \in \{\exists, \forall\}$. Then, every operator in C_ψ is in the *priority scope* of Qx . For every $Q'y$ in C_ψ I say that Qx is *superordinate* to $Q'y$, and conversely, that $Q'y$ is *subordinate* to Qx . The same terminology applies to restricted quantifiers and boolean connectives.

For any pair $C, C' \in C_\phi$, let me write $C >_\phi C'$ if C is superordinate to C' in ϕ .

For the ease of exposition let me define the binding scope relation of a \mathbf{FOV}' -formula ϕ also in terms of C_ϕ . Strictly speaking this is a sloppy usage of notation, since the binding scope is a relation between quantifiers and variables, and not between quantifiers and quantifiers. Let \succ_ϕ denote the binding scope relation in ϕ . It is defined such that for any (restricted quantifier) C and arbitrary operator C' from C_ϕ , it is the case that $C \succ_\phi C'$ if C' occurs in the binding scope of C in ϕ . It was pointed out by Hintikka and Sandu, that for every \mathbf{FOV}' -formula ϕ , $>_\phi$ and \succ_ϕ are inextricable. That is, for every (restricted) quantifier C and arbitrary operator C' from C_ϕ , $C >_\phi C'$ iff $C \succ_\phi C'$. Independence-friendly logic loosens this requirement.

The syntax of Independence-friendly logic is obtained from \mathbf{FOV}' . Several definitions of the language of IF logic can be found in the literature, see (Hintikka 1996; Hodges 1997; Caicedo and Krynicki 1999; Janssen and Dechesne pear). The definition used in this dissertation differs from the languages in (Sandu 1993; Hintikka 1996; Hintikka and Sandu 1997) in that it allows for slashing over existentially quantified variables. On the other hand, I will ignore the possibility of slashing universally quantified variables and connectives.

3.2.2. DEFINITION. Let τ be a vocabulary. For the ease of exposition assume that every (restricted) quantifier Q appearing in a $\mathbf{FOV}'(\tau)$ is enclosed by brackets: (Q) . Then, $\mathbf{IFV}(\tau)$ is the smallest superset of $\mathbf{FOV}'(\tau)$ closed under the following condition:

(IF) If Φ is an $\mathbf{IFV}(\tau)$ -formula and $(\exists x)$ occurs in Φ in the priority scope of quantifiers and restricted quantifiers among which

$$Q_1y_1, \dots, Q_my_m, \bigcirc_{j_1 \in J_1} \dots \bigcirc_{j_n \in J_n},$$

then the formula resulting from replacing $(\exists x)$ by $(\exists x/\{\vec{y}, \vec{j}\})$ is also an $\mathbf{IF}(\tau)$ -formula, where $Q_i \in \{\exists, \forall\}$ and $\bigcirc_h \in \{\vee, \wedge\}$.

The set $\{\vec{y}, \vec{j}\}$ is called a *slash-set*. Let $\mathbf{IF}(\tau)$ be the smallest superset of $\mathbf{FO}'(\tau)$ closed under (IF).

I will write $\exists x$ rather than $(\exists x/\emptyset)$, and similarly for restricted quantifiers. Observe that $\mathbf{IF}\forall$ -formulae may contain existential quantifiers slashed over indices, but no slashed restricted quantifiers. Extensions in this respect are readily obtained, but will not be used in this chapter.

The condition (IF) can be seen to constitute an *IF procedure* that generates the formulae of IF logic. Henceforth, I will refer to this procedure as the *IF procedure constituted by (IF)*. Observe that the IF procedure constituted by (IF) preserves vocabulary, that is, it does not introduce new relation symbols or function symbols that were not already present in τ . Furthermore, observe that for every $\mathbf{IF}\forall$ -formula Φ it is the case that if it contains a negation, the negation occurs immediately in front of a relation symbol, and that every variable and index in Φ is quantified at most once.

Let the set of IF-formulae that are generated from a $\mathbf{FO}\forall'$ -formula ϕ by means of the IF procedure constituted by (IF) be denoted by $\mathbf{IF}(\phi)$, that is, $\mathbf{IF}(\phi)$ denotes the closure of $\{\phi\}$ under (IF).

3.2.3. EXAMPLE. (Continuation of Example 2.3.1) Revisit the formula $\phi(x_1) = \bigwedge_{i \in \{a,b\}} \exists x_2 R(x_1, x_2)$. Apart from $\phi(x_1)$ itself, the set $\mathbf{IF}(\phi(x_1))$ also contains the formula $\bigwedge_{i \in \{a,b\}} (\exists x_2/\{i\}) R(x_1, x_2)$. \square

3.2.4. EXAMPLE. Let $\psi(x_1)$ be the following \mathbf{FO}' -formula:

$$\forall x_2 (\neg R(x_1, x_2) \vee \forall x_3 (\neg R(x_2, x_3) \vee \exists x_4 (R(x_3, x_4))))).$$

Then, $\mathbf{IF}(\psi(x_1))$ contains four formulae—for every subset X of $\{x_2, x_3\}$ the following formula is in it:

$$\forall x_2 (\neg R(x_1, x_2) \vee \forall x_3 (R(x_2, x_3) \vee (\exists x_4/X)(R(x_3, x_4)))).$$

\square

For an $\mathbf{IF}\forall(\tau)$ -formula Φ , let $Free(\Phi)$ denote the set of *free variables* in Φ , inductively defined as follows:

$$\begin{aligned} Free(R(x_1, \dots, x_n)) &= \{x_1, \dots, x_n\}, & \text{for } R \in \tau \\ Free(R_i(x_1, \dots, x_n)) &= \{i, x_1, \dots, x_n\}, & \text{for } i \in \text{IND} \\ Free(\neg\Phi) &= Free(\Phi) \\ Free(\Phi \circ \Psi) &= Free(\Phi) \cup Free(\Psi), & \text{for } \circ \in \{\vee, \wedge\} \\ Free((\exists x/\dots)\Phi) &= Free(\Phi) - \{x\} \\ Free(\forall x \Phi) &= Free(\Phi) - \{x\} \\ Free(\bigcirc_{i \in I} \Phi) &= Free(\Phi) - \{i\}, & \text{for } \bigcirc \in \{\bigvee, \bigwedge\}. \end{aligned}$$

If a variable x occurs in a formula Φ and $x \notin \text{Free}(\Phi)$, then x is said to be *bound* in Φ . Likewise for indices. If $\text{Free}(\Phi) = \emptyset$, then Φ is called a *sentence*. Observe that if a variable x appears in the slash-set Y of the quantifier $(\exists y/Y)$ in an **IF**-formula Φ , then there is a quantifier $\mathbf{Q}z$ superordinate to $(\exists y/Y)$ in Φ such that $x = z$.

For an **IFV**(τ)-formula Φ , let $\text{Sub}(\Phi)$ be the set of *subformulae* in Φ , where

$$\begin{aligned} \text{Sub}(R(x_1, \dots, x_n)) &= \{R(x_1, \dots, x_n)\}, \quad \text{for } R \in \tau \\ \text{Sub}(R_i(x_1, \dots, x_n)) &= \{R_i(x_1, \dots, x_n)\}, \quad \text{for } i \in \text{IND} \\ \text{Sub}(\neg\Phi) &= \{\neg\Phi\} \cup \text{Sub}(\Phi) \\ \text{Sub}(\Phi \circ \Psi) &= \{\Phi \vee \Psi\} \cup \text{Sub}(\Phi) \cup \text{Sub}(\Psi), \quad \text{for } \circ \in \{\vee, \wedge\} \\ \text{Sub}((\exists x/X) \Phi) &= \{(\exists x/X) \Phi\} \cup \text{Sub}(\Phi) \\ \text{Sub}(\forall x \Phi) &= \{\forall x \Phi\} \cup \text{Sub}(\Phi) \\ \text{Sub}(\bigcirc_{i \in I} \Phi) &= \{\bigcirc_{i \in I} \Phi\} \cup \text{Sub}(\Phi), \quad \text{for } \bigcirc \in \{\bigvee, \bigwedge\}. \end{aligned}$$

Note that subformulae in an **IFV**-formula Φ are not necessarily **IFV**-formulae themselves. For instance, $(\exists y/\{x\}) \Psi$ is a subformula in $\forall x(\exists y/\{x\}) \Psi$ but it cannot be generated by the IF procedure constituted by (IF).

For an **IFV**(τ)-formula Φ , the set of operators C_Φ contains a unique token of every operator in Φ . So in case $(\exists x/X)$ appears in Φ , then $(\exists x/X) \in C_\Phi$; and similarly for $\bigvee_{i \in I}$. Since every variable and index is quantified at most once in **IFV**-formulae, variables can be used to identify quantifiers uniquely, just as in **FOV**'.

The notion of super- and subordinateness can be inherited from first-order logic without further ado, causing the priority scope relation \succ_Φ to be well-defined over the operators in **IFV**-formulae Φ .

The / affects the binding scope relation as follows: For every quantifier $\mathbf{Q}x$ and operator \mathbf{Q}' from C_Φ , write $\mathbf{Q}x \succ_\Phi \mathbf{Q}'$ if $\mathbf{Q}x >_\Phi \mathbf{Q}'$, and if \mathbf{Q}' stands for $(\exists y/Z)$, then $x \notin Z$; for every restricted quantifier $\bigcirc_{i \in I}$ and operator \mathbf{Q}' from C_Φ , write $\bigcirc_{i \in I} \succ_\Phi \mathbf{Q}'$ if $\bigcirc_{i \in I} >_\Phi \mathbf{Q}'$, and if \mathbf{Q}' stands for $(\exists y/Z)$, then $i \notin Z$. For instance, in case of (3.1) and (3.2), observe that

$$\begin{aligned} >_{(3.1)} &= >_{(3.2)} \\ \succ_{(3.1)} - \{\langle \forall x_1, \exists y_2 \rangle\} &= \succ_{(3.2)}. \end{aligned}$$

The relation \succ_Φ is also called Φ 's *independence scheme*.

Skolem semantics. It is a seminal result in the theory of first-order logic, see (Enderton 1972), that the first-order formula

$$\forall x_1 \exists y_1 \forall x_2 \exists y_2 R(x_1, y_1, x_2, y_2)$$

is true on a suitable structure \mathfrak{A} if, and only if, the following statement with function variables f_1 and f_2 holds:

$$\mathfrak{A} \models \exists f_1 \exists f_2 \forall x_1 \forall x_2 R(x_1, f_1(x_1), x_2, f_2(x_1, y_1, x_2)).$$

Here, the functions f_1 and f_2 are called *Skolem functions* and reflect that y_1 depends only on x_1 , whereas y_2 depends on x_1, y_1, x_2 . In Henkin quantifiers and IF logic, quantifier *independence* will be formalized by restricting the arguments of the Skolem function.

I paraphrase a procedure dubbed *inside-out Skolemization* by Dechesne (2005, pg. 43-4), for reasons that will become clear in due course. The result of this inside-out Skolemization is a second-order formula that correctly reflects the independence scheme at hand.²

Let Φ be an $\mathbf{IF}(\tau)$ -formula and let $(\exists x_1/X_1), \dots, (\exists x_m/X_m)$ be its existential quantifiers. The *Skolemization* of Φ is the second-order-formula

$$Sk(\Phi) = \exists f_1 \dots \exists f_m Sk'(\Phi),$$

where the formula $Sk'(\Phi)$ is the result of applying the procedure below to Φ : Put $\Psi_0 = \Phi$. If Ψ_i contains no existential quantifiers, put $Sk'(\Phi) = \Psi_i$; otherwise, increase i by 1 and apply the following rule:

- If $(\exists x_j/\{z_1, \dots, z_k\}) \Xi$ occurs as a subformula in Ψ_{i-1} under the priority scope of the (restricted) quantifiers $\mathbf{Q}_1 z_1, \dots, \mathbf{Q}_k z_k, \mathbf{Q}_{k+1} z_{k+1}, \dots, \mathbf{Q}_l z_l$ and Ξ does not contain existential quantifiers, then replace $(\exists x_j/\{z_1, \dots, z_k\}) \Xi$ in Ψ_{i-1} by $\Xi[x_j \mapsto f_j(z_{k+1}, \dots, z_l)]$, where f_j is an $(l-k)$ -ary function variable and $\Xi[x_j \mapsto f_j(z_{k+1}, \dots, z_l)]$ is the result of replacing every occurrence of x_j in Ξ by $f_j(z_{k+1}, \dots, z_l)$. Denote the result of this action by Ψ_i .

Let τ be a vocabulary, let \mathfrak{A} be a τ -structure, and let α be an assignment in \mathfrak{A} . Then, an $\mathbf{IF}\forall(\tau)$ -formula Φ is *true under Skolem semantics* on \mathfrak{A} under α iff $\mathfrak{A} \models Sk(\Phi)[\alpha]$. In this manner, two $\mathbf{IF}\forall(\tau)$ -formulae Φ and Ψ are equivalent, if for every τ -structure \mathfrak{A} and assignment α in \mathfrak{A} , it is the case that

$$\mathfrak{A} \models Sk(\Phi)[\alpha] \quad \text{iff} \quad \mathfrak{A} \models Sk(\Psi)[\alpha].$$

Now the *truth conditions* of Φ are defined but the *falsity conditions* of Φ remain undefined, i.e., the cases where Φ is false under Skolem semantics on \mathfrak{A} under α . I will spend some words on this issue but not until I have introduced the second semantical interpretation.

²The Skolemization procedure put forward in Sandu (1993) does not work, because the language $\mathbf{IF}\forall$ allows for slashing over existential quantifiers.

3.2.5. EXAMPLE. (Continuation of Examples 2.3.1 and 3.2.3) The Skolemization of $\Phi(x_1) = \bigwedge_{i \in \{a,b\}} (\exists x_2 / \{i\}) R(x_1, x_2)$ is

$$\exists f \bigwedge_{i \in \{a,b\}} R(x_1, f),$$

where f is a nullary function symbol, that is, f does not take any arguments. Thus $\Phi(x_1)$ is true on \mathfrak{A} under α iff there exists one object c that is both $R_a^{\mathfrak{A}}$ and $R_b^{\mathfrak{A}}$ -accessible from $\alpha(x_1)$. In comparison with the meaning of the unslashed formula $\phi(x_1)$ from Example 2.3.1 note that the operators “for both” and “exists” are effectively swapped. The swapping of slashed operators will be instrumental in the remainder of this chapter when proving multiple results about IF logic. \square

3.2.6. EXAMPLE. Consider the IF-formula with free variable w :

$$\Omega = \forall x \exists y (\exists z / \{x\}) (x = z \wedge w \neq y)$$

inspired by (Caicedo and Krynicki 1999, Example 1.4). Firstly observe that Ω 's dependence scheme is not transitive: Although $\forall x \succ_{\Omega} \exists y$ and $\exists y \succ_{\Omega} (\exists z / \{x\})$, it is the case that z is independent of $\forall x$:

$$\forall x \not\succeq_{\Omega} (\exists z / \{x\}).$$

The Skolemization of Ω is the formula $\exists f \exists g Sk'(\Omega)$, where

$$\begin{aligned} \Psi_0 &= \forall x \exists y (\exists z / \{x\}) (x = z \wedge w \neq y) \\ \Psi_1 &= \forall x \exists y (x = f(y) \wedge w \neq y) \\ Sk'(\Omega) &= \forall x (x = f(g(x)) \wedge w \neq g(x)). \end{aligned}$$

Note that the Skolemization procedure works from Ω 's inner quantifier to its outer quantifier. Hence the qualification “inside-out”.

To see the Skolem semantics at work, observe that Ω is true under Skolem semantics on a structure \mathfrak{A} under α iff

$$\mathfrak{A} \models \exists f \exists g \forall x (x = f(g(x)) \wedge w \neq g(x))[\alpha].$$

The formula Ω expresses *Dedekind infinity* of the universe A . The set A is Dedekind infinite, if for some $\alpha(w) \in A$, there exists a bijection from A to $A - \{\alpha(w)\}$. Suppose that A is Dedekind infinite, in virtue of the bijection $F : A \rightarrow A - \{\alpha(w)\}$. Let F^{-1} denote the inverse of F . Then, observe that

$$\mathfrak{A} \models \forall x (x = f(g(x)) \wedge w \neq g(x))[\alpha.f/F^{-1}, g/F]$$

since for every x , $F^{-1}(F(x)) = x$ and $F(x) \neq \alpha(w)$. Conversely, suppose that for some functions F, G , and $\alpha(w)$ it is the case that

$$\mathfrak{A} \models \forall x (x = f(g(x)) \wedge w \neq g(x))[\alpha.f/F, g/G].$$

Then, F must be the inverse of the bijection G . Since for every x it is the case that $G(x) \neq \alpha(w)$, G is a bijection of the required type.

It is worthwhile to observe that in the term $f(g(x))$ the function f still depends on x , albeit indirectly. This reflects the fact that in Ω , the variable z only depends on $\exists y$, which itself only depends on $\forall x$. Thus, using the game theorist’s parlance, the quantifier $\exists y$ is said to *signal* the value of x to z . The phenomenon of signaling in IF logic was first reported in (Hodges 1997) and studied in (Janssen and Dechesne to appear). \square

Game-theoretic semantics. IF logic has been given a game-theoretic semantics using extensive games with imperfect information. The game rules for the semantic games for the IF-formula Φ on the structure \mathfrak{A} under the assignment α are the same as the game rules for $Sem\text{-}game^{\mathbf{FO}}(\Phi^\spadesuit[\alpha], \mathfrak{A})$, where Φ^\spadesuit is the result of replacing every occurrence of $(\exists x/X)$ in Φ by $\exists x$. Thus, Φ^\spadesuit is a first-order formula, and the game rules of the semantic game for Φ on \mathfrak{A} under α are the same as the ones of a perfect information game.

About the game rules for IF semantic games that give rise to these extensive games, Hintikka says:

“How are the semantical game rules for IF first-order logic languages related to those of ordinary first-order languages? The relation is an interesting one—and a simple relationship at that. It is one of identity. The rules for making moves in a semantical game in IF first-order logic are precisely the same as those used in ordinary first-order logic [...]” (Hintikka 1996, pg. 57-8)

But how can this be?

How can IF semantic games generalize semantic games for first-order logic, if their game rules coincide? The answer is: the way Hintikka puts them to use. To make this point clearly, let me return to the game-theoretic semantics for first-order logic. In a sense, the sole purpose of these games is to allow for a winning strategy for Eloise, or not. For instance, it is very much indifferent to the game-theoretic framework what winning strategy is used exactly and with what histories it is in accordance with. In this manner semantic games for first-order logic merely serve to substantiate the phrase “winning strategy”.

Following Hintikka and Sandu’s lead the same holds for IF semantic games: truth of an IF-formula Φ under game-theoretic semantics on \mathfrak{A} under α is also defined as Eloise having a winning strategy in $Sem\text{-}game^{\mathbf{FO}}(\Phi^\spadesuit[\alpha], \mathfrak{A})$, but under the proviso that this winning strategy be *uniform* with respect to the independence scheme of Φ . That is, roughly speaking, if in Φ the variable x is independent of $\mathbf{Q}y$, then a winning strategy in the semantic game for Φ may not take the object assigned to y as an argument when deciding on x . Crucially, Hintikka and Sandu define the difference between first-order and IF semantic games on the

level of strategies, that is, by putting the constraint of uniformity on the class of strategies that are at Eloise's disposal.

The link with imperfect information games can be established as follows: Let $G = \text{Sem-game}^{\text{FO}}(\Phi^\spadesuit[\alpha], \mathfrak{A})$ be a semantic game induced by the IF-formula Φ , which is a game of perfect information. Then, there exists a transformation T that preserves the property of Eloise having a winning strategy, such that for every strategy S in G , S is uniform with respect to Φ 's independence scheme iff S is a strategy in the imperfect information game $T(G)$. Recall that in the context of extensive games with imperfect information, a strategy is a function assigning actions to *information partitions* rather than histories. In relation to the transformation T —to be specified—this means that every strategy in the semantic game for Φ has it that a move for x is prescribed only on the basis of those moves associated with the (restricted) quantifiers Q_1, \dots, Q_n , such that $Q_i \succ_\Phi Qx$, for all $1 \leq i \leq n$. Thus the game-theoretic face of uniformity constraints induced by independence schemes is imperfect information.

The transformation T comes rather naturally, yet in the literature no account is given how to understand the imperfect information on a game-theoretic level. That is, what game rules, attributes, or cognitive bounds one has to presuppose to explain the origins of the imperfect information in $T(G)$. This topic will be investigated in Section 3.3.

Before proceeding, let me spell out the details of the extensive game with imperfect information as one finds them in the literature, thus formalizing the result of the transformation T . The definitions are neatly described in (Dechesne 2005, Section 4.4) and I mostly follow those.

3.2.7. DEFINITION. Let τ be a vocabulary, let \mathfrak{A} be a τ -structure whose universe A does not contain the objects 0 and 1, and let α be an assignment in \mathfrak{A} . Let Φ be an $\mathbf{IF}(\tau)$ -formula. For convenience, let Qx range over quantifiers, $\bigcirc_{i \in I}$ over restricted quantifiers, and \circ over boolean connectives. Then, $\text{Sem-game}^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$ denotes $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, \langle U_i \rangle_{i \in N} \rangle$ the *semantic evaluation game in extensive form* of Φ on \mathfrak{A} under α , such that:

- $N = \{\text{Eloise, Abelard}\}$ are the players.
- $H = \bigcup_\Psi H_\Psi$ is the set of histories in the game, where Ψ ranges over the subformulae in Φ . H_Ψ is inductively defined as follows:
 - If $\Psi = \Phi$, then $H_\Psi = \{\varepsilon\}$ where ε is the initial history.
 - If $\Psi = \Psi_0 \circ \Psi_1$, then $H_{\Psi_i} = \{hi \mid h \in H_\Psi\}$, for $i \in \{0, 1\}$.
 - If $\Psi = Qx \Psi'$, then $H_{\Psi'} = \{ha \mid h \in H_\Psi \text{ and } a \in A\}$.
 - If $\Psi = \bigcirc_{i \in I} \Psi'$, then $H_{\Psi'} = \{hj \mid h \in H_\Psi \text{ and } j \in I\}$.

Histories give rise to assignments and subformulae. If $h \in H$, then the assignment induced by h , denoted α_h , is inductively defined as follows:

$$\begin{aligned}\alpha_\varepsilon &= \emptyset \\ \alpha_{hi} &= \alpha_h, \quad \text{if } h \in H_{\Psi \circ \Psi'}, \text{ for some } \Psi, \Psi' \\ \alpha_{hj} &= \alpha_h, \quad \text{if } h \in H_{\bigcirc_{i \in I} \Psi}, \text{ for some } \Psi \\ \alpha_{ha} &= \alpha_h \cup \{\langle x, a \rangle\}, \quad \text{if } h \in H_{\mathbf{Q}x \Psi}, \text{ for some } \Psi.\end{aligned}$$

If $h \in H$, then the subformula induced by h , denoted Φ_h , is inductively defined as follows:

$$\begin{aligned}\Psi_\varepsilon &= \Psi \\ (\Psi \circ \Psi')_{h0} &= \Psi_h \\ (\Psi \circ \Psi')_{h1} &= \Psi'_h \\ (\bigcirc_{i \in I} \Psi)_{hj} &= \Psi_h \\ (\mathbf{Q}x \Psi)_{ha} &= \Psi_h.\end{aligned}$$

Note that α_h interprets all free variables in Φ_h .

The set of actions that can be taken at history h is obviously read off from the subformula induced by h :

$$A(h) = \begin{cases} \{0, 1\} & \text{if } \Phi_h = \Psi \circ \Psi', \text{ for some } \Psi, \Psi' \\ I & \text{if } \Phi_h = \bigcirc_{i \in I} \Psi, \text{ for some } \Psi \\ A & \text{if } \Phi_h = \mathbf{Q}x \Psi, \text{ for some } \Psi. \end{cases}$$

- $P : H - Z \rightarrow N$ is the player function, naturally defined such that

$$P(h) = \begin{cases} \text{Abelard} & \text{if } h \in H_\Psi \text{ and } \Psi\text{'s main operator is } \forall, \wedge, \text{ or } \bigwedge \\ \text{Eloise} & \text{if } h \in H_\Psi \text{ and } \Psi\text{'s main operator is } \exists, \vee, \text{ or } \bigvee. \end{cases}$$

- $\mathcal{I}_{\text{Eloise}} = \bigcup_h \{I_h\}$ is the information set of Eloise, where h ranges over all histories such that $P(h) = \text{Eloise}$. In order to define the information partition I_h , let Ψ be the subformula of Φ such that $h \in H_\Psi$. Since $P(h) = \text{Eloise}$, Ψ 's main operator is either $(\exists x/X)$, \vee , or \bigvee . In the former case, put $Y = X$; otherwise, put $Y = \emptyset$. Define I_h as the following set:

$$\{h' \in H_\Psi \mid \text{for every } x \in (\text{dom}(\alpha_h) - Y), \alpha_h(x) = \alpha_{h'}(x)\}. \quad (3.3)$$

Recall that α_h is an assignment interpreting first-order variables and indices. The information set of Abelard contains only singleton information partitions: $\mathcal{I}_{\text{Abelard}} = \bigcup_{h \in P^{-1}(\text{Abelard})} \{\{h\}\}$.

- $U_{\text{Eloise}} : Z \rightarrow \{\text{win}, \text{lose}\}$ is the utility function, such that

$$U_{\text{Eloise}}(h) = \begin{cases} \text{win} & \text{if } \mathfrak{A} \models \Phi_h[\alpha_h] \\ \text{lose} & \text{if } \mathfrak{A} \not\models \Phi_h[\alpha_h], \end{cases}$$

where Φ_h and α_h are the formula and assignments induced by h , respectively.

Before proceeding observe that for every IF-formula Φ , structure \mathfrak{A} , and assignment α , the game $\text{Sem-game}^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$ satisfies both the von Neumann-Morgenstern property (indistinguishable histories have equal length) and the action consistency requirement (if $h, h' \in I \in \mathcal{I}_\exists$, then $A(h) = A(h')$). As such they are well-behaved extensive games with imperfect information.

Recall the notion of winning strategy in imperfect information, win-loss games G , from Section 2.2. A function S is a strategy for player i in G , if it assigns an action to every information partition belonging to i ; it is winning if it results in a win, against every strategy of i 's opponent. Thus, define: Φ is *true under game-theoretic semantics* on \mathfrak{A} under α iff Eloise has a winning strategy in $\text{Sem-game}^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$.

Better safe than sorry: We have two semantics for IF logic now. It is not hard to prove that truth under Skolem semantics and truth under game-theoretic semantics coincides. I refer the reader to (Dechesne 2005) for a more thorough exposition of this equivalence. Henceforth, I will write “ $\mathfrak{A} \models \Phi[\alpha]$ ” to denote that $\mathfrak{A} \models \text{Sk}(\Phi)[\alpha]$, or equivalently, that Eloise has a winning strategy in $\text{Sem-game}_{\text{Coal}}^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$.

The issue of falsity for IF logic is naturally discussed in terms of games. Truth of an IF-formula Φ on \mathfrak{A} under α is defined as the existence of a winning strategy for Eloise in $\text{Sem-game}^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$. Falsity of Φ on \mathfrak{A} under α is thus defined as the existence of a winning strategy for Abelard. Here the consequence of relying on imperfect information games manifests itself: Φ may be neither true nor false on \mathfrak{A} under α as games with imperfect information are not determined in general. In actual fact, undetermined IF semantic games are readily construed. Consider the sentence $\Phi = \forall x(\exists y/\{x\}) \text{BEAT}(x, y)$ and the structure $\mathfrak{A} = \langle A, \text{BEAT}^{\mathfrak{A}} \rangle$, where

$$\begin{aligned} A &= \{\text{rock}, \text{paper}, \text{scissors}\} \\ \text{BEAT}^{\mathfrak{A}} &= \{\langle \text{rock}, \text{scissors} \rangle, \langle \text{scissors}, \text{paper} \rangle, \langle \text{paper}, \text{rock} \rangle\}. \end{aligned}$$

Clearly neither Eloise nor Abelard has a winning strategy in Rock, Paper, Scissors.

On the whole, falsity conditions for IF logic have received only little attention in the literature, but see (Dechesne 2005) for an exception.

The following theorem about \mathbf{IF} 's expressive power is well-known.

3.2.8. THEOREM (SANDU (1993)). $\mathbf{IF} = \Sigma_1^1$.

A number of meta-logical properties follow from this result, including non-recursive axiomatizability. On finite structures, infer from Fagin's Theorem that **IF** captures **NP**. Finite model theory for IF logic was addressed in but a small number of publications, although it was considered a promising field of research, cf. (Caicedo and Krynicky 1999, pg. 18). Exceptions are (Sandu 1997) and, somewhat more loosely related, Chapter 5 of the present dissertation.

Other approaches to independence in logic. The Skolem semantics for IF logic is inherently non-compositional. That is, it does not determine the truth of an IF-formula on the basis of its subformulae and its logical constants, cf. (Janssen 1997). In fact, Hintikka and Sandu proclaim on numerous occasions, including (Hintikka 1996; Hintikka and Sandu 1997), that a compositional semantics for IF logic cannot exist.

In reaction, Hodges (1997) put forward a compositional semantics for IF-sentences. The innovative idea that underlies Hodges system is the evaluation of subformulae of IF-sentences with respect to *sets of assignments*, rather than single assignments as one has in Tarski semantics for first-order logic. In this manner, the interpretation of a subformula Φ of an IF-sentence on a structure \mathfrak{A} , denoted $\Phi^{\mathfrak{A}}$, is a set of sets of assignments. In case $X \in \Phi^{\mathfrak{A}}$, write $\mathfrak{A} \models_X \Phi$ and say that X is a *trump* of Φ on \mathfrak{A} . The elements in X are assignments to the free variables in Φ , coined *deals* by Hodges. The notion of trump, is defined inductively by means of the following set of clauses (for an IF-language without slashed connectives and universal quantifiers):

- A set X of deals is a trump for $R(\vec{x})$ on \mathfrak{A} iff it is not empty and for every $\alpha \in X$, $\mathfrak{A} \models R(\vec{x})[\alpha]$.
- A set X of deals is a trump for $\Psi \vee \Psi'$ on \mathfrak{A} iff X is not empty and there are U and V such that $X \subseteq U \cup V$, and U is a trump for Ψ on \mathfrak{A} and V is trump for Ψ' on \mathfrak{A} .
- A set X of deals is a trump for $\Psi \wedge \Psi'$ on \mathfrak{A} iff X is not empty and X is a trump for Ψ on \mathfrak{A} and X is trump for Ψ' on \mathfrak{A} .
- A set X of deals is a trump for $(\exists x/W) \Psi$ on \mathfrak{A} iff it is not empty and there is a trump Y for Ψ on \mathfrak{A} such that for each \simeq_W -set $Z \subseteq X$ there is an object $a \in A$ for which $\{\alpha.x/a \mid \alpha \in Z\} \subseteq Y$.
- A set X of deals is a trump for $\forall x \Psi$ on \mathfrak{A} iff the set $\{\alpha.x/a \mid \alpha \in X, a \in A\}$ is a trump for Ψ on \mathfrak{A} .

Here, $Z \subseteq X$ is a \simeq_W set, if it is non-empty and its assignments agree on all variables other than the ones in W .

A sentence Φ is *true under trump semantics* on \mathfrak{A} if, and only if, $\{\emptyset\}$ is a trump for Φ on \mathfrak{A} , where \emptyset is the empty assignment. Hodges (1997) proves that trump

semantics coincides with the Hintikka-Sandu interpretation of **IF**-sentences (i.e., under Skolem or game-theoretic semantics). Cameron and Hodges (2001) show by means of a combinatorial argument that there cannot exist a compositional semantics for IF logic in terms of single assignments. For the repercussions on the issue of compositionality and IF logic, see (Sandu and Hintikka 2001) and (Hodges 2001).

Compositional interpretations for independence-friendly languages are also provided in (Caicedo and Krynicki 1999; Väänänen 2002).

The interpretation of IF-formulae as sets of sets of assignments is taken one step further in current research by Väänänen (unpublished) under the header of *Team logic*. In his approach new predicates are introduced in the object language in order to make assertions about sets of assignments. In this manner, Väänänen subtly makes a distinction between the *dependence type* of a formula and its truth conditions. The dependence type of a formula is, inter alia, determined by the new predicates $=(x_1, \dots, x_n)$ that intuitively states that the variable x_n depends only on the variables x_1, \dots, x_{n-1} . So the IF-sentence

$$\forall x_1 \exists y_1 \forall x_2 (\exists y_2 / \{x_1, y_1\}) R(x_1, y_1, x_2, y_2)$$

is cast using the $=(\dots)$ predicate as

$$\forall x_1 \forall x_2 \exists y_1 \exists y_2 (=(x_1, y_1) \wedge =(x_2, y_2) \wedge R(x_1, x_2, y_1, y_2)).$$

As in trump semantics, formulae from Väänänen's logic are evaluated on a structure \mathfrak{A} with respect to a set of assignments X , called a *team*. Write $\mathfrak{A} \models_X =(x_1, \dots, x_n)$ iff the team X has the dependence type of $=(x_1, \dots, x_n)$, that is, for every $\alpha, \beta \in X$,

$$\text{if } \alpha(x_1) = \beta(x_1), \dots, \alpha(x_{n-1}) = \beta(x_{n-1}), \text{ then } \alpha(x_n) = \beta(x_n).$$

More detailed flows of information were studied in the *Partial Information logic* developed in (Parikh and Väänänen 2005) whose formulae give rise to imperfect information games in which Eloise may be partially informed about the previous actions. In Partial Information logic, the formula $\forall x (\exists y // f(x)) R(x, y)$ typically gives rise to a semantic game in which Eloise is not aware of x , but she is cognizant of $f(x)$. So in case the function f returns 1 if x is even and 0 otherwise, Eloise does not know x but she knows its parity. Partial Information logic explores the connection between social software and logic.

Janssen (2002) disagrees with the claim that IF logic formalizes a reasonable notion of independence in logic. According to his intuitions, the sentence $\exists x (\exists y / \{x\}) (x = y)$ is bound to be untrue on any structure with more than one element, since y has to be chosen equal to x yet independent of it. The sentence turns out true on any structure \mathfrak{A} though, in virtue of the strategy *play a, play a*, where a is any object in \mathfrak{A} . Janssen disagrees with Hintikka on the applications

for IF logic in natural language and quantum mechanics. Instead, Janssen (2002) develops a rival semantics for IF logic, coined *Subgame semantics*, that meets his intuitions and is compositional.

3.3 A proper rule book for IF games

In this section, I give a set of rules that give rise to games whose extensive game trees are described by IF-formulae, given that they are properly executed. This result shows that semantic games for IF logic rely on game-theoretic structures that can be accounted for by means of goal-oriented interaction, involving imperfect information. Essentially the approach taken in this section considers teams of Eloises. This approach is in line with suggestions from the literature: “One interpretation offered in the folklore is that all slashes make sense when we assume that [Eloise] and [Abelard] are really *teams* [...] but no precise formulation of this form has been specified so far by Hintikka or his critics” (van Benthem 2006, pg. 494).

Stating that the game-theoretic semantics for IF logic uses games with imperfect information is justified by the purely technical transformation T , as I observed in the previous section. As I put it, every semantic extensive game G can be transformed into an extensive game with imperfect information $T(G)$ preserving the property of Eloise having a winning strategy. However this does not specify how this transformation can be accounted for on the level of interacting agents. The game rules for IF semantic games simpliciter describe games of perfect information, and obviously, the transformation manoeuvre does not retrospectively change the game rules for IF semantic games in such a way that they give rise to games of imperfect information.

Indeed, no natural set of game rules has been articulated that gives rise to a semantic game of imperfect information $Sem\text{-}game^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$ in such a way that Φ 's independence scheme is reflected in the game's information sets. This very observation is due to van Benthem:

“*How can one play IF games?*” (van Benthem lecture notes, pg. 173)

and

“IF syntax is just a ‘specification’ for patterns of knowledge and ignorance. It does not say whether one can *design actual games* that meet these specifications. [...] Some ignorance is ‘public’, and part of the legitimate design of a game. Examples are putting moves in envelopes, or the dealing of hands to players. Such games can be played by ideal players without limitations on their capacities for reasoning and observation. Another, quite different source of ignorance

are players' limitations: they may not pay attention, have bounded memory, cheat, and so on." (van Benthem lecture notes, pg. 174)

Note that here, in the terminology used in Chapter 1 of this dissertation, van Benthem points at two different sources of imperfect information: through rules and through cognitive boundaries. Furthermore, note that van Benthem aims to introduce imperfect information on the level of agents, whereas Hintikka and Sandu introduced imperfect information on the level of strategies.

A complicating factor when putting together a set of game rules is the fact that IF-formulae typically give rise to games that violate the principle of *perfect recall*, cf. (van Benthem 2006, pg. 497). I defer a more elaborate treatment of perfect recall to Section 3.4. For now, let us say that player i enjoys perfect recall in an extensive game with imperfect information, if i recalls all her own previous moves (action recall), and she does not forget any item of information along the way (knowledge memory). Games that do not have perfect recall are hard to imagine in a game-theoretic framework, the reason being that it is hard to conceive of an otherwise perfectly rational agent who forgets information at some stage in the game and relearns it at a later stage.

Analogous to the card game of Bridge, Hintikka (1996, pg. 49) proposes to define game rules for IF semantic games in terms of a coalition of Eloises battling Abelard. The idea, stemming from (Barwise 1979) reappearing in (Pietarinen 2001), has it that every single existential variable x in an IF-formula is controlled by one player: Eloise $_x$.—Again, I content myself with a discussion of **IF**, ignoring restricted quantifiers. Including them in the analysis is straightforward.—The set of existential variables in Φ thus gives rise to a team of players, who share the same utility function. This move overcomes the troublesome property of IF semantic games violating perfect recall, for if Eloise $_x$ is to move, this is the first and the last time during the game that she has to move. So action recall cannot possibly be violated, since every Eloise moves at most once; and knowledge memory cannot be either, since game theory assigns knowledge to an agent only at stages in the game where the agent is to move. The idea of restoring perfect recall by installing teams of players has been discussed in game theory, see (von Neumann and Morgenstern 1944; Binmore 1996).

I introduce one Eloise, who controls the disjunctions: Eloise $_\vee$. This is motivated by the assumption that disjunctions are not slashed in **IF**. Consequently, therefore, Eloise $_\vee$ is perfectly informed throughout the game and is not subject to the "perfect recall problem". If one would introduce slashed disjunctions in one's IF syntax, one should split Eloise $_\vee$ in as many sisters as there are disjunctions in the formula at hand in very much the same way as I just assigned an Eloise to every variable.

Using hints from (van Benthem lecture notes) and (van Benthem 2006; Hintikka 1996), I introduce imperfect information by letting the players put their chosen object in envelopes—any semantic game for Φ requires as many envelopes

as there are bound variables in Φ . In particular, let ϵ carry the contents of the envelopes and let $\epsilon(x)$ denote the object assigned to variable x during the game. The function ϵ is effectively an assignment and I will regard it as such. Initially, all envelopes are empty: $\epsilon = \emptyset$.

Let Φ be an **IF**-formula, let \mathfrak{A} be a suitable structure, and let α be an assignment in \mathfrak{A} . Furthermore, let B be the set of Φ 's bound variables. Then, the *coalitional semantic game* for Φ on \mathfrak{A} under α starts out from the position $\langle \Phi[\alpha], \mathfrak{A}, \emptyset \rangle$. The game rules determine the next move in the game on the basis of the game's current position, as follows:

- In $\langle (\Phi_0 \vee \Phi_1)[\alpha], \mathfrak{A}, \epsilon \rangle$ player Eloise_\vee is shown the contents of the envelopes for the variables in $\text{dom}(\epsilon)$, that is, all variables bound so far. Then, she chooses $i \in \{0, 1\}$ and the game proceeds as $\langle \Phi_i[\alpha], \mathfrak{A}, \epsilon \rangle$.
- In $\langle (\Phi_0 \wedge \Phi_1)[\alpha], \mathfrak{A}, \epsilon \rangle$ Abelard chooses $i \in \{0, 1\}$; the game proceeds as $\langle \Phi_i[\alpha], \mathfrak{A}, \epsilon \rangle$.
- In $\langle (\exists x/X) \Phi[\alpha], \mathfrak{A}, \epsilon \rangle$ player Eloise_x is shown the contents of the envelopes for the variables in $\text{dom}(\epsilon) - X$. Then, she chooses $a \in A$, writes a on a piece of paper and puts it in the designated envelope. The game proceeds as $\langle \Phi[\alpha], \mathfrak{A}, \epsilon.x/a \rangle$.
- In $\langle \forall x \Phi[\alpha], \mathfrak{A}, \epsilon \rangle$ Abelard observes the content of all envelopes. Next, he chooses $a \in A$, writes a on a piece of paper and puts it in the designated envelope. The game proceeds as $\langle \Phi[\alpha], \mathfrak{A}, \epsilon.x/a \rangle$.
- $\langle R(x_1, \dots, x_n)[\alpha], \mathfrak{A}, \epsilon \rangle$ marks the end of the game. Eloise wins if the tuple $\langle \iota(x_1), \dots, \iota(x_n) \rangle$ is contained in $R^{\mathfrak{A}}$; otherwise, Abelard wins. The function ι is the assignment with domain $\text{dom}(\alpha) \cup \text{dom}(\epsilon)$ such that

$$\iota(x) = \begin{cases} \alpha(x) & \text{if } x \notin B \\ \epsilon(x) & \text{if } x \in B. \end{cases}$$

- $\langle \neg R(x_1, \dots, x_n)[\alpha], \mathfrak{A}, \epsilon \rangle$ is similar to the previous rule, with the winning conditions swapped.

The extensive coalitional game for an **IF**-formula Φ on \mathfrak{A} under α can be obtained from $\text{Sem-game}^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$, in a rather straightforward way. The main obstacle is that in coalitional IF games, the set of players N now equals $\{\forall\} \cup E$, where

$$\begin{aligned} E &= \{\text{Eloise}_\vee\} \cup E' \\ E' &= \{\text{Eloise}_x \mid x \in B\}. \end{aligned}$$

The player function needs to be revised in this respect, as follows:

$$P(h) = \begin{cases} \text{Abelard} & \text{if } h \in H_{\Psi \wedge \Psi'} \text{ or } h \in H_{\forall y \Psi}, \text{ for some } \Psi, \Psi' \text{ and } y \\ \text{Eloise}_{\forall} & \text{if } h \in H_{\Psi \vee \Psi'}, \text{ for some } \Psi \text{ and } \Psi' \\ \text{Eloise}_x & \text{if } h \in H_{(\exists x/X) \Psi}, \text{ for some } \Psi, X \text{ and } x. \end{cases}$$

The information set $\mathcal{I}_{\text{Eloise}}$ from $\text{Sem-game}^{\text{IF}}(\Phi[\alpha], \mathfrak{A})$ is partitioned into the information sets for all Eloises, using the player function, as follows:

$$\mathcal{I}_{\text{Eloise}_x} = \mathcal{I}_{\text{Eloise}} \cap \{h \in H \mid P(h) = \text{Eloise}_x\}.$$

Finally the utility function needs to be defined for every player in E . These steps lead to the extensive game with imperfect information $\text{Sem-game}_{\text{Coal}}^{\text{IF}}(\Phi[\alpha], \mathfrak{A})$. In order to use these games, one says that the coalition E has a winning strategy in $\text{Sem-game}_{\text{Coal}}^{\text{IF}}(\Phi[\alpha], \mathfrak{A})$, if for every player $i \in E$ there exists a strategy S_i such that the compound object $\langle S_i \rangle_{i \in E}$ constitutes a winning strategy in $\text{Sem-game}^{\text{IF}}(\Phi[\alpha], \mathfrak{A})$. I state without proof that Eloise has a winning strategy in the game $\text{Sem-game}^{\text{IF}}(\Phi[\alpha], \mathfrak{A})$ iff the coalition E has a winning strategy in $\text{Sem-game}_{\text{Coal}}^{\text{IF}}(\Phi[\alpha], \mathfrak{A})$. This equivalence establishes the adequacy of the previously defined rule book for IF semantic games.

Interestingly, trump semantics for IF logic (Hodges 1997) is compositional but at the “cost” of using sets of assignments instead of single assignments as employed for first-order logic. In (Cameron and Hodges 2001) it is shown that in fact no compositional semantics can be given for IF logic based on single assignments. My rule book shows that resolving the property of perfect recall also goes at the cost of teams of Eloises. This observation becomes all the more striking in view of the next section’s main result, which holds that IF-formulae whose semantic games can be played by one Eloise with perfect recall have first-order expressive power.

3.4 Perfect recall and IF logic

In the previous section, one of the troubles in designing a rule book for IF games was explaining semantic games without perfect recall. In this section, I will go about orthogonally, by isolating the fragment of IF logic all of whose formulae give rise to games with one Eloise who enjoys perfect recall. I show in Theorem 3.4.3 that $\text{IF}^{\text{PR}} = \text{FO}$.

The notion of perfect recall was introduced by Kuhn (1950, 1953). Games lacking perfect recall have not been very popular in game theory to say the least, cf. (Osborne and Rubinstein 1994, pg. 203-4). It is remarked by Piccione and Rubinstein (1997, pg. 4) that “traditional game theory has excluded games with imperfect recall from its scope.” So what is known about imperfect information

games in game theory actually concerns games with a restricted kind of imperfect information. In the literature perfect recall is often split in two complementary notions: *action recall* and *knowledge memory*. See (van Benthem 2001; Bonanno 2004) for axiomatizations of perfect recall and an overview of the literature.

Informally, a game G with imperfect information has action recall, if all of its players recall all moves they made themselves at any stage of the game they were to move. G has knowledge memory, if every move that is known at some point by some player i is known by i from then on.

3.4.1. DEFINITION. Let G be an extensive game with imperfect information. Define the *action experiences* of player i in history h , denoted $AR_i(h)$, as the sequence consisting of the actions the player undertook in h :

$$\begin{aligned} AR_i(\varepsilon) &= \langle \rangle \\ AR_i(ha) &= \begin{cases} AR_i(h) & \text{if } P(h) \neq i \\ \langle AR_i(h), a \rangle & \text{if } P(h) = i, \end{cases} \end{aligned}$$

where $\langle \rangle$ is the empty tuple.

Define the *knowledge experiences* of player i in history h , denoted $KM_i(h)$, as the sequence consisting of the information sets the player encountered in h :

$$\begin{aligned} KM_i(\varepsilon) &= \langle \rangle \\ KM_i(ha) &= \begin{cases} KM_i(h) & \text{if } P(h) \neq i \\ \langle KM_i(h), I \rangle & \text{if } P(h) = i, \end{cases} \end{aligned}$$

where I is the information partition in \mathcal{I}_i containing h .

G is said to have *action recall* (*knowledge memory*), if for every information partition I in G , if $h, h' \in I \in \mathcal{I}_i$ then $AR_i(h) = AR_i(h')$ ($KM_i(h) = KM_i(h')$). G is said to have the *perfect recall* property, if G has action recall and knowledge memory.

It was observed that IF semantic games violate perfect recall. Dechesne (2005, pg. 74) points out that the extensive game with imperfect information that would truthfully serve as a means to decide truth of

$$\exists x(\exists y/\{x\}) (x = y)$$

on a structure with two or more objects in its universe violates action recall. Likewise, van Benthem (2006) observes that an extensive game for

$$\forall x \exists y(\exists z/\{x\}) (x = z)$$

violates knowledge memory.

In this section I will study the fragment of IF logic whose semantic games enjoy perfect recall. Say that an **IF**-formula Φ has perfect recall (action recall,

knowledge memory), if for every structure \mathfrak{A} and assignment α the extensive game with imperfect information $Sem\text{-}game^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$ has perfect recall (action recall, knowledge memory). This definition of an IF formula having perfect recall is semantic driven in that it depends solely on its semantic games. Note that I shall restrict my exposition to \mathbf{IF} rather than $\mathbf{IF}\forall$, for no reasons other than simplicity. The same results can be obtained for $\mathbf{IF}\forall$.

3.4.2. DEFINITION. Let τ be a vocabulary. Let $\mathbf{IF}^{\mathbf{PR}}(\tau)$ be the fragment of $\mathbf{IF}(\tau)$ all of whose formulae have perfect recall.

In the remainder of this section, I will concern myself with the expressive power of $\mathbf{IF}^{\mathbf{PR}}$ and prove the following result.

3.4.3. THEOREM. $\mathbf{FO} = \mathbf{IF}^{\mathbf{PR}}$.

Theorem 3.4.3 shows, then, that the constraint of perfect recall is so restrictive that nothing of \mathbf{IF} 's second-order expressive powers remains. Conversely, it shows that every \mathbf{IF} -formula that has no first-order equivalent violates perfect recall.³

One direction of the proof of Theorem 3.4.3 is easy: Every first-order formula ϕ is an $\mathbf{IF}^{\mathbf{PR}}$ -formula (reading $\exists x$ as $(\exists x/\emptyset)$) whose semantic games have perfect information, hence perfect recall. The converse direction is more involved and hinges on an inductive rewriting argument. To get an intuitive understanding of this argument let Φ be an $\mathbf{IF}^{\mathbf{PR}}$ -formula and let G be one of Φ 's semantic games. Then suppose that at some stage in G a move for Eloise is triggered by $(\exists y/Y)$, and that at every earlier stage of the game where it was Eloise's turn she was perfectly informed about all earlier moves. The set Y indicates the imperfect information Eloise experiences, but note that every move she is incognizant of must have been made after her previous move—if any. This follows from the assumption that Eloise knew at every earlier stage of G what action has been performed and the assumption that G satisfy knowledge memory. So all variables in Y are universally quantified in Φ . Using two translation schemes (given in Propositions 3.4.7 and 3.4.8), I show that one can swap the universal quantifiers appearing in the priority scope of $(\exists y/Y)$ without affecting the truth conditions, in such a way that $\forall x_1 \dots \forall x_n (\exists y/Y)$ is a string in the newly obtained formula $\Phi^\#$, where $Y = \{x_1, \dots, x_n\}$. The final argument, backed by yet another translation scheme (given in Proposition 3.4.9), holds that in $\Phi^\#$ one can safely replace $\forall x_1 \dots \forall x_n (\exists y/Y)$ by $\exists y \forall x_1 \dots \forall x_n$. Thus one empties all slash-sets and obtains a first-order formula.

Before I get to the proof of Theorem 3.4.3, I explore in Propositions 3.4.4 and 3.4.5 what can be inferred about the syntax of an IF-formula with action recall

³In a computational study on games with imperfect information (Koller and Megiddo 1992) it was shown that perfect recall games are computationally less complex than imperfect information games, unless $\mathbf{P} = \mathbf{NP}$.

and/or knowledge memory. Then, the necessary translation schemes for IF logic are supplied in Propositions 3.4.7, 3.4.8, and 3.4.9.

It is straightforward to establish that the particulars of \mathfrak{A} and α are highly insignificant to $Sem\text{-}game^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$'s enjoying perfect recall: For every structure \mathfrak{B} and assignment β in \mathfrak{B} ,

$$Sem\text{-}game(\Phi[\alpha], \mathfrak{A}) \text{ has perfect recall iff } Sem\text{-}game(\Phi[\beta], \mathfrak{B}) \text{ has,} \quad (3.4)$$

provided A and B contain two or more objects. The reason for this proviso being that if a structure contains only one object, any semantic game it partakes in is a game with perfect information. Throughout this section I assume that all structures under consideration have two or more objects in their universe. Then, $Sem\text{-}game^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$'s having perfect recall is fully determined by the characteristics of the formula Φ .

3.4.4. PROPOSITION. *Let Φ be an \mathbf{IF} -formula. Then, the following are equivalent.*

- (1) *For every pair of existential quantifiers $(\exists x_i/X_i)$ and $(\exists x_j/X_j)$ in Φ , where $(\exists x_i/X_i)$ is superordinate to $(\exists x_j/X_j)$, it is the case that $x_i \notin X_j$.*
- (2) *Φ has action recall.*

Proof. (1) implies (2). This direction follows rather easily from Definition 3.2.7 of semantic games for IF-formulae and the assumption that (1). It needs notice, though, that disjunctions are not constrained by the assumption that (1). However, crucially, disjunctions are not slashed in \mathbf{IF} -formulae. Therefore, a move triggered by a disjunction implies that Eloise knows at that stage of the game all previous moves, including her own.

(2) implies (1). Suppose Φ has action recall. Let \mathfrak{A} be a suitable structure, with $\|A\| \geq 2$, and let α be an assignment in \mathfrak{A} . Since Φ has action recall, so has the game $Sem\text{-}game^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$. For the sake of the argument, suppose that $(\exists x_j/X_j) \Xi$ is a subformula in $(\exists x_i/X_i) \Psi$, and that the latter formula is a subformula in Φ . Hence, $(\exists x_i/X_i)$ is superordinate to $(\exists x_j/X_j)$. Let a, a' be two distinct objects in the universe of \mathfrak{A} and let h be a history in $H_{(\exists x_i/X_i) \Psi}$. By definition, h can be extended with a and a' . Therefore, $ha, ha' \in H_{\Psi}$. Iteratively extend $g = ha$ until $g \in H_{(\exists x_j/X_j) \Xi}$ while putting g to

$$\begin{aligned} g0 & \quad \text{if } g \in H_{\Psi_0 \vee \Psi_1} \text{ and } (\exists x_j/X_j) \Xi \text{ is a subformula of } \Psi_0 \\ g1 & \quad \text{if } g \in H_{\Psi_0 \vee \Psi_1} \text{ and } (\exists x_j/X_j) \Xi \text{ is a subformula of } \Psi_1 \\ ga & \quad \text{if } g \in H_{(\exists y/Y) \Psi'}, \end{aligned}$$

where g is used as a *program variable*. This process of extending ha is deterministic since every variable is quantified at most once. So if $(\exists x/X) \Psi$ is a subformula

of Φ , then it is the only occurrence of this particular subformula of Φ . Hence, this procedure terminates and returns a history from $H_{(\exists x_j/X_j)} \Xi$. Let g' be the result of extending ha' with elements from $\{0, 1, a\}$ as above.

Consider the partial assignments α_g and $\alpha_{g'}$. Since ha and ha' are prefixes of g and g' respectively, it is the case that α_g and $\alpha_{g'}$ are extensions of α_h and that $\alpha_g(x_i) = a$ and $\alpha_{g'}(x_i) = a'$. Furthermore, for every variable y in $\text{rng}(\alpha_g) - \{x_i\}$ on which α_h is undefined, it is the case that $\alpha_g(y) = \alpha_{g'}(y) = a$.

For contradiction, suppose that (1) does not hold, i.e., that $x_i \in X_j$. Then, for every variable $y \in (\text{rng}(\alpha_g) - X_j)$, $\alpha_g(y) = \alpha_{g'}(y)$, since α_g and $\alpha_{g'}$ only disagree on x_i and $x_i \in X_j$. Hence by (3.3) in Definition 3.2.7 on page 34, g and g' sit in the same information partition I belonging to Eloise. Yet, the action experiences of Eloise in g and g' are not equal, formally $AR_{\text{Eloise}}(g) \neq AR_{\text{Eloise}}(g')$, as the former contains a and the latter a' . This contradicts the initial assumption of $\text{Sem-game}^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$ having action recall. So $x_i \notin X_j$ after all. \square

If an IF-formula has knowledge memory, in any of its semantic games when Eloise is to select an object for variable x_i and she knows the objects assigned to the variables z_1, \dots, z_m , then at any later stage of the game where she has to move, she still knows the objects assigned to z_1, \dots, z_m . Proposition 3.4.5 formalizes this intuition but uses a rather complicated formulation to facilitate a more perspicuous proof. Before we get to the proposition, let me rephrase its condition (1) in line with the previous intuitive account of knowledge memory. Let $(\exists x_i/X_i)$ and $(\exists x_j/X_j)$ be quantifiers in an \mathbf{IF} -formula Φ with knowledge memory such that $(\exists x_i/X_i)$ is superordinate to $(\exists x_j/X_j)$ in Φ . Then, if $\mathbf{Q}z$ appears superordinate to $(\exists x_i/X_i)$ in Φ and $z \notin X_i$, then $z \notin X_j$.

3.4.5. PROPOSITION. *Let Φ be an \mathbf{IF} -formula. Then, (1) and (2) are equivalent:*

(1) *For every $\mathbf{Q}x_i$ and $(\exists x_j/X_j)$, where $\mathbf{Q}x_i$ stands for $\forall x_i$ or $(\exists x_i/X_i)$ the former implies the latter:*

- (a) *$\mathbf{Q}x_i$ is superordinate to $(\exists x_j/X_j)$ in Φ and X_j does not contain x_i .*
- (b) *There is a $(\exists x_k/X_k)$ such that in Φ , $(\exists x_k/X_k)$ is subordinate to $\mathbf{Q}x_i$, superordinate or equal to $(\exists x_j/X_j)$, $x_i \notin X_k$, and for every $(\exists x_l/X_l)$ subordinate to $\mathbf{Q}x_i$ and superordinate to $(\exists x_k/X_k)$, $x_i \in X_l$.*

(2) *Φ has knowledge memory.*

Proof. (1) implies (2). Follows easily, see also the comments in the proof of Proposition 3.4.4.

(2) implies (1). Suppose Φ has knowledge memory. Let \mathfrak{A} be a suitable structure, whose universe contains two or more elements and let α be an assignment in \mathfrak{A} . Then, $\text{Sem-game}^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$ is a game with knowledge memory. Suppose (a) holds for Φ , and for the sake of contradiction suppose (b) does not hold for Φ . For

convenience, assume the variables in Φ are numbered in such a way that $i < j$ and for every $i \leq m, m' \leq j$, $\mathbf{Q}x_m$ is superordinate to $\mathbf{Q}x_{m'}$ in Φ iff $m < m'$. On this convention, it follows from the fact that not-(b) that for every $i < k \leq j$ such that $\mathbf{Q}_k x_k = (\exists x_k/X_k)$ and $x_i \in X_k$, there is an $i < l < k$ such that $\mathbf{Q}_l x_l = (\exists x_l/X_l)$ and $x_i \notin X_l$. Since k may also be j it follows from $x_i \notin X_j$, that there is an $i < k \leq j$ such that $\mathbf{Q}_k x_k = (\exists x_k/X_k)$ and $x_i \notin X_k$. Fix any such k and let $i < l < k$ be such that $\mathbf{Q}_l x_l = (\exists x_l/X_l)$ and $x_i \in X_l$.

Let the quantifiers $(\exists x_l/X_l)$ Ξ_l and $(\exists x_k/X_k)$ Ξ_k be the main operators of the subformulae $(\exists x_l/X_l)$ Ξ_l and $(\exists x_k/X_k)$ Ξ_k in Φ , respectively. Since $l < k$, $(\exists x_k/X_k)$ Ξ_k is a subformula in $(\exists x_l/X_l)$ Ξ_l .

Let a, a' be two distinct objects in \mathfrak{A} 's universe. Let h be a history in $H_{\mathbf{Q}x_i \Psi}$. Then, ha and ha' are distinct histories in H_Ψ extending h . Iteratively extend $g_l = ha$ until $g_l \in H_{(\exists x_l/X_l) \Xi_l}$ while putting g_l to

$$\begin{aligned} g_l 0 & \quad \text{if } g_l \in H_{\Psi_0 \vee \Psi_1} \text{ and } (\exists x_l/X_l) \Xi_l \text{ is a subformula of } \Psi_0 \\ g_l 1 & \quad \text{if } g_l \in H_{\Psi_0 \vee \Psi_1} \text{ and } (\exists x_l/X_l) \Xi_l \text{ is a subformula of } \Psi_1 \\ g_l a & \quad \text{if } g_l \in H_{(\exists y/Y) \Psi'}, \end{aligned}$$

where, again, g_l is used as a program variable. Let g'_l be the result of extending ha' with elements from $\{0, 1, a\}$ as above. By the same argument used in the proof of Proposition 3.4.4, derive that α_{g_l} and $\alpha_{g'_l}$ agree on every variable except x_i . Since $x_i \notin X_l$ it is the case that g_l and g'_l sit in different information partitions for Eloise. Call these information partitions I and I' , respectively.

Iteratively extend $g_k = g_l$ until $g_k \in H_{(\exists x_k/X_k) \Xi_k}$ while putting g_k equal to

$$\begin{aligned} g_k 0 & \quad \text{if } g_k \in H_{\Psi_0 \vee \Psi_1} \text{ and } (\exists x_k/X_k) \Xi_k \text{ is a subformula of } \Psi_0 \\ g_k 1 & \quad \text{if } g_k \in H_{\Psi_0 \vee \Psi_1} \text{ and } (\exists x_k/X_k) \Xi_k \text{ is a subformula of } \Psi_1 \\ g_k a & \quad \text{if } g_k \in H_{(\exists y/Y) \Psi'}. \end{aligned}$$

Let g'_k be the result of extending g'_l in the same way. Still, α_{g_k} and $\alpha_{g'_k}$ agree on every variable but x_i . Since $x_i \in X_k$, it is the case that g_k and g'_k are indistinguishable for Eloise, see (3.3) in Definition 3.2.7 on page 34. That is, they sit in the same information partition belonging to Eloise. Yet, the knowledge experiences of Eloise in g_k and g'_k are not equal, since $KM_{\text{Eloise}}(g_k)$ contains I and $KM_{\text{Eloise}}(g'_k)$ contains I' . This contradicts the initial assumption of Φ having knowledge memory. \square

3.4.6. PROPOSITION. *Let Φ be an IF-formula with knowledge memory. Then, for every $\mathbf{Q}x_i$ and $(\exists x_j/X_j)$, where $\mathbf{Q}x_i$ stands for $\forall x_i$ or $(\exists x_i/X_i)$, (a) implies (b):*

- (a) *There is a \vee in Φ , such that $\mathbf{Q}x_i$ is superordinate to \vee and \vee is superordinate to $(\exists x_j/X_j)$.*

(b) X_j does not contain x_i .

Proof. Similar to the proof of Proposition 3.4.5, using the fact that **IF** does not contain slashed disjunctions. This has the effect that at every move triggered by a disjunction Eloise knows all past moves. \square

In order to prove this section's main theorem, I lay down some easy equivalence schemes. More advanced schemes can be found in (Caicedo, Dechesne, and Janssen in preparation).

3.4.7. PROPOSITION. *Let Φ be an **IF**-formula containing $\forall x (\Psi \wedge \Psi')$ as a subformula. Let x' be a variable not occurring in Φ and let $\Psi'[x \mapsto x']$ denote the result of replacing every occurrence of x in Ψ' by x' . Let Φ' denote the result of replacing $\forall x (\Psi \wedge \Psi')$ in Φ by $(\forall x \Psi \wedge \forall x' \Psi'[x \mapsto x'])$. Then, the following hold:*

- (1) Φ' is an **IF**-formula.
- (2) Φ and Φ' are equivalent.
- (3) Φ has perfect recall iff Φ' has.

Proof.

- (1) Trivial. (Note that the more naive $(\forall x \Psi \wedge \forall x \Psi')$ is not an **IF**-formula on account of the fact that x is quantified twice. In Φ' this obstacle is avoided in a straightforward manner.)
- (2) Consider the Skolemization of Φ of which $\forall x (\exists \wedge \exists')$ is a subformula. $Sk(\Phi)$ is a formula from well-understood, second-order logic. So, in $Sk(\Phi)$ pushing the universal quantifier $\forall x$ inside the brackets goes without affecting the truth conditions. Furthermore, renaming of variables is unproblematic, and in particular $\forall x \exists'$ is equivalent to $\forall x' \exists'[x \mapsto x']$. Observe that $Sk(\Phi')$ is syntactically equivalent to $Sk(\Phi)$ after pushing $\forall x$ inside and renaming x to x' in \exists' .
- (3) Follows from Propositions 3.4.4 and 3.4.5.

This concludes the proof. \square

3.4.8. PROPOSITION. *Let Φ be an **IF**-formula containing $\forall x \forall y \Psi$ as a subformula. Let Φ' denote the result of replacing $\forall x \forall y \Psi$ in Φ by $\forall y \forall x \Psi$. Then, the following hold*

- (1) Φ' is an **IF**-formula.
- (2) Φ and Φ' are equivalent.

(3) Φ has perfect recall iff Φ' has.

The argumentation of the soundness of the latter schemes is very much in line with the argumentation from first-order logic. The next proposition has genuine IF influx.

3.4.9. PROPOSITION. *Let Φ be an **IF**-formula containing $\forall x(\exists y/Y) \Psi$ as a subformula, where $x \in Y$. Let Φ' denote the result of replacing $\forall x(\exists y/Y) \Psi$ in Φ by $(\exists y/Y - \{x\})\forall x \Psi$. Then, the following hold:*

- (1) Φ' is an **IF**-formula.
- (2) Φ and Φ' are equivalent.
- (3) Φ has perfect recall iff Φ' has.

Proof.

- (1) Trivial.
- (2) This is most easily observed by considering the Skolemizations of Φ and Φ' . The inside-out Skolemization is defined non-deterministically, so one cannot say right away that $Sk(\Phi)$ and $Sk(\Phi')$ are syntactic copies of each other. Instead, once one has $Sk(\Phi)$, it is easy to apply the inside-out Skolemization procedure to Φ' in such a way that it returns a syntactic copy of $Sk(\Phi)$. Hence, the claim follows.
- (3) Follows from Propositions 3.4.4 and 3.4.5.

This concludes the proof. □

As noted by van Benthem (lecture notes, pg. 166) the latter equivalence scheme resembles the Thompson transformation *interchange of moves* known from game theory, see (Thompson 1952). A true application of game theory indeed! The relation between Thompson transformations and IF logic was taken up in (Dechesne 2006). As an aside, observe that despite the fact the truth conditions are preserved under the transformation assessed in Proposition 3.4.9, the *falsity conditions* are subject to change. Those are not of interest to the current end.

Armed with the formal apparatus, we now prove the main result of this section, Theorem 3.4.3.

Proof of Theorem 3.4.3. From left to right. Let ϕ be a **FO**-formula. Then, ϕ has an equivalent first-order formula ϕ' in which every variable is quantified at most once. In ϕ' replace every $\exists x$ by $(\exists x/\emptyset)$, and refer to the result by Φ . The object Φ is an *iflogic*-formula in the vocabulary of ϕ' , and ϕ' and Φ are

equivalent.⁴ Since all Φ 's slash-sets are empty, all of Φ 's semantic games are games of perfect information. That is, every information partition of Eloise is a singleton. It follows trivially that Φ has both action recall and knowledge memory, and therefore perfect recall. Hence, ϕ is equivalent to the $\mathbf{IF}^{\mathbf{PR}}$ -formula Φ .

From right to left. Let A be a search algorithm that works on the syntactic tree of \mathbf{IF} -formulae and returns the first quantifier of the form $(\exists y/Y)$ it encounters, where Y non-empty. A proceeds *depth-first* and *top down* preferring the left subformula over the right one, in case a connective is encountered. Let $A(\Phi)$ denote the output of A when applied to the syntactic tree of the \mathbf{IF} -formula Φ . If Φ does not have a subformula of the form $(\exists y/Y) \Psi$, where Y non-empty, $A(\Phi)$ stands for *false*.

Fix an $\mathbf{IF}^{\mathbf{PR}}$ -formula Φ . In case $A(\Phi) = \textit{false}$, Φ is in fact equivalent to a first-order formula and the result follows directly.

In case $A(\Phi) = (\exists y/Y)$, the way A proceeds guarantees that the path leading from the root $\langle \rangle$ of the syntactic tree to $(\exists y/Y)$ looks schematically as follows:

$$\langle \rangle >_{\Phi} \dots >_{\Phi} \star >_{\Phi} C_1 >_{\Phi} \dots >_{\Phi} C_n >_{\Phi} (\exists y/Y), \quad (3.5)$$

where \star stands for $\langle \rangle$, \vee , or $(\exists z/\emptyset)$; and C_i stands for $\forall x_{f(i)}$ or \wedge . So before Eloise's move was triggered by $(\exists y/Y)$, Abelard moved for the operators C_1, \dots, C_n . The symbol \star indicates the beginning of the game or a previous move by Eloise. Here, f is a bookkeeping device mapping the integers $1, \dots, n$ onto Φ 's variables $x_{f(1)}, \dots, x_{f(n)}$.

A 's way of traversing the tree guarantees that \star cannot equal $(\exists z/Z)$ where Z non-empty, for otherwise A would have terminated on $(\exists z/Z)$ instead of $(\exists y/Y)$.

Let $J = \{j_1, \dots, j_m\}$ be the set of integers j in the interval from 1 to n , so that C_j stands for $\forall x_{f(j)}$. I lay down the following claim.

Claim. Y is a subset of $\{f(j) \mid j \in J\}$.

Proof of claim. In order to prove this claim, we make a case distinction:

- $\star = \langle \rangle$: In fact, $(\exists y/Y)$ is in the scope of exactly $\forall x_{f(j_1)}, \dots, \forall x_{f(j_m)}$, since in Φ no more quantifiers are superordinate to $(\exists y/Y)$. Hence, the claim follows.
- $\star = \vee$: Follows from Proposition 3.4.6.
- $\star = (\exists z/\emptyset)$: Suppose for the sake of contradiction that there is a $u \in Y$ such that $\mathbf{Q}u$ is superordinate to $(\exists z/\emptyset)$ in Φ . Then, clause (a) from Proposition 3.4.5 holds with respect to u and Y . Clause (b) fails, however, since $u \notin \emptyset$. Contradiction. \square

⁴Variables have to be renamed, because it was observed in (Janssen 2002) that ϕ' does *not* have to be equivalent to Φ , once a variable in ϕ' is quantified more than once.

The latter claim holds that Eloise can only be ignorant of the moves triggered by the universal quantifiers among C_1, \dots, C_n . Any move that was made before \star —if any—is known to Eloise when she is to move for $(\exists y/Y)$, as well as the choices made for the conjunctions among C_1, \dots, C_n .

Suppose J itself is not equal to an interval $\{\min(J), \dots, n\}$ in $\{1, \dots, n\}$. That is, the universal quantifiers $\forall x_{f(\min(J))}, \dots, \forall x_{f(n)}$ do not form a block that itself is superordinate to $(\exists y/Y)$. Then, there is a greatest integer j in J for which C_{j+1} stands for \wedge . Consider the subformula $\forall x_{f(j)} (\Psi C_{j+1} \Psi')$. By Proposition 3.4.7, it may be replaced in Φ by

$$(\forall x_{f(j)} \Psi C_{j+1} \forall x'_{f(j)} \Psi' [x_{f(j)} \mapsto x'_{f(j)}])$$

without affecting Φ 's truth condition, while preserving perfect recall. Let Φ' denote the result of this replacement. Clearly, it is the case that $A(\Phi') = (\exists y/Y)$. That is, by this move no new independences were introduced in the subformulae in Φ that were checked by A before it found $(\exists y/Y)$. In the path in the syntactic tree of Φ' leading from the root to $(\exists y/Y)$, the logical operators C_j and C_{j+1} appear swapped as compared to the path (3.5).

In this manner, swap all universal quantifiers and conjunctions, until the path from the root to $(\exists y/Y)$ looks like

$$\langle \rangle >_{\Phi} \dots >_{\Phi} \star >_{\Phi} \wedge >_{\Phi} \dots >_{\Phi} \wedge >_{\Phi} \forall x_{f(j_1)} >_{\Phi} \dots >_{\Phi} \forall x_{f(j_m)} >_{\Phi} (\exists y/Y).$$

(Modulo the fact that $x_{f(j)}$ may be renamed, and actually be called $x'_{f(j)}$ now. But this detail should not be too disturbing.) As I pointed out before, the swapping of universal quantifiers and conjunctions can be done preserving truth conditions and perfect recall. Put the result of doing so Φ^* , and observe again that $A(\Phi^*) = (\exists y/Y)$.

By the claim proved earlier, $Y \subseteq J$. Consider a pair $j, j' \in \{1, \dots, n\}$, such that $x_{f(j)} \in Y$, $x_{f(j')} \notin Y$, and $\forall x_{f(j)} \forall x_{f(j')} \Psi$ is a subformula in Φ^* —if any. By Proposition 3.4.8, one may replace $\forall x_{f(j)} \forall x_{f(j')} \Psi$ in Φ^* by

$$\forall x_{f(j')} \forall x_{f(j)} \Psi,$$

preserving the truth conditions and perfect recall. Furthermore, A applied to the formula in which $\forall x_{f(j)} \forall x_{f(j')}$ are swapped, still returns $(\exists y/Y)$. Continue swapping universal quantifiers following the routine, until for every pair $j, j' \in \{1, \dots, n\}$ if $x_{f(j)} \in Y$ and $x_{f(j')} \notin Y$ then $\forall x_{f(j')}$ is superordinate to $\forall x_{f(j)}$ in the resulting formula denoted $\Phi^\#$. The path in the syntactic tree of $\Phi^\#$ from \star to $(\exists y/Y)$ looks as follows:

$$\star >_{\Phi} \wedge >_{\Phi} \dots >_{\Phi} \wedge >_{\Phi} \forall u_1 >_{\Phi} \dots >_{\Phi} \forall u_k >_{\Phi} \forall v_1 >_{\Phi} \dots >_{\Phi} \forall v_l >_{\Phi} (\exists y/Y),$$

where $Y = \{v_1, \dots, v_l\}$ and $J - Y = \{u_1, \dots, u_k\}$.

Consider the subformula $\forall v_l(\exists y/Y) \Psi$ of $\Phi^\#$. By definition, $v_l \in Y$. Hence, by Proposition 3.4.9 $\forall v_l(\exists y/Y) \Psi$ may be replaced by

$$(\exists y/Y - \{v_l\})\forall v_l \Psi$$

in $\Phi^\#$, preserving truth conditions and perfect recall. Continue moving up $(\exists y/\dots)$ in the syntactic tree until $(\exists y/\dots)$ is superordinate to $\forall x_{v_1}, \dots, \forall x_{v_l}$. Or equivalently, until \dots stands for the empty set. Let Φ^+ be the result of this procedure. Observe that $\Phi^\#$ and Φ^+ are equivalent and that Φ^+ enjoys perfect recall, due to Proposition 3.4.9. Finally, observe that the number of subformulae in Φ^+ of the form $(\exists w/W) \Psi$, where W non-empty, has decreased by one, as compared to the number of such subformulae in $\Phi^\#$.

Thus apply the above routine to every subformula in Φ^+ of the form $(\exists w/W) \Psi$, where W non-empty. In the resulting formula only empty sets appear and its truth conditions are therefore first-order. This completes the proof. \square

In relation to the theory of Independence-friendly logic, Theorem 3.4.3 shows that the difference in expressive power between **FO** and **IF** (or **FO** \vee and **IF** \vee) is due to exactly those **IF**-formulae that violate perfect recall. That is, every **IF**-formula that has no first-order equivalent violates perfect recall.

From a game-theoretic perspective Theorem 3.4.3 holds that for every **IF**^{PR}-formula Φ there is a **FO**-formula ϕ such that for every suitable structure \mathfrak{A} and assignment α in \mathfrak{A} , it is the case that Eloise has a winning strategy in the game $Sem\text{-}game^{\mathbf{IF}}(\Phi[\alpha], \mathfrak{A})$ iff she has one in $Sem\text{-}game^{\mathbf{IF}}(\phi[\alpha], \mathfrak{A})$. This is interesting because it shows that certain classes of games with imperfect information—the ones of Φ for instance—are equivalent (from Eloise's perspective) to certain classes of perfect information games. On the other hand the games are very much different from Abelard's perspective, since semantic games for first-order formulae are determined, whereas this need not be the case for games with perfect recall. Here the consequence is observed of only focusing on truth conditions and neglecting the falsity conditions of IF logic.

As I observed at the end of Section 3.3, it is known from (Cameron and Hodges 2001) that no compositional semantics can be provided for IF logic based on single assignments. Likewise my rule book introduces a team of Eloises to circumvent the troublesome phenomenon of imperfect recall in semantic games for IF logic. Interestingly, the perfect recall fragment of IF logic has it that its formulae give rise to imperfect information games, even though they have first-order equivalents. The question arises whether **IF**^{PR} can be given a compositional semantics in terms of single assignments such as Tarski semantics for first-order logic.

In this section I studied the *perfect recall* fragment of IF logic, but other game-theoretic notions can be transferred just as well. In this genre, I mention the notion of positional determinacy that plays a key role in decidability results for modal languages.

3.5 Modal logic and IF logic

Modal logic (Blackburn et al. 2001) originated from analytic philosophy and currently enjoys attention from disciplines including proof theory (Boolos 1993; Joosten 2004), computer science, theoretical linguistics, and game theory, see (van Benthem lecture notes) and (van Benthem 2001; Pauly 2001; Kooi 2003; de Bruin 2004). Basic modal logic has a *computation profile* that is “more attractive” than first-order logic’s: its satisfiability problem is decidable, and its model checking problem is tractable (**P**-computable). Extending basic modal logic’s expressive power whilst preserving these nice computational properties is a lively branch of research; see (ten Cate 2005) for an in-depth study.

Be this as it may, independence or imperfect information never played an important role in this research. In this section, I will explore the imperfect information dimension of modal logic. I define a new IF modal language, that is basically a fragment of Hintikka and Sandu’s IF logic, and I prove this logic undecidable.

In seminal publications (Sandu 1993; Hintikka 1996; Hintikka and Sandu 1997) on IF logic, the applications of slashing in a first-order modal context are mentioned. The issue of informational independence in epistemic first-order logic related to the *de dicto-de re* distinction in linguistics was taken up in (Hintikka 1993; Pietarinen 1998).

The topic of independence-friendliness in modal propositional logics was studied in a series of publications (Bradfield 2000; Bradfield and Fröschle 2002; Tulenheimo 2003; Tulenheimo 2004; Hyttinen and Tulenheimo 2005), that are of a highly explorative nature. That is to say, no two of these publications share the same syntax and/or semantic interpretation (except for (Tulenheimo 2003) and (Tulenheimo 2004), but the latter is based on the former). This raises the question what is *the* independence-friendly modal logic—if any.

In Section 3.5.1, I give a recap of the approach taken in (Tulenheimo 2003; Tulenheimo 2004; Hyttinen and Tulenheimo 2005) toward IF modal logic, and discuss the fact that the involved semantic games violate the action consistency requirement.

In Section 3.5.2, I isolate the modal fragment of $\mathbf{IF}\forall$, by resorting to the first-order correspondence language of $\mathbf{ML}\forall$. Furthermore, being a fragment of $\mathbf{IF}\forall$, their semantic fragments do not violate the action consistency requirement.

In Section 3.5.3, I show that the modal fragment of $\mathbf{IF}\forall$ is undecidable.

3.5.1 Uniformity interpretation for modal logic

In (Tulenheimo 2004) three interpretations for independence-friendly modal languages are studied: the *uniformity*, *backwards-looking*, and *algebraic* interpreta-

tion. Most attention has been with IF modal logic under the uniformity interpretation, cf. (Tulenheimo 2003; Hyttinen and Tulenheimo 2005). Tulenheimo (2004, pg. 15) contemplates on this interpretation as follows: “The uniformity interpretation aims at bringing a very straightforward modal-logical analogue of the IF first-order logic of Hintikka and Sandu: it makes use of semantical games, and implements the notion of independence by imposing appropriate conditions of uniformity on winning strategies.”

Throughout this review I shall assume some familiarity with modal logic’s syntax and semantics. A thorough treatment is supplied in Section 3.5.2 below.

The semantic game for a basic modal logic ϕ on a Kripke-model \mathfrak{M} at w can be seen as a *pebble-moving* game. The game starts from position $\langle \phi, \mathfrak{M}, w \rangle$ indicating that the pebble is at w . If the game reaches position $\langle \langle R_a \rangle \psi, \mathfrak{M}, u \rangle$, Eloise is to move the pebble from u along the accessibility relation R_a to world v . If she cannot push the pebble to a successor state, Eloise gets stuck and loses the game; otherwise, the game advances to position $\langle \psi, \mathfrak{M}, v \rangle$. The game rule for the $[R_a]$ operator is similar, but now Abelard moves. Connectives trigger a choice among the main subformulae by either player as usual. If the game reaches position $\langle (\neg)p, \mathfrak{M}, u \rangle$ it stops and Eloise wins if the world u where the pebble is at makes p true (false); otherwise, Abelard wins.

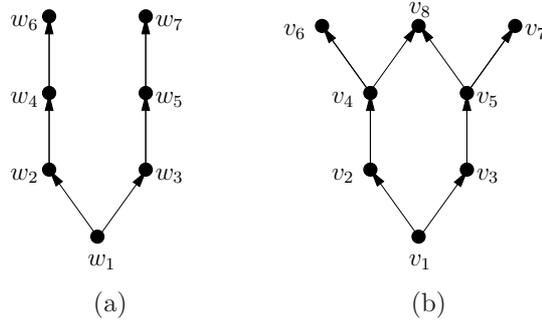
To exhibit the apparatus from (Tulenheimo 2004), let me introduce a toy language that serves this overview’s ends—this language has not been assessed in the literature. Consider the language of *well-organized IF modal logic* consisting of all strings

$$M_1 \dots M_n \psi, \quad (3.6)$$

where ψ is a basic modal formula and M_i is an operator of the form \Box_i or (\Diamond_i/I_i) , where $I_i \subseteq \{1, \dots, i-1\}$. If ϕ is a well-organized IF modal formula, let ϕ^\spadesuit denote the result of replacing every (\Diamond_i/I_i) in ϕ by \Diamond and every \Box_i by \Box . Clearly, ϕ^\spadesuit is a formula from basic modal logic. The turn taking in the semantic evaluation game for a well-organized IF modal logic formula ϕ on \mathfrak{M} at w is similar to the turn taking in the one for ϕ^\spadesuit on \mathfrak{M} at w . Furthermore, Abelard and the Eloises still pick up vertices where to move the pebble, but they put their move in envelopes rather than moving the pebble directly—I will stick to the coalitional interpretation laid out in Section 3.3. In consequence, at some stage of the game an Eloise may be uninformed about the pebble’s current position! For the sake of concreteness consider the formula

$$\phi = \Box_1 \Diamond_2 (\Diamond_3 / \{1, 2\}) \top \quad (3.7)$$

in whose semantic game Abelard kicks off and Eloise₂ and Eloise₃ move during the second and third round, respectively. (The symbol \top denotes the proposition that is true at every world.) Eloise₂ is shown the move previously made by Abelard; Eloise₃ is not shown the contents of any of the envelopes. Truth of a

Figure 3.1: The models \mathfrak{M} and \mathfrak{N} .

well-organized IF modal formula ϕ on a pointed model $\langle \mathfrak{M}, w \rangle$ is defined as the Eloises having a uniform winning strategy in the related semantic game.

Consider the models $\mathfrak{M} = \langle M, R \rangle$ and $\mathfrak{N} = \langle N, S \rangle$ displayed in Figure 3.1.a and 3.1.b respectively, where

$$\begin{aligned}
 M &= \{w_1, \dots, w_7\} \\
 R &= \{\langle w_1, w_2 \rangle, \langle w_1, w_3 \rangle, \langle w_2, w_4 \rangle, \langle w_3, w_5 \rangle, \langle w_4, w_6 \rangle, \langle w_5, w_7 \rangle\} \\
 N &= \{v_1, \dots, v_8\} \\
 S &= \{\langle v_1, v_2 \rangle, \langle v_1, v_3 \rangle, \langle v_2, v_4 \rangle, \langle v_3, v_5 \rangle, \langle v_4, v_6 \rangle, \langle v_5, v_7 \rangle, \langle v_4, v_8 \rangle, \langle v_5, v_8 \rangle\}
 \end{aligned}$$

and focus on \mathfrak{M} and w_1 . On the first round, triggered by \Box_1 , Abelard puts w_i in the appropriate envelope, where $i \in \{2, 3\}$. In order not to lose right way, Eloise₂ puts w_{i+2} in the appropriate envelope. On the third round Eloise₃ remains uninformed about the position of the pebble: w_4 or w_5 ? Whatever world she picks she is not certain to move the pebble there along the accessibility relation from its current location. For instance, suppose the pebble is at w_4 she may try and unsuccessfully move the pebble to w_7 .

By contrast, turn to the pointed model $\langle \mathfrak{N}, v_1 \rangle$, where Eloise₃'s ignorance passes unnoticed; she simply moves the pebble to v_8 which is possible from both v_4 and v_5 .

After properly introducing the semantical apparatus, one concludes on the basis of such game-theoretic considerations that $\mathfrak{M}, w_1 \not\models \phi$, whereas $\mathfrak{N}, v_1 \models \phi$. Note that the pointed models $\langle \mathfrak{M}, w_1 \rangle$ and $\langle \mathfrak{N}, v_1 \rangle$ cannot be distinguished by any formula from basic modal logic.⁵ Hence, any IF modal language that contains basic modal logic and accepts ϕ as a formula, has greater expressive power than basic modal logic, given that its formulae are evaluated as above.

⁵The pointed models $\langle \mathfrak{M}, w_1 \rangle$ and $\langle \mathfrak{N}, v_1 \rangle$ are bisimilar, in virtue of the following *bisimilarity relation*: $\{\langle w_i, v_i \rangle \mid 1 \leq i \leq 7\} \cup \{\langle w_6, v_8 \rangle, \langle w_7, v_8 \rangle\}$. For a rigorous treatment of the notion of bisimulation and its impact on the theory of modal logic, consult (van Benthem 1976; Blackburn, de Rijke, and Venema 2001).

The framework laid down in (Tulenheimo 2004) may be considered a proof of principle: if one extends basic modal logic with a formula like $\Box_1 \Diamond_2 (\Diamond_3 / \{1, 2\}) \top$ the resulting logic's expressive power increases, under an interpretation of informational independence that is inspired by the one from IF logic.

From a game-theoretic point of view, the semantic games for well-organized IF modal logic are ill-defined objects. Namely, they do not meet the action consistency requirement. That is, there are semantic games for well-organized IF modal logic, for which there exists one information partition I that contains two histories h, h' such that $A(h) \neq A(h')$. A case in point is the game constituted by the pointed model $\langle \mathfrak{M}, w_1 \rangle$ from Figure 3.1 and the formula ϕ . During the third round of this semantic game, Eloise₃ controls one information partition: $\{h, h'\}$, where $h = \langle w_1, w_2, w_4 \rangle$ and $h' = \langle w_1, w_3, w_5 \rangle$. Although h and h' sit in the same information partition, it is the case that $A(h) = \{w_6\} \neq \{w_7\} = A(h')$. Notably, the authors are well-aware of this point, cf. (Tulenheimo 2004, Section 2.3.1).

It was observed (Hodges 2001, pg. 546) in the context of restricted quantifiers, that things that are obviously equivalent in first-order logic split apart when one inserts the slash device. In (Tulenheimo 2004) the slash device is introduced in such a way that the resulting logic called **EIFML** has amongst its formulae the string

$$\bigwedge_{i \in \{a, b\}} (\langle R_i \rangle_1 / \{i\}) \top. \quad (3.8)$$

The semantic game of (3.8) has two rounds, during the first of which Abelard picks up a modality $i \in \{a, b\}$ and puts it in an envelope. On the second round Eloise₁ has to pick a successor to the current world along the accessibility relation R_i , but i is unbeknownst to her. So Eloise₁ wins the game on \mathfrak{M} at w , if there exists a world v , that is both R_a and R_b -accessible from w ; or equivalently—in *Propositional Dynamic Logic* notation—

$$\mathfrak{M}, w \models \langle R_a \cap R_b \rangle \top.$$

In (Tulenheimo 2004, Lemma 3.3.8) it is shown that **EIFML** is not translatable into first-order logic, if three or more modality types are involved.

In (Hyttinen and Tulenheimo 2005) yet another IF modal logic is considered. In principle this language can be regarded the result of applying an IF procedure to the syntax of basic modal logic. This IF procedure would now apply to numerals that identify the modal operators rather than variables, as is the case in IF logic. In order to establish decidability for this language, the authors impose three conditions on the formulae at stake and denote the result by **IFML^{PR}**. These three conditions are syntactical, and are associated with perfect recall.

3.5.2 The modal fragment of IF logic

All in all, various IF modal languages have been proposed each with its own set of particulars: restricted quantifiers versus infix connectives; quantification over modalities; arbitrary independence schemes versus perfect recall ones. In this section, I will introduce yet another IF modal logic. More in particular, this IF modal logic will be defined as a fragment of IF logic. This approach is inspired by current research on modal logic. Because, although notationally basic modal logic is an extension of propositional logic, nowadays it is usually conceived of as a fragment of first-order logic. Milestone results that brought about this change include the standard translation and van Benthem's Theorem (van Benthem 1976), that characterizes modal logic as the *bisimulation invariant* fragment of first-order logic.

Before I come to defining the modal fragment of $\mathbf{IF}\forall$, I give the syntax and semantics of basic modal logic, extended with restricted quantifiers.

3.5.1. DEFINITION. Let $\pi = \{p_1, p_2, \dots\}$ be a set of proposition letters and let $\mu = \{R_1, R_2, \dots\}$ be a set of modalities. Associate with μ the set of tokens $\text{Token}(\mu)$, such that for every $t \in \text{Token}(\mu)$, R_t identifies a unique modality in μ . Let IND be the set of indices as before. Then, the formulae of the *basic modal logic with restricted quantifiers*, denoted by $\mathbf{ML}\forall(\pi, \mu)$, are exactly those strings that are generated by the following grammar:

$$\phi ::= p \mid \neg p \mid \phi \vee \phi \mid \phi \wedge \phi \mid \langle S \rangle \phi \mid [S] \phi \mid \langle R_i \rangle \phi \mid [R_i] \phi \mid \bigvee_{i \in I} \phi \mid \bigwedge_{i \in I} \phi,$$

where p ranges over π , S over μ , i over IND, and I over the finite subsets of $\text{Token}(\mu)$.

Note that all $\mathbf{ML}(\pi, \mu)$ -formulae are in negation normal form, in that negation symbols only occur in front of proposition letters. As usual, I will sometimes omit to mention the sets π and μ , and simply write $\mathbf{ML}\forall$. By saying that ϕ is an $\mathbf{ML}\forall$ -formula, I mean that there are π and μ such that ϕ is an $\mathbf{ML}\forall(\pi, \mu)$ -formula.

Indices are the only items that are being quantified over in basic modal logic with restricted quantifiers. Like variables in first-order logic, indices in $\mathbf{ML}\forall$ may appear bound or free. If an $\mathbf{ML}\forall$ -formula contains no free indices, it is called an $\mathbf{ML}\forall$ -*sentence*. Henceforth all discussion will be restricted to $\mathbf{ML}\forall$ -sentences, even if I refer to them as formulae.

3.5.2. DEFINITION. Let $\pi = \{p_1, p_2, \dots\}$ be a set of proposition letters and let $\mu = \{R_1, R_2, \dots\}$ be a set of modalities. Then, let $\mathfrak{M} = \langle M, \langle S^{\mathfrak{M}} \rangle_{S \in \mu}, V \rangle$ be a π, μ -*model*, if

$$\begin{aligned} S^{\mathfrak{M}} &\subseteq M \times M \\ V(p) &\subseteq M, \end{aligned}$$

where S ranges over μ and p over π . For $w \in M$, $\langle \mathfrak{M}, w \rangle$ is a *pointed* π, μ -model. I will sometimes omit the brackets enclosing pointed models. A function of type $\text{IND} \rightarrow \text{Token}(\mu)$ is an assignment in \mathfrak{M} .

The satisfaction relation of $\mathbf{MLV}(\pi, \mu)$ is defined for $\mathbf{MLV}(\pi, \mu)$ -formulae with respect to pointed π, μ -models $\langle \mathfrak{M}, w \rangle$ and assignments α in \mathfrak{M} :

$$\begin{aligned}
\mathfrak{M}, w \models p[\alpha] & \text{ iff } w \in V(p), \quad \text{for } p \in \pi \\
\mathfrak{M}, w \models (\neg p)[\alpha] & \text{ iff } w \notin V(p), \quad \text{for } p \in \pi \\
\mathfrak{M}, w \models (\phi \vee \psi)[\alpha] & \text{ iff } \mathfrak{M}, w \models \phi[\alpha] \text{ or } \mathfrak{M}, w \models \psi[\alpha] \\
\mathfrak{M}, w \models (\phi \wedge \psi)[\alpha] & \text{ iff } \mathfrak{M}, w \models \phi[\alpha] \text{ and } \mathfrak{M}, w \models \psi[\alpha] \\
\mathfrak{M}, w \models (\langle S \rangle \phi)[\alpha] & \text{ iff for some } v \in M \text{ with } \langle w, v \rangle \in S^{\mathfrak{M}}, \mathfrak{M}, v \models \phi[\alpha] \\
\mathfrak{M}, w \models ([S] \phi)[\alpha] & \text{ iff for all } v \in M \text{ with } \langle w, v \rangle \in S^{\mathfrak{M}}, \mathfrak{M}, v \models \phi[\alpha] \\
\mathfrak{M}, w \models (\langle R_i \rangle \phi)[\alpha] & \text{ iff for some } v \in M \text{ with } \langle w, v \rangle \in R_{\alpha(i)}^{\mathfrak{M}}, \mathfrak{M}, v \models \phi[\alpha] \\
\mathfrak{M}, w \models ([R_i] \phi)[\alpha] & \text{ iff for all } v \in M \text{ with } \langle w, v \rangle \in R_{\alpha(i)}^{\mathfrak{M}}, \mathfrak{M}, v \models \phi[\alpha] \\
\mathfrak{M}, w \models \bigvee_{i \in I} \phi[\alpha] & \text{ iff for some } j \in I, \mathfrak{M}, w \models \phi[\alpha.i/j] \\
\mathfrak{M}, w \models \bigwedge_{i \in I} \phi[\alpha] & \text{ iff for all } j \in I, \mathfrak{M}, w \models \phi[\alpha.i/j].
\end{aligned}$$

If $\mathfrak{M}, w \models \phi[\alpha]$, I say that ϕ is true on \mathfrak{M} at w under α . If ϕ is in fact a sentence, the phrase “under α ” can be dropped harmlessly.

Observe that in \mathbf{MLV} the choosing of a modality and moving along this modality are separated. For instance, in a semantic game for $\bigvee_{i \in \{R_a, R_b\}} [R_i]p$, Eloise first chooses either R_a or R_b , after which Abelard points out a successor along the chosen accessibility relation.

Showing that for every \mathbf{MLV} -formula there is an equivalent \mathbf{MLV} -formula *without* restricted quantifiers is an easy exercise, to which end one first has to define the proper notion of equivalence for \mathbf{MLV} . Furthermore, it is well-known that basic modal logic is translatable into first-order logic. The witness of this result is commonly referred to as the *standard translation*. Before I get to the standard translation, a word on vocabularies. Let $\pi = \{p_1, \dots, p_m\}$ be a finite set of proposition letters and let $\mu = \{R_1, \dots, R_n\}$ be a finite set of modalities. Then, $\tau(\pi, \mu)$ is defined as the *vocabulary constituted by π and μ* , and is defined as

$$\left(\bigcup_{p \in \pi} \{P_p\} \right) \cup \mu,$$

where the symbols P_p are unary relation symbols and the symbols inherited from μ are binary.

Then, the standard translation ST maps $\mathbf{MLV}(\pi, \mu)$ to $\mathbf{FOV}(\tau(\pi, \mu))$, as

follows:

$$\begin{aligned}
ST_{x_t}(p) &= P_p(x_t), \quad \text{for } p \in \pi \\
ST_{x_t}(\neg p) &= \neg P_p(x_t), \quad \text{for } p \in \pi \\
ST_{x_t}(\phi \vee \psi) &= ST_{x_t}(\phi) \vee ST_{x_t}(\psi) \\
ST_{x_t}(\phi \wedge \psi) &= ST_{x_t}(\phi) \wedge ST_{x_t}(\psi) \\
ST_{x_t}(\langle R_a \rangle \phi) &= \exists x_{t+1} (R_a(x_t, x_{t+1}) \wedge ST_{x_{t+1}}(\phi)), \quad \text{for } R_a \in \mu \\
ST_{x_t}([R_a] \phi) &= \forall x_{t+1} (R_a(x_t, x_{t+1}) \rightarrow ST_{x_{t+1}}(\phi)), \quad \text{for } R_a \in \mu \\
ST_{x_t}(\langle R_i \rangle \phi) &= \exists x_{t+1} (R_i(x_t, x_{t+1}) \wedge ST_{x_{t+1}}(\phi)), \quad \text{for } i \in \text{IND} \\
ST_{x_t}([R_i] \phi) &= \forall x_{t+1} (R_i(x_t, x_{t+1}) \rightarrow ST_{x_{t+1}}(\phi)), \quad \text{for } i \in \text{IND} \\
ST_{x_t} \left(\bigvee_{i \in I} \phi \right) &= \bigvee_{i \in I} ST_{x_t}(\phi) \\
ST_{x_t} \left(\bigwedge_{i \in I} \phi \right) &= \bigwedge_{i \in I} ST_{x_t}(\phi).
\end{aligned}$$

3.5.3. EXAMPLE. (Continuation of Example 3.2.4) Consider the formula $\xi = [R][R]\langle R \rangle \top$, whose standard translation $ST_{x_1}(\xi)$ is

$$\forall x_2 (R(x_1, x_2) \rightarrow \forall x_3 (R(x_2, x_3) \rightarrow \exists x_4 (R(x_3, x_4))))).$$

This formula was already focus of attention in Example 3.2.4, be it that I did not use the shorthand notation “ \rightarrow ”. \square

Let \mathfrak{M} be a π, μ -model. Then,

$$\mathfrak{A}_{\mathfrak{M}} = \langle M, \langle V(p) \rangle_{p \in \pi}, \langle R^{\mathfrak{M}} \rangle_{R \in \mu} \rangle$$

is the $\tau(\pi, \mu)$ -structure that is said to be *constituted* by \mathfrak{M} . Adequacy of the standard translation is thus cast as follows: For every $\mathbf{ML}\forall(\pi, \mu)$ -sentence ϕ and every pointed π, μ -model $\langle \mathfrak{M}, w \rangle$, it is the case that

$$\mathfrak{M}, w \models \phi \quad \text{iff} \quad \mathfrak{A}_{\mathfrak{M}} \models ST_{x_1}(\phi)[x_1/w].$$

Put $ST'(\mathbf{ML}\forall(\pi, \mu)) = \bigcup_{\phi} ST_{x_1}(\phi)$, where ϕ ranges over all $\mathbf{ML}\forall(\pi, \mu)$ -sentences, and call this fragment of $\mathbf{FO}\forall(\tau(\mu, \pi))$ the *first-order correspondence language* of $\mathbf{ML}\forall(\mu, \pi)$. I will obtain the modal fragment of IF logic by applying the IF procedure from Section 3.2 to the first-order correspondence language of $\mathbf{ML}\forall$. In order to do so, it is to be observed that we are considering a fragment of $\mathbf{FO}\forall'$ rather than $\mathbf{FO}\forall$. That is, the first-order correspondence language of $\mathbf{ML}\forall$ has it that in all of its formulae

- (1) every variable is quantified at most once; and

(2) negations appear only in front of atoms.

The language $ST'(\mathbf{MLV}(\pi, \mu))$ does not meet condition (1), in view of, for instance, the following formula:

$$ST_{x_1}(\langle R \rangle p \wedge \langle R \rangle q) = (\exists x_2 (R(x_1, x_2) \wedge P_p(x_2))) \wedge (\exists x_2 (R(x_1, x_2) \wedge P_q(x_2))).$$

However, an elementary renaming argument shows that every $ST'(\mathbf{MLV}(\pi, \mu))$ -formula Φ can be rewritten so as to meet (1), without changing its truth conditions. Let $ST(\mathbf{MLV}(\pi, \mu))$ be the language that contains all $ST'(\mathbf{MLV}(\pi, \mu))$ -formulae whose variables are thus renamed.

To see that every $ST(\mathbf{MLV}(\pi, \mu))$ -formula satisfies condition (2), recall that every $\mathbf{MLV}(\pi, \mu)$ -formula is in negation normal form. Consequently, if $\neg\phi$ is a subformula of any $\mathbf{MLV}(\pi, \mu)$ -formula, then for some $p \in \pi$, $\phi = p$. Since $ST_{x_t}(\neg p) = \neg P_p(x_t)$, every negation in a $\mathbf{MLV}(\pi, \mu)$ -formula finds itself translated in front of an atom. Covertly, the rules $ST_{x_t}([R_a]\phi)$ and $ST_{x_t}([R_i]\phi)$ introduce negations, since $\Phi \rightarrow \Psi$ is shorthand for $\neg\Phi \vee \Psi$. Those rules cause no problematic negations though, since every $\neg\Phi$ thus introduced is of the form $\neg R_a(\dots)$ or $\neg R_i(\dots)$.

Hence, $ST(\mathbf{MLV}(\pi, \mu))$ is a fragment of $\mathbf{FOV}'(\tau(\pi, \mu))$. As such the language is perfectly open to application of the IF procedure constituted by (IF). This insight underlies the following definition.

3.5.4. DEFINITION. Let $ST(\mathbf{MLV}(\pi, \mu))$ be the fragment of $\mathbf{FOV}'(\tau(\pi, \mu))$ as above. Then, define $\mathbf{IFV}^{\mathbf{ML}}(\tau(\pi, \mu))$ as the closure of $ST(\mathbf{MLV}(\pi, \mu))$ under the IF procedure constituted by (IF), from Definition 3.2.2 on page 28.

Being a fragment of \mathbf{IFV} , the logic $\mathbf{IFV}^{\mathbf{ML}}$ may fully rely on the formal apparatus set out for the former logic. In particular, if Φ is an $\mathbf{IFV}^{\mathbf{ML}}$ -formula then its Skolemization $Sk(\Phi)$ is readily defined. Observe that in every $\mathbf{IFV}^{\mathbf{ML}}$ -formula Φ the variable x_1 is free. From the definition of the Skolemization procedure it follows that also x_1 is the free variable in $Sk(\Phi)$. Therefore, if \mathfrak{A} is a suitable structure and a is an object from A , then one directly has that $\mathfrak{A} \models Sk(\Phi)[x_1/a]$ iff the Eloises have a winning strategy in $Sem\text{-}game_{Coal}^{\mathbf{IF}}(\Phi[x_1/a], \mathfrak{A})$.

By setting up an IF modal logic along these lines, the property of the semantic games from (Tulenheimo 2003; Tulenheimo 2004; Hyttinen and Tulenheimo 2005) games violating the action consistency requirement for imperfect information games is resolved. Recall namely that semantic games for IF logic do meet this requirement.

3.5.5. EXAMPLE. (Continuation of Examples 2.3.1, 3.2.3, and 3.2.5) Consider the $\mathbf{MLV}(\emptyset, \{R_a, R_b\})$ -formula $\psi = \bigwedge_{i \in \{a, b\}} \langle R_i \rangle \top$.⁶ The standard translation

⁶Strictly speaking the symbol “ \top ” is not included in the language \mathbf{MLV} . Throughout the examples it can be safely replaced by “ $(p \vee \neg p)$ ”.

$ST_{x_1}(\psi)$ is

$$\bigwedge_{i \in \{a,b\}} \exists x_2 R(x_1, x_2),$$

which is the \mathbf{FOV}' -formula from which Example 2.3.1 departs. Conclude from Example 3.2.5 that also $\bigwedge_{i \in \{a,b\}} (\exists x_2 / \{i\}) R(x_1, x_2)$ is an $\mathbf{IFV}^{\mathbf{ML}}$ -formula. \square

The logics in (Tulenheimo 2003; Tulenheimo 2004; Hyttinen and Tulenheimo 2005) were defined in a modal-style notation and came with a Kripke-style semantics. By contrast, I obtained the logic $\mathbf{IFV}^{\mathbf{ML}}$ by isolating a fragment of \mathbf{IFV} , relying on first-order structures. This discrepancy does not imply that the latter languages are strictly incomparable to the ones introduced in the above publications. For instance, revisit the well-organized IF modal logic consisting of all strings

$$M_1 \dots M_n \psi, \tag{3.9}$$

where ψ is a basic modal formula and M_i is an operator of the form \Box_i or (\Diamond_i / I_i) , where $I_i \subseteq \{1, \dots, i-1\}$. For now, extend the standard translation ST in such a way that it maps a well-organized IF modal formula ϕ as in (3.9) to the following $\mathbf{IFV}^{\mathbf{ML}}$ -formula:

$$Q_1 x_2 (R(x_1, x_2) \circ_1 \dots Q_n x_{n+1} (R(x_n) \circ_n ST_{x_n}(\psi)) \dots),$$

where

$$Q_i x_{i+1} = \begin{cases} \forall x_{i+1} & \text{if } M_i \text{ is } \Box \\ (\exists x_{i+1} / \{x_j \mid j \in I_i\}) & \text{if } M_i \text{ is } (\Diamond_i / I_i) \end{cases}$$

and

$$\circ_i = \begin{cases} \rightarrow & \text{if } M_i \text{ is } \Box \\ \wedge & \text{if } M_i \text{ is } (\Diamond_i / \dots). \end{cases}$$

In line with (Tulenheimo 2004, Lemma 3.2.2) one proves that for every well-organized IF modal formula ϕ and suitable pointed model $\langle \mathfrak{M}, w \rangle$, it is the case that

$$\mathfrak{M}, w \models \phi \text{ iff } \mathfrak{A}_{\mathfrak{M}} \models ST_{x_1}(\phi)[x_1/w].$$

Along the same lines, one proves that the language from (Tulenheimo 2003) is translatable into $\mathbf{IFV}^{\mathbf{ML}}$, and that $\mathbf{IFV}^{\mathbf{ML}}$ is translatable into the language \mathbf{EIFML} from (Tulenheimo 2004). I will comment on the perfect recall language $\mathbf{IFML}^{\mathbf{PR}}$ from (Hyttinen and Tulenheimo 2005) and its relation to $\mathbf{IFV}^{\mathbf{ML}}$ but not until Section 3.6.

Let me close this section with some results that chart $\mathbf{IFV}^{\mathbf{ML}}$'s expressive power.

3.5.6. PROPOSITION. *The following hold:*

$$(1) ST(\mathbf{MLV}) < \mathbf{IFV}^{\mathbf{ML}}.$$

$$(2) \mathbf{IFV}^{\mathbf{ML}} \not\leq \mathbf{FOV}.$$

Proof. Ad (1): By construction, every $ST(\mathbf{MLV})$ -formula is an $\mathbf{IFV}^{\mathbf{ML}}$ -formula. This settles the inclusion. As for strictness, observe that $\mathbf{IFV}^{\mathbf{ML}}$ contains the formula

$$\forall x_2 (R(x_1, x_2) \rightarrow \forall x_3 (R(x_2, x_3) \rightarrow (\exists x_4 / \{x_2, x_3\})(R(x_3, x_4))))), \quad (3.10)$$

in virtue of Examples 3.2.4 and 3.5.3. Observe that the extended standard translation maps $\Box_1 \Box_2 (\Diamond_3 / \{1, 2\}) \top$ on (3.10), and recall that the formula ϕ was earlier seen to discriminate bisimilar pointed models, amongst which the ones in Figure 3.1. The result follows since no formula in \mathbf{MLV} can tell bisimilar pointed models apart, see (Blackburn et al. 2001).

Ad (2): The proof uses a straightforward model comparison game argument, see also Section 4.8.1. For further details I refer the reader to (Tulenheimo and Sevenster forthcoming). \square

3.5.3 $\mathbf{IFV}^{\mathbf{ML}}$ is undecidable

In this section I prove satisfiability undecidable for $\mathbf{IFV}^{\mathbf{ML}}$, by showing that there exists a formula in this language that can “tile the plane”.

The following easy translation scheme will be used.

3.5.7. PROPOSITION. *Let $\Phi \in \mathbf{IFV}^{\mathbf{ML}}$ contain $(\Psi \vee (\exists x/X)\Psi')$ as a subformula, where Ψ is a possibly negated atom and does not contain the variable x . Let Φ' denote the result of replacing $(\Psi \vee (\exists x/X)\Psi')$ in Φ by $(\exists x/X)(\Psi \vee \Psi')$. Then, Φ and Φ' are equivalent.*

Recall the problem of **TILING**. Let a finite set T of *tiles* be given; a tile t being a 1×1 square, fixed in orientation, each side of which has a color $right(t)$, $left(t)$, $up(t)$, and $down(t)$. The tiling problem is defined such that $T \in \mathbf{TILING}$ iff one can cover every node in the $\mathbb{N} \times \mathbb{N}$ grid with tiles from T in such a way that adjacent—horizontally and vertically—tiles have the same color on the common edge. A composition of tiles that covers the $\mathbb{N} \times \mathbb{N}$ grid is called a *tiling*.

TILING is Π_1^0 -complete, cf. (Harel 1985; van Emde Boas 1996). (Note that the non-boldfaced Π_1^0 and Σ_1^1 do not indicate logical languages but complexity measures in the arithmetical and analytical hierarchy, respectively.) In Theorem 3.5.8 it is shown that **TILING** reduces to the satisfiability problem of $\mathbf{IFV}^{\mathbf{ML}}$,

which renders it Π_1^0 -hard, hence undecidable. Since $\mathbf{IFV}^{\mathbf{ML}}$ is a fragment of \mathbf{IFV} , the upper bound of $\mathbf{IFV}^{\mathbf{ML}}$'s complexity is Σ_1^1 . The precise classification of the complexity of $\mathbf{IFV}^{\mathbf{ML}}$'s satisfiability problem is left as an open problem.

3.5.8. THEOREM. *Let T be a finite set of tiles. Let π contain the letters p and q plus the proposition letter p_t for every tile $t \in T$, and let $\mu = \{R_a, R_b, R_{\uparrow}, R_{\Rightarrow}, =\}$. Let $\tau = \tau(\pi, \mu)$. Then, there exists an $\mathbf{IFV}^{\mathbf{ML}}(\tau)$ -formula Φ_T , such that Φ_T is satisfiable iff T can tile the $\mathbb{N} \times \mathbb{N}$ grid. Hence, the satisfiability problem of $\mathbf{IFV}^{\mathbf{ML}}(\tau)$ is undecidable.*

The proof of the theorem is rather lengthy for which reason I will first state its underlying idea. The formula Φ_T is an implementation of the *spy point technique* that is used to prove undecidability for hybrid modal languages, see (Blackburn and Seligman 1995). In order to prove undecidability for a logic by means of the spy point technique one typically shows that the logic at hand can enforce the following properties on a structure \mathfrak{A} with respect to the object w :

- (1) If there is a path of $R_{\uparrow}^{\mathfrak{A}} \cup R_{\Rightarrow}^{\mathfrak{A}}$ -transitions from w to v , then $\langle w, v \rangle \in R_a^{\mathfrak{A}}$.
- (2) The relation $R_{\uparrow}^{\mathfrak{A}} \cup R_{\Rightarrow}^{\mathfrak{A}}$ constitutes a grid structure containing w .
- (3) The predicates $\langle P_t^{\mathfrak{A}} \rangle_{t \in T}$ describe a proper tiling of the objects in the grid.

Having observed that the logic under consideration can express (1)-(3), it suffices to observe that there is a way of enforcing that any two neighboring objects in the grid (hence R_a -accessible from w) carry appropriate tiles.

I will pursue this line of attack also in the proof of Theorem 3.5.8, by showing that conditions (1)-(3) are expressible in $\mathbf{IFV}^{\mathbf{ML}}$.

That condition (3) can be expressed in $\mathbf{IFV}^{\mathbf{ML}}$ is no big surprise, since it can be expressed in basic modal logic. As for condition (2) the hard part is to express the existence of a common successor. This cannot be done in basic modal logic, but it can be done in $\mathbf{IFV}^{\mathbf{ML}}$. For the sake of illustration, consider the $\mathbf{IFV}^{\mathbf{ML}}$ -formula

$$\forall x_2 (R(x_1, x_2) \rightarrow (\exists x_3 / \{x_2\}) (R(x_2, x_3)))$$

which is equivalent to

$$\forall x_2 (\neg R(x_1, x_2) \vee (\exists x_3 / \{x_2\}) (R(x_2, x_3))).$$

Because of Propositions 3.4.9 and 3.5.7 the latter formula is equivalent to

$$\exists x_3 \forall x_2 (R(x_1, x_2) \rightarrow R(x_2, x_3)),$$

expressing that all R -successors of x_1 have a common R -successor x_3 . This example does not provide direct evidence of the claim that $\mathbf{IFV}^{\mathbf{ML}}$ can express

condition (2), but it may strike the reader more plausible now. Showing that $\mathbf{IFV}^{\mathbf{ML}}$ can express condition (3) is less perspicuous, and may thus be regarded the main achievement of the proof. It is established using the relation R_b , which intuitively is such that from every object v that is R_a -accessible from the spy point w , one can return to w via one R_b -transition.

Before I get to the proof of Theorem 3.5.8, let me first go through the meaning of a series of $\mathbf{IFV}^{\mathbf{ML}}(\tau)$ -formulae, whose conjunction constitutes Φ_T .

To enhance readability of the formulae appearing in the upcoming claims, I start out from the formulae in modal-style notation. Generally speaking, they are somewhat shorter and may please the reader who is familiar with modal languages. Note that this language is used merely as shorthand notation. The “non-shorthand” $\mathbf{IFV}^{\mathbf{ML}}$ -formula it conveys, is invariably given in the first line of the proof of every claim (possibly modulo some truth preserving simplifications). I shall write $\langle a \rangle$ and $[a]$ instead of $\langle R_a \rangle$ and $[R_a]$, respectively; and while working with first-order syntax I shall write “ $x = y$ ” rather than “ $R_=(x, y)$ ”. Furthermore, the predicates P_p , P_q , and P_{p_t} , for $t \in T$ shall be written as P , Q , and P_t , respectively. Every $\mathbf{IFV}^{\mathbf{ML}}$ -formula’s free variable shall be x_1 . Throughout the claims, the object assigned to x_1 will be the object w .

Claim. $\mathfrak{A} \models (3.11)[x_1/w]$ iff $\langle w, w \rangle \in R_a^{\mathfrak{A}}$ and $\langle w, w \rangle \in R_b^{\mathfrak{A}}$;

$$\bigwedge_{i \in \{a, b, =\}} (\langle i \rangle / \{i\}) \top. \quad (3.11)$$

Proof. Consider the following equivalences:

$$\begin{aligned} (3.11) \quad & \text{iff} \quad \bigwedge_{i \in \{a, b, =\}} (\exists x / \{i\}) (R_i(w, x)) \\ (\star) \quad & \text{iff} \quad \exists x \bigwedge_{i \in \{a, b, =\}} (R_i(w, x)) \\ & \text{iff} \quad \exists x (R_a(w, x) \wedge R_b(w, x) \wedge w = x) \\ & \text{iff} \quad R_a(w, w) \wedge R_b(w, w). \end{aligned}$$

Ad (\star) : Just as with $\forall x(\exists y/\{x\})$ one may replace $\bigwedge_{i \in I}(\exists y/\{i\})$ by $\exists y \bigwedge_{i \in I}$ in an $\mathbf{IFV}^{\mathbf{ML}}$ -formula without affecting its truth conditions. See also Proposition 3.4.9. \square

I will not mention the soundness of swapping quantifiers in subsequent claims.

Claim. $\mathfrak{A} \models (3.12)[x_1/w]$ iff if for some v , $\langle w, v \rangle \in R_b^{\mathfrak{A}}$, then $w = v$;

$$\bigwedge_{i \in \{b, =\}} [i]_1 (\langle = \rangle / \{i, 1\}) \top \quad (3.12)$$

Proof. Consider the following equivalences:

$$\begin{aligned}
(3.12) \quad & \text{iff } \bigwedge_{i \in \{b, =\}} \forall x (R_i(w, x) \rightarrow (\exists y / \{i, x\})(x = y)) \\
(\star) \quad & \text{iff } \bigwedge_{i \in \{b, =\}} \forall x (\exists y / \{i, x\}) (R_i(w, x) \rightarrow (x = y)) \\
& \text{iff } \bigwedge_{i \in \{b, =\}} (\exists y / \{i\}) \forall x (R_i(w, x) \rightarrow (x = y)) \\
& \text{iff } \exists y \bigwedge_{i \in \{b, =\}} \forall x (R_i(w, x) \rightarrow x = y) \\
& \text{iff } \exists y (\forall x (R_b(w, x) \rightarrow x = y) \wedge \forall x (w = x \rightarrow x = y)) \\
(\circ) \quad & \text{iff } \exists y (\forall x (R_b(w, x) \rightarrow x = y) \wedge (w = y)) \\
& \text{iff } \forall x (R_b(w, x) \rightarrow x = w).
\end{aligned}$$

Ad (\star) : In virtue of Proposition 3.5.7.

Ad (\circ) : In first-order logic $\forall x (w = x \rightarrow x = y)$ is equivalent to $(w = y)$. \square

Claim. $\mathfrak{A} \models (3.13)[x_1/w]$ iff for every v reachable from w through one R_a -transition followed by one R_Δ -transition it is the case that $\langle v, v \rangle \in R_a^{\mathfrak{A}}$, and there is no other u such that $\langle v, u \rangle \in R_a^{\mathfrak{A}}$, for $\Delta \in \{\uparrow, \Rightarrow\}$;

$$[a][\Delta] \left(\bigwedge_{i \in \{a, =\}} (\langle i \rangle / \{i\}) \top \wedge \bigwedge_{i \in \{a, =\}} [i]_1 (\langle = \rangle / \{i, 1\}) \top \right). \quad (3.13)$$

Proof. I prove for Δ equal to \uparrow . Consider the following equivalences:

$$\begin{aligned}
(3.13) \quad & \text{iff } \forall x (R_a(w, x) \rightarrow \forall y (R_\uparrow(x, y) \rightarrow \Psi)), \text{ where} \\
\Psi & = \exists u (R_a(y, u) \wedge y = u) \wedge \bigwedge_{i \in \{a, =\}} \forall z (R_i(y, z) \rightarrow (\exists u / \{i, z\}) (u = z)) \\
& \text{iff } R_a(y, y) \wedge \exists u \bigwedge_{i \in \{a, =\}} \forall z (R_i(y, z) \rightarrow (u = z)) \\
& \text{iff } R_a(y, y) \wedge \exists u (\forall z (R_a(y, z) \rightarrow z = u) \wedge \forall z (y = z \rightarrow z = u)) \\
& \text{iff } R_a(y, y) \wedge \exists u (\forall z (R_a(y, z) \rightarrow z = u) \wedge y = u) \\
& \text{iff } R_a(y, y) \wedge \forall z (R_a(y, z) \rightarrow z = y).
\end{aligned}$$

\square

The following claims are readily observed, without using any non-standard notation.

Claim. $\mathfrak{A} \models (3.14)[x_1/w]$ iff for every v such that $\langle w, v \rangle \in R_a^{\mathfrak{A}}$ there exists at least one u with $\langle v, u \rangle \in R_{\uparrow}^{\mathfrak{A}}$, where u disagrees with v on p but agrees with v on q ;

$$[a]((p \wedge q) \rightarrow \langle \uparrow \rangle (\neg p \wedge q) \wedge (\neg p \wedge q) \rightarrow \langle \uparrow \rangle (p \wedge q) \wedge (p \wedge \neg q) \rightarrow \langle \uparrow \rangle (\neg p \wedge \neg q) \wedge (\neg p \wedge \neg q) \rightarrow \langle \uparrow \rangle (p \wedge \neg q)). \quad (3.14)$$

Claim. $\mathfrak{A} \models (3.15)[x_1/w]$ iff for every v such that $\langle w, v \rangle \in R_a^{\mathfrak{A}}$ there exists at least one u with $\langle v, u \rangle \in R_{\Rightarrow}^{\mathfrak{A}}$, where u disagrees with v on q but agrees with v on p ;

$$[a]((p \wedge q) \rightarrow \langle \Rightarrow \rangle (p \wedge \neg q) \wedge (\neg p \wedge q) \rightarrow \langle \Rightarrow \rangle (\neg p \wedge \neg q) \wedge (p \wedge \neg q) \rightarrow \langle \Rightarrow \rangle (p \wedge q) \wedge (\neg p \wedge \neg q) \rightarrow \langle \Rightarrow \rangle (\neg p \wedge q)). \quad (3.15)$$

Claim. $\mathfrak{A} \models (3.16)[x_1/w]$ iff for every v such that $\langle w, v \rangle \in R_a^{\mathfrak{A}}$ there exists at most one u with $\langle v, u \rangle \in R_{\Delta}^{\mathfrak{A}}$, for $\Delta \in \{\uparrow, \Rightarrow\}$;

$$[a][\Delta]_1(\langle \Rightarrow \rangle / \{1\})\top. \quad (3.16)$$

Proof. Similar to the proof of (3.12). \square

The following claim establishes that if v is R_a -accessible from w , then w is R_b -accessible from v .

Claim. Let $\mathfrak{A} \models (3.11) \wedge (3.12)[x_1/w]$. Then, $\mathfrak{A} \models (3.17)[x_1/w]$ iff for every v and u , such that $\langle w, v \rangle \in R_a^{\mathfrak{A}}$ and $\langle v, u \rangle \in R_{\Delta}^{\mathfrak{A}}$, $\langle v, w \rangle \in R_b^{\mathfrak{A}}$ and $\langle u, w \rangle \in R_b^{\mathfrak{A}}$, for $\Delta \in \{\uparrow, \Rightarrow\}$;

$$[a]_1 \bigwedge_{i \in \{=, \Delta\}} [i]_2(\langle b \rangle / \{1, i, 2\})\top \quad (3.17)$$

Proof. I prove for Δ equal to \uparrow . Consider the following equivalences:

$$\begin{aligned} (3.17) \text{ iff } & \exists z \forall x \left(R_a(w, x) \rightarrow \bigwedge_{i \in \{\uparrow, =\}} \forall y (R_i(x, y) \rightarrow R_b(y, z)) \right) \\ & \text{iff } \exists z \forall x (R_a(w, x) \rightarrow (\forall y (R_{\uparrow}(x, y) \rightarrow R_b(y, z)) \wedge \forall y (x=y \rightarrow R_b(y, z)))) \\ & \text{iff } \exists z \forall x (R_a(w, x) \rightarrow (\forall y (R_{\uparrow}(x, y) \rightarrow R_b(y, z)) \wedge R_b(x, z))) \\ (\star) \text{ iff } & \exists z \forall x ((w = x \vee R_a(w, x)) \rightarrow (\forall y (R_{\uparrow}(x, y) \rightarrow R_b(y, z)) \wedge R_b(x, z))) \\ (\circ) \text{ iff } & \exists z \forall x (w = x \rightarrow (\forall y (R_{\uparrow}(x, y) \rightarrow R_b(y, z)) \wedge R_b(x, z)) \wedge \\ & R_a(w, x) \rightarrow (\forall y (R_{\uparrow}(x, y) \rightarrow R_b(y, z)) \wedge R_b(x, z))) \\ & \text{iff } \exists z \forall x (\forall y (R_{\uparrow}(w, y) \rightarrow R_b(y, z)) \wedge R_b(w, z) \wedge \\ & R_a(w, x) \rightarrow (\forall y (R_{\uparrow}(x, y) \rightarrow R_b(y, z)) \wedge R_b(x, z))) \\ (\dagger) \text{ iff } & \forall y (R_{\uparrow}(w, y) \rightarrow R_b(y, w)) \wedge R_b(w, w) \wedge \\ & \forall x (R_a(w, x) \rightarrow (\forall y (R_{\uparrow}(x, y) \rightarrow R_b(y, w)) \wedge R_b(x, w))) \\ (\ddagger) \text{ iff } & \forall x (R_a(w, x) \rightarrow (\forall y (R_{\uparrow}(x, y) \rightarrow R_b(y, w)) \wedge R_b(x, w))). \end{aligned}$$

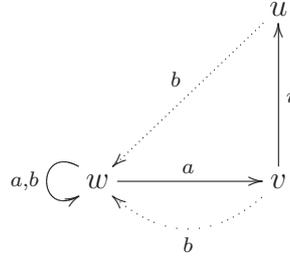
Ad (\star): Derive from $\mathfrak{A} \models (3.11)[x_1/w]$ that $\langle w, w \rangle \in R_a^{\mathfrak{A}}$.

Ad (\circ): In first-order logic $(\phi \vee \psi) \rightarrow \chi$ is equivalent to $(\phi \rightarrow \chi) \wedge (\psi \rightarrow \chi)$.

Ad (\dagger): Derive from $\mathfrak{A} \models (3.12)[x_1/w]$ that if $\langle w, v \rangle \in R_b^{\mathfrak{A}}$, then $w = v$. So z must be assigned w for the conjunct $R_b(w, z)$ to be true on \mathfrak{A} .

Ad (\ddagger): Derive from $\mathfrak{A} \models (3.11)[x_1/w]$ that $\langle w, w \rangle \in R_a^{\mathfrak{A}}$ and that $\langle w, w \rangle \in R_b^{\mathfrak{A}}$. Hence, the conjunct $R_b(w, w)$ can be dropped. Furthermore, if v is reachable from w via an R_{\uparrow} -edge, then one can also get to v by first traversing the reflexive R_a -edge at w and thereafter moving to v along the R_{\uparrow} -edge. So the left conjunct is entailed by the right conjunct and can therefore be omitted. \square

The reader may appreciate a more loose description of what is expressed by the fact that $\mathfrak{A} \models (3.11) \wedge (3.12) \wedge [a]_1 \bigwedge_{i \in \{\uparrow, =\}} [i]_2 (\langle b \rangle / \{1, i, 2\}) \top [x_1/w]$. To this end, consider three objects w , v , and u such that $\langle w, v \rangle \in R_a^{\mathfrak{A}}$ and $\langle v, u \rangle \in R_i^{\mathfrak{A}}$, where i is either \uparrow or $=$ depending on Abelard's choice for $\bigwedge_{i \in \{\uparrow, =\}}$. As indicated by $(\langle b \rangle / \{1, i, 2\})$ Eloise has to move ignorant of all of Abelard's moves. In particular she does not know v nor i , for which reason she also does not know whether $v = u$. Since $\mathfrak{A} \models (3.11)[x_1/w]$, the spy point w is R_a -accessible from itself. So after Abelard's move for $[a]_1$ Eloise conceives it possible that the object v equals w . Because she does not know whether i is $=$, Eloise holds it possible that u equals w . Thus Eloise is able to find a common R_b -successor of w , v , and u . By $\mathfrak{A} \models (3.12)[x_1/w]$ it follows that the only common R_b -successor is w itself since there is no other object R_b -accessible from w . See also the diagram below:



In the following claim it is established that there exists one object u such that every object v that can be reached from w through one R_a -transition followed by one R_{Δ} -transition is R_b accessible from v . If (3.17) holds on \mathfrak{A} under x_1/w , then u is in fact the spy point w .

Claim. $\mathfrak{A} \models (3.18)[x_1/w]$ iff there exists a t , such that for every r, r', s, s' , if $\langle r, r' \rangle \in R_a^{\mathfrak{A}}$, $\langle r', s \rangle \in R_{\Delta}^{\mathfrak{A}}$, and $\langle s, s' \rangle \in R_b^{\mathfrak{A}}$, then $t = s'$;

$$[a]_1 [\Delta]_2 [b]_3 (\langle = \rangle / \{1, 2, 3\}) \top. \quad (3.18)$$

Proof. I prove for Δ equal to \uparrow . Consider the following equivalence:

$$(3.18) \text{ iff } \exists u \forall x (R_a(w, x) \rightarrow (\forall y (R_{\uparrow}(x, y) \rightarrow (\forall z (R_b(y, z) \rightarrow (z = u)))))).$$

□

The following claim shows that $\mathbf{IFV}^{\mathbf{ML}}$ can enforce that R_a be a universal accessibility relation from w , cf. condition (1) above. Put differently, $\mathbf{IFV}^{\mathbf{ML}}$ can enforce that if there is a path of R_{\uparrow} and R_{\Rightarrow} -transitions from w to v , then v is R_a -accessible from w .

Claim. Let $\mathfrak{A} \models (3.13) \wedge (3.14) \wedge (3.16) \wedge (3.17) \wedge (3.18)[x_1/w]$. Then, $\mathfrak{A} \models (3.19)[x_1/w]$ iff for every v and u , if $\langle w, v \rangle \in R_a^{\mathfrak{A}}$ and $\langle v, u \rangle \in R_{\Delta}^{\mathfrak{A}}$ then $\langle w, u \rangle \in R_a^{\mathfrak{A}}$, for $\Delta \in \{\uparrow, \Rightarrow\}$;

$$[a][\Delta] \bigwedge_{i \in \{b, =\}} [i]_1(\langle a \rangle / \{i, 1\}) \top. \quad (3.19)$$

Proof. I prove for Δ equal to \uparrow . Consider the following equivalences:

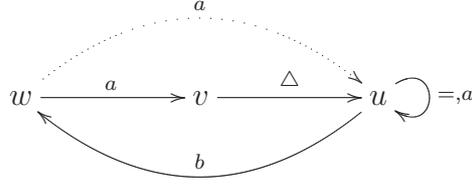
$$\begin{aligned} (3.19) & \text{ iff } \forall x (R_a(w, x) \rightarrow \forall y (R_{\uparrow}(x, y) \rightarrow \Psi)), \text{ where} \\ \Psi & = \exists u (\forall z (R_b(y, z) \rightarrow R_a(z, u)) \wedge \forall z (y = z \rightarrow R_a(z, u))) \\ & \text{ iff } \exists u (\forall z (R_b(y, z) \rightarrow R_a(z, u)) \wedge R_a(y, u)) \\ (\star) & \text{ iff } \exists u (\forall z (R_b(y, z) \rightarrow R_a(z, u)) \wedge R_a(y, u) \wedge u = y) \\ (\circ) & \text{ iff } \forall z (R_b(y, z) \rightarrow R_a(z, y)) \\ (\dagger) & \text{ iff } R_a(w, y). \end{aligned}$$

Ad (\star) : Derive from $\mathfrak{A} \models (3.14) \wedge (3.16)[x_1/w]$ that if $\langle v, v' \rangle \in R_{\uparrow}^{\mathfrak{A}}$, then v and v' disagree on p . Obviously then, v and v' are distinct objects. So for every v, v' , and v'' , if $\langle w, v \rangle \in R_a^{\mathfrak{A}}$, $\langle v, v' \rangle \in R_{\uparrow}^{\mathfrak{A}}$, and $\langle v', v'' \rangle \in R_a^{\mathfrak{A}}$, then $w \neq v''$. By $\mathfrak{A} \models (3.13)[x_1/w]$ it follows that $v' = v''$. So the variables y and u must be assigned the same object for the formula to be true on \mathfrak{A} .

Ad (\circ) : Derive from $\mathfrak{A} \models (3.13)[x_1/w]$ that $\langle v, v \rangle \in R_a^{\mathfrak{A}}$, for every object v . So the conjunct $R_a(y, y)$ is dispensable.

Ad (\dagger) : Derive from $\mathfrak{A} \models (3.17) \wedge (3.18)[x_1/w]$ that for every v, v' , if $\langle v, v' \rangle \in R_b^{\mathfrak{A}}$ then $v' = w$. □

For an informal formulation of the argument, let \mathfrak{A} be a structure from the premise of the claim on which $[a][\Delta] \bigwedge_{i \in \{b, =\}} [i]_1(\langle a \rangle / \{i, 1\}) \top$ is true under x_1/w . Suppose Abelard makes two moves: first he moves along an R_a -edge from w to v ; second he moves along an R_{Δ} -edge from v to u . Third, he chooses the modality R_b or $R_{=}$. The gist is that for every $i \in \{b, =\}$ there is exactly one R_i -successor of u . By the previous claims, namely, w is the only R_b -accessible object from u ; and by definition only u itself is $=$ -accessible from u . Eloise does not know the modality R_i along which Abelard moved previously, but for the aforementioned reasons she knows that she has to point out a common R_a -successor of w and u . Her array of choices appears to be restricted to u , because by $\mathfrak{A} \models (3.13)[x_1/w]$ it follows that u has only one R_a -successor, namely u itself. Therefore, there u must be an R_a -successor of the spy point w . See also the following diagram:



It remains to be shown that $\mathbf{IFV}^{\mathbf{ML}}$ can enforce a grid structure, cf. condition (3) above. It was already observed that all R_a -accessible worlds from w have one R_{\uparrow} -successor s and one R_{\Rightarrow} -successor s' , that are different from each other. It suffices thus to show that for every such pair s and s' there exists one object t that is R_{\Rightarrow} -accessible from s and R_{\uparrow} -accessible from s' .

Claim. Let $\mathfrak{A} \models (3.14) \wedge (3.15) \wedge (3.16)[x_1/w]$. Then, $\mathfrak{A} \models (3.20) \wedge (3.21)[x_1/w]$ iff for every v, s, s', t, t' , such that $\langle w, v \rangle \in R_a^{\mathfrak{A}}$, $\langle v, s \rangle, \langle s', t' \rangle \in R_{\uparrow}^{\mathfrak{A}}$, and $\langle v, s' \rangle, \langle s, t \rangle \in R_{\Rightarrow}^{\mathfrak{A}}$, it is the case that $t = t'$;

$$[a] \left(p \rightarrow \left(\bigwedge_{i \in \{\uparrow, \Rightarrow\}} [i]_2 \bigvee_{j \in \{\uparrow, \Rightarrow\}} (\langle j \rangle / \{i, 2, j\}) \neg p \right) \right) \quad (3.20)$$

$$[a] \left(\neg p \rightarrow \left(\bigwedge_{i \in \{\uparrow, \Rightarrow\}} [i]_2 \bigvee_{j \in \{\uparrow, \Rightarrow\}} (\langle j \rangle / \{i, 2, j\}) p \right) \right). \quad (3.21)$$

Proof. Consider the following equivalences:

$$\begin{aligned} (3.20) \quad & \text{iff } \forall x (R_a(w, x) \rightarrow (P(x) \rightarrow \exists z (\Psi))), \text{ where} \\ & \Psi = \forall y ((R_{\uparrow}(x, y) \rightarrow ((R_{\uparrow}(y, z) \wedge \neg P(z)) \vee (R_{\Rightarrow}(y, z) \wedge \neg P(z)))) \wedge \\ & \quad \forall y ((R_{\Rightarrow}(x, y) \rightarrow ((R_{\uparrow}(y, z) \wedge \neg P(z)) \vee (R_{\Rightarrow}(y, z) \wedge \neg P(z)))))) \\ (\star) \quad & \text{iff } \forall y ((R_{\uparrow}(x, y) \rightarrow ((R_{\uparrow}(y, z) \wedge \neg P(z)) \vee (R_{\Rightarrow}(y, z) \wedge \neg P(z)))) \wedge \\ & \quad (R_{\Rightarrow}(x, y) \rightarrow ((R_{\uparrow}(y, z) \wedge \neg P(z)) \vee (R_{\Rightarrow}(y, z) \wedge \neg P(z)))))) \\ (\circ) \quad & \text{iff } \forall y ((R_{\uparrow}(x, y) \rightarrow (R_{\uparrow}(y, z) \wedge \neg P(z)) \vee \\ & \quad R_{\uparrow}(x, y) \rightarrow (R_{\Rightarrow}(y, z) \wedge \neg P(z))) \\ & \quad \wedge \\ & \quad (R_{\Rightarrow}(x, y) \rightarrow (R_{\uparrow}(y, z) \wedge \neg P(z)) \vee \\ & \quad R_{\Rightarrow}(x, y) \rightarrow (R_{\Rightarrow}(y, z) \wedge \neg P(z)))) \\ (\dagger) \quad & \text{iff } \forall y ((R_{\uparrow}(x, y) \rightarrow R_{\Rightarrow}(y, z) \wedge \neg P(z)) \wedge \\ & \quad (R_{\Rightarrow}(x, y) \rightarrow R_{\uparrow}(y, z) \wedge \neg P(z))) \\ (\ddagger) \quad & \text{iff } \forall y ((R_{\uparrow}(x, y) \rightarrow R_{\Rightarrow}(y, z)) \wedge (R_{\Rightarrow}(x, y) \rightarrow R_{\uparrow}(y, z))). \end{aligned}$$

Ad (\star) : In first-order logic $(\forall x \phi \wedge \forall x \psi)$ is equivalent to $\forall x (\phi \wedge \psi)$.

Ad (\circ) : In first-order logic $\phi \rightarrow (\psi \vee \chi)$ is equivalent to $(\phi \rightarrow \psi) \vee (\phi \rightarrow \chi)$.

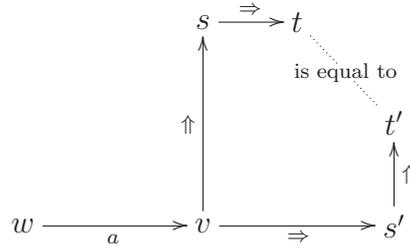
Ad (†): Derive from $\mathfrak{A} \models (3.14) \wedge (3.15) \wedge (3.16)[x_1/w]$ that for every v, v' , and v'' , such that $R_{\Delta}(v, v')$ and $R_{\Delta}(v', v'')$, v and v'' agree on p , for $\Delta \in \{\uparrow, \Rightarrow\}$. As the antecedent holds true if the object assigned to x makes p true, the disjuncts

$$R_{\uparrow}(x, y) \rightarrow (R_{\uparrow}(y, z) \wedge \neg P(z)) \quad \text{and} \quad R_{\Rightarrow}(x, y) \rightarrow (R_{\Rightarrow}(y, z) \wedge \neg P(z))$$

are never true and can be omitted without affecting the formula's truth conditions.

Ad (‡): Derive from $\mathfrak{A} \models (3.14) \wedge (3.15)[x_1/w]$ that for every v, v' , and v'' , such that $R_{\uparrow}(v, v')$ and $R_{\Rightarrow}(v', v'')$, v and v'' disagree on p . So if p is true on v , $\neg p$ is true on v'' and the conjuncts $\neg P(z)$ are dispensable. \square

To grasp the underlying idea, consider a structure \mathfrak{A} that meets the premises of the claim and an object w that makes P_p true and is R_a -accessible from w . Furthermore, consider two Eloises: Eloise_j and $\text{Eloise}_{\langle j \rangle}$, the former controlling the disjunctive quantifier, the latter controlling the modal operator. As for the actual game playing, it is the case that Abelard moves either along the R_{\uparrow} -edge to the object s or along the R_{\Rightarrow} -edge to the object s' . (Recall that every object R_a -accessible from w has exactly one R_{\uparrow} -successor and exactly one R_{\Rightarrow} -successor, due to $\mathfrak{A} \models (3.14) \wedge (3.15) \wedge (3.16)[x_1/w]$.) Since $\mathfrak{A} \models (3.14) \wedge (3.15)[x_1/w]$, s is not a P -object, whereas s' is. For $\text{Eloise}_{\langle j \rangle}$ to be able to move to a non- P -object, she must be able to move along the R_{\Rightarrow} -edge if Abelard chose $i = \uparrow$, and along the R_{\uparrow} -edge if Abelard chose $i = \Rightarrow$. However, all $\text{Eloise}_{\langle j \rangle}$ knows is actually the object v —she does not know whether Abelard advanced to s or s' . So for the Eloises to win there must be an object, call it t , such that t is R_{\Rightarrow} -accessible from s and R_{\uparrow} -accessible from s' . Eloise_j takes care the appropriate modality is chosen, by setting $j = \Rightarrow$ if Abelard moved to s , and $j = \uparrow$ if Abelard moved to s' . The case in which v is not a P -object is similar, using the fact that $\mathfrak{A} \models (3.16)[x_1/w]$. See also the diagram below:



The proof of Theorem 3.5.8 follows readily in view of the aforementioned rationale and the claims.

Proof of Theorem 3.5.8. Put

$$\Phi_T = (3.11) \wedge \dots \wedge (3.20) \wedge [a]\theta_T,$$

where θ_T is the conjunction of

$$\bigvee_{t \in T} \left(p_t \wedge \bigwedge_{t' \in T - \{t\}} \neg p_{t'} \right)$$

and

$$\bigwedge_{t \in T} \left(\left(p_t \rightarrow [\Rightarrow] \bigvee_{t' \in T(t, \Rightarrow)} \neg p_{t'} \right) \wedge \left(p_t \rightarrow [\Uparrow] \bigvee_{t' \in T(t, \Uparrow)} \neg p_{t'} \right) \right).$$

In the latter formulae, the set $T(t, \Uparrow)$ contains all tiles t' from T such that $up(t) = down(t')$, and similarly for $T(t, \Rightarrow)$. The formula θ_T is copied from (Blackburn, de Rijke, and Venema 2001, Theorem 6.31), and ensures that every object carries exactly one tile and that two adjacent objects are appropriately tiled with respect to their common edge.

It remains to be proved that there exists a τ -structure \mathfrak{A} and an object w such that $\mathfrak{A} \models \Phi_T[x_1/w]$ iff $T \in \text{TILING}$. I.e., Φ_T is satisfiable iff T can tile $\mathbb{N} \times \mathbb{N}$.

From left to right. Suppose there exists a τ -structure \mathfrak{A} and an object w such that $\mathfrak{A} \models \Phi_T[x_1/w]$. Consider the set of objects $B \subseteq A$, such that

$$B = \{c \in A \mid \langle w, c \rangle \text{ is in the transitive, reflexive closure of } R_{\Uparrow}^{\mathfrak{A}} \cup R_{\Rightarrow}^{\mathfrak{A}}\}.$$

Consider the structure $\mathfrak{B} = \langle B, \langle S^{\mathfrak{B}} \rangle_{S \in \tau} \rangle$, where $S^{\mathfrak{B}} = S^{\mathfrak{A}} \cap B^a$ and a is the arity of S . \mathfrak{B} can be regarded a *substructure* of \mathfrak{A} . In \mathfrak{B} the relations R_a , R_{\Uparrow} , and R_{\Rightarrow} from τ are interpreted such that:

- (1) For every $c \in B$, $\langle w, c \rangle \in R_a^{\mathfrak{B}}$, due to (3.13), (3.14), (3.16), (3.17), (3.18), and (3.19).
- (2a) For every $c \in B$, there are exactly two distinct d and d' such that $\langle c, d \rangle \in R_{\Rightarrow}^{\mathfrak{B}}$ and $\langle c, d' \rangle \in R_{\Uparrow}^{\mathfrak{B}}$, due to (3.14), (3.15), and (3.16). It follows that $R_{\Rightarrow}^{\mathfrak{B}}$ and $R_{\Uparrow}^{\mathfrak{B}}$ are functional on B .
- (2b) For every $c, d, d', e, e' \in B$, such that $\langle c, d \rangle, \langle d', e' \rangle \in R_{\Rightarrow}^{\mathfrak{B}}$ and $\langle c, d' \rangle, \langle d, e \rangle \in R_{\Uparrow}^{\mathfrak{B}}$, it is the case that $e = e'$, due to (3.14), (3.15), (3.16), (3.20), and (3.21). That is, $R_{\Rightarrow}^{\mathfrak{B}}$ and $R_{\Uparrow}^{\mathfrak{B}}$ commute on B .

From (2a) and (2b) it follows that \mathfrak{B} is a grid. From (1) and the fact that $\mathfrak{A} \models [a]\theta_T[x_1/w]$, it follows that θ_T holds on every object in B . Hence, \mathfrak{B} witnesses the fact that $T \in \text{TILING}$.

From right to left. Suppose T can tile $\mathbb{N} \times \mathbb{N}$. Let $f : \mathbb{N} \times \mathbb{N} \rightarrow T$ be a tiling of the plane. Let the τ -structure \mathfrak{C} be defined on the basis of f as follows:

$$\begin{aligned}
C &= (\mathbb{N} \times \mathbb{N}) \\
R_a^{\mathfrak{C}} &= (\langle 0, 0 \rangle \times C) \cup R_{=}^{\mathfrak{C}} \\
R_b^{\mathfrak{C}} &= (C \times \langle 0, 0 \rangle) \\
R_{\uparrow}^{\mathfrak{C}} &= \{ \langle \langle n, n' \rangle, \langle n+1, n' \rangle \rangle \mid \langle n, n' \rangle \in C \} \\
R_{\Rightarrow}^{\mathfrak{C}} &= \{ \langle \langle n, n' \rangle, \langle n, n'+1 \rangle \rangle \mid \langle n, n' \rangle \in C \} \\
P^{\mathfrak{C}} &= \{ \langle n, 2n' \rangle \mid \langle n, n' \rangle \in C \} \\
Q^{\mathfrak{C}} &= \{ \langle 2n, n' \rangle \mid \langle n, n' \rangle \in C \} \\
P_t^{\mathfrak{C}} &= \{ \langle n, n' \rangle \in C \mid f(n, n') = t \}.
\end{aligned}$$

Clearly, $\mathfrak{C} \models \Phi_T[x_1/\langle 0, 0 \rangle]$, and the result follows. \square

Thus observe that $ST(\mathbf{MLV})$ loses decidability once it is subject to the IF procedure. As I mentioned before, Theorem 3.5.8 establishes the lower bound of $\mathbf{IFV}^{\mathbf{ML}}$'s complexity. Its upper bound is trivially Σ_1^1 , since it is a fragment of \mathbf{IFV} . The exact complexity of $\mathbf{IFV}^{\mathbf{ML}}$ is left as an open problem.

Undecidability for $\mathbf{IFV}^{\mathbf{ML}}$ gives rise to yet another question: What restrictions should be imposed in order to regain decidability? Natural restrictions suggest themselves from the proofs of the previous claims. For instance, all but three of the conjuncts (3.11) $\wedge \dots \wedge$ (3.20) use the equality symbol. By this observation it seems that the equality symbol is essential for the proof, and the question remains if $\mathbf{IFV}^{\mathbf{ML}}$ would be decidable in its absence. In a similar vein, note that for every conjunct in Φ_T that has a conjunctively quantified index i it is the case that i appears in the slash-set of the conjunct's modal operator belonging to Eloise: $([\dots]/\{\dots, i, \dots\})$.

3.6 Concluding remarks

In this chapter, a natural set of game rules was introduced, that span the class of IF semantic games. This rule book requires as many Eloises as there are existential variables in the formula at stake. Introducing multiple Eloises is a way to overcome the ‘‘perfect recall problem’’. A more direct way to overcome this problem is by ignoring all IF-formulae whose semantic games violate perfect recall. In this manner, I defined the perfect recall fragment of IF logic. Theorem 3.4.3 shows that the perfect recall fragment of IF logic coincides with first-order logic, when it comes to expressive power. In the spirit of ongoing research on modal logic, I defined a new independence-friendly modal logic, $\mathbf{IFV}^{\mathbf{ML}}$, naturally as a fragment of \mathbf{IFV} as follows:

- obtain the first-order correspondence language of \mathbf{MLV} : $ST'(\mathbf{MLV})$;

- produce the language $ST(\mathbf{ML}\forall)$ by seeing to it that every variable in every $ST'(\mathbf{ML}\forall)$ -formula is quantified at most once without affecting the formula's truth condition; and
- apply the IF procedure constituted by (IF) to $ST(\mathbf{ML}\forall)$ and get $\mathbf{IF}\forall^{\mathbf{ML}}$.

I showed that this logic's semantic games meet the action consistency requirement. $\mathbf{IF}\forall^{\mathbf{ML}}$ was proved undecidable by means of a reduction from TILING.

Basic modal logic's satisfiability problem is decidable, yet Theorem 3.5.8 shows that $\mathbf{IF}\forall^{\mathbf{ML}}$ is Π_1^0 -hard. Theorem 3.5.8 does not settle the case for $\mathbf{IF}\forall^{\mathbf{ML}}$ up to completeness. Further research on this topic is desirable, which may take into account also the influence of the equality symbol and slashing over indices. This enterprise can be seen as an investigation of the trade-off between modal languages on the one side and imperfect information languages on the other side.

Other interesting questions concerning $\mathbf{IF}\forall^{\mathbf{ML}}$ are readily conceived and include the issues of frame definability and bisimulation.

In (Hyttinen and Tulenheimo 2005) it was shown that the "perfect recall" language $\mathbf{IFML}^{\mathbf{PR}}$ is decidable. The authors baptized this language "perfect recall", on account of the association of the language's syntax with perfect recall games. I do not wish to argue that this predicate is ill-chosen, only I point out that it is chosen on *syntactic* grounds. By contrast, recall that in this dissertation the language $\mathbf{IF}^{\mathbf{PR}}$ was defined on semantic grounds, namely on the basis of the semantic games of the formulae. It is interesting to investigate whether the reported result for the language $\mathbf{IFML}^{\mathbf{PR}}$ can be repeated for the semantically defined perfect recall fragment of $\mathbf{IF}\forall^{\mathbf{ML}}$.

In (Cameron and Hodges 2001) it was shown that there cannot exist a compositional semantics for IF logic in terms of single assignments, such as Tarski semantics for first-order logic. Theorem 3.4.3 shows that there exist fragments of IF logic whose semantic games have imperfect information whose expressive power does not exceed first-order logic's. This raises the question whether for those fragments it is possible to construct a compositional semantics in terms of single assignments.

This chapter drew upon the connection between independence-friendliness in logic and games with imperfect information in game theory. It was discovered that one can apply notions from game theory to IF logic and study their impact, from a model-theoretic and computational perspective. The insights thus obtained are mixed: imposing perfect recall on full IF logic decreases the expressive power to the level of first-order logic, yet the basic modal logic that was extended so as to incorporate imperfect information becomes undecidable.

The results in this chapter show that IF logic harbors interesting fragments with surprising properties. Those fragments may have common motivations from computational logic; but also fresh ones from game theory.

Chapter 4

Partially ordered connectives

Henkin quantifiers (Henkin 1961) can be regarded as blocks of independence-friendly quantifiers, with highly regular independence schemes. Traditionally Henkin quantifiers have been remotely associated with imperfect information, but no publications investigate this viewpoint. In this chapter I will do so, and observe that the regularity of the Henkin quantifiers' independence schemes opens up different ways of explaining the origin of imperfect information. Two of them are in fact cognitively involved: *limited memory* and *absentmindedness*. Furthermore, I will show that restricting the number of actions leads to so-called *partially ordered connectives* introduced in (Sandu and Väänänen 1992). A substantial part of this chapter is devoted to mapping out the finite model theory of logics with partially ordered connectives.

4.1 Introduction

Fagin's Theorem (Fagin 1974), cited as Theorem 2.5.1 in this dissertation, reveals the intimate connection between finite model theory and complexity theory. As a methodological consequence it appears that questions and results regarding a complexity class may be relevant to logic and vice versa. For instance, the complexity theorist's headaches caused by the $\mathbf{NP} = \mathbf{coNP}$ -problem can now be shared by the logician working on the $\Sigma_1^1 = \Pi_1^1$ -problem. Solving the $\mathbf{NP} = \mathbf{coNP}$ -problem is worth a headache: if $\mathbf{NP} \neq \mathbf{coNP}$, then $\mathbf{P} \neq \mathbf{NP}$. Indeed, logicians working within finite model theory address this problem. By and large they proceed by mapping out *fragments* of various logics. A case in point is Fagin's (1975) study of the *monadic* fragments of Σ_1^1 and Π_1^1 , showing that they do not coincide. (The *monadic (k -ary)* fragment of Σ_1^1 contains only those formulae in which relation variables are unary (have arity $\leq k$). For a formal definition, see Definition 4.3.3.) Ajtai, Fagin, and Stockmeyer (2000, pg. 661) reflect on this approach in the following way:

“Instead of considering Σ_1^1 (= **NP**) and its complement Π_1^1 (= **coNP**) in their full generality, we could consider the monadic restriction of these classes, i.e., the restriction obtained by allowing second-order quantification only over sets (as opposed to quantification over, say, binary relations). [...] The hope is that the restriction to the monadic classes will yield more tractable questions and will serve as a training ground for attacking the problems in their full generality.”

To the finding that monadic $\Sigma_1^1 \neq$ monadic Π_1^1 one might respond: *One down, infinitely many to go*. For Fagin’s result does not apply to the respective binary fragments, let alone the k -ary fragments, for any $k \geq 2$. But this response would be overly pessimistic. Because in the first place we observe that monadic Σ_1^1 can describe problems that are amongst the ones most typical for **NP**: **NP**-complete problems, such as 3-COLORABILITY and SAT. Secondly, there is the empirical observation that the vast majority of **NP**-computable problems encountered in everyday practice can be characterized by a formula in binary Σ_1^1 . Bearing this observation in mind, *one down, one to go* might be a more appropriate reaction.

The results in (Fagin 1975) aroused a lot of interest in monadic languages (Turán 1984; Ajtai and Fagin 1990; Ajtai, Fagin, and Stockmeyer 2000), but disappointingly, it is unknown whether binary, existential, second-order logic can be separated from 3-ary, existential, second-order logic, cf. (Durand et al. 1998), or from binary, universal, second-order logic.

Parts of this chapter are within the tradition of mapping out the finite model theory of fragments of Σ_1^1 . In this chapter I will concern myself with languages with *Henkin quantifiers* and *partially ordered connectives*. The theory of partially ordered quantification was started by Henkin (1961). Henkin entertained himself with an innovative way of quantification, found in special *prefixes* generally known as *Henkin quantifiers*. A Henkin quantifier with *dimensions* k and n is depicted by

$$\left(\begin{array}{cccc} \forall x_{11} & \dots & \forall x_{1k} & \exists y_1 \\ \vdots & \ddots & \vdots & \vdots \\ \forall x_{n1} & \dots & \forall x_{nk} & \exists y_n \end{array} \right). \quad (4.1)$$

In the interest of brevity, I will normally write $H_k^n \vec{x} \vec{y}$ or even H_k^n to denote the Henkin quantifier (4.1).

This two-dimensional way of representation aims to convey that the variable y_i depends on $\forall \vec{x}_i = \forall x_{i1} \dots \forall x_{ik}$ and on $\forall \vec{x}_i$ only. In this way, the Henkin quantifier, with dimensions k and n , is defined by the following string of quantifiers from Independence-friendly logic:

$$\forall x_{11} \dots \forall x_{1k} (\exists y_1 / Y_1) \dots \forall x_{n1} \dots \forall x_{nk} (\exists y_n / Y_n), \quad (4.2)$$

where $Y_h = \{x_{ij} \mid 1 \leq i < h, 1 \leq j \leq k\}$.

Let the logic $\mathbf{H}(\tau)$ be the result of applying Henkin quantifiers of arbitrary (but finite) dimensions to first-order formulae over the vocabulary τ :

$$\Phi ::= \psi \mid \mathbf{H}_k^n \psi,$$

where k, n are integers and ψ ranges over $\mathbf{FO}(\tau)$. The semantics for $\mathbf{H}(\tau)$ is typically given in terms of Skolem functions, which are very much similar to IF logic. Let \mathfrak{A} be a τ -structure and let α be an assignment in \mathfrak{A} .

Then, define

$$\mathfrak{A} \models \mathbf{H}_k^n \vec{x}_1 \dots \vec{x}_n y_1 \dots y_n \psi(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n)[\alpha]$$

iff there exist k -ary functions f_1, \dots, f_n on A such that

$$\mathfrak{A} \models \forall \vec{x}_1 \dots \forall \vec{x}_n \psi(\vec{x}_1, \dots, \vec{x}_n, f_1(\vec{x}_1), \dots, f_n(\vec{x}_n))[\alpha].$$

The finding that the logic \mathbf{H} coincides with full Σ_1^1 , cf. (Enderton 1970; Walkoe 1970) is a milestone result in the theory of partially ordered quantification. In fact, the proof of Theorem 3.2.8 on page 36 about the expressive power of IF logic is derived from the latter result and the insight that (4.1) and (4.2) define each other.

Blass and Gurevich (1986) drew upon the connection between \mathbf{H} , Σ_1^1 , and \mathbf{NP} on finite structures. Their publication departs from the view that Fagin's Theorem can be cast in terms of partially ordered quantifiers: \mathbf{H} captures \mathbf{NP} . Bearing this in mind, the study of fragments of \mathbf{H} can be justified by the same arguments that justify the interest in arity bounded fragments of Σ_1^1 .

In this chapter I will study languages with partially ordered connectives that feature restricted quantifiers $\bigvee_{i \in \{0,1\}}$ instead of existential quantifiers. The logic \mathbf{D} is the result of prefixing first-order logic with partially ordered connectives, analogous to \mathbf{H} . As it contains *restricted* quantifier prefixes, the hope is justified that \mathbf{D} gives rise to new fragments of $\mathbf{H} = \mathbf{NP}$, over finite structures.

This chapter contributes several results to finite model theory, the first one being a characterization of \mathbf{D} as a fragment of Σ_1^1 . Furthermore it is shown that (a) \mathbf{D} can express a property expressible in $k+1$ -ary, existential, second-order logic that cannot be expressed in k -ary, existential, second-order logic, and that (b) \mathbf{D} is not closed under complementation, as it can express 2-COLORABILITY but not its complement. Since the characterization of \mathbf{D} is rather natural it may provide handles for future research on Σ_1^1 .

In Section 4.2, I introduce the game-theoretic framework for Henkin quantifiers and partially ordered connectives, which provides a viewpoint to appreciate Henkin quantifiers and the like. It is interesting to compare the game-theoretic semantics of Henkin quantifiers with those of IF logic. I showed in Section 3.3 that the imperfect information in IF semantic games can be accounted for by

teams of players and the attribute of envelopes. Interestingly, I show that the imperfect information in Henkin quantifiers and related quantifiers can be seen to be brought about by cognitive restrictions on agents, such as a limited number of memory cells, and absent-mindedness.

In Section 4.3, I collect the definitions of the logics with partially ordered connectives and the relevant fragments of second-order logic. Roughly speaking, I will look at two languages: the one being first-order logic prefixed with one partially ordered connective, denoted by \mathbf{D} , the other being the full closure of this language, written $\mathcal{L}(\mathbf{D})$, that allows for nested partially ordered connectives, interspersed with negations and first-order quantifiers.

In Section 4.4, I review a selection of the literature on partially ordered connectives, either because it will be used, or because it outlines the field of research.

In Section 4.5, I give a translation from \mathbf{D} into of Σ_1^1 .

In Section 4.6 this translation result is strengthened by giving the characterization of \mathbf{D} in terms of a natural fragment of Σ_1^1 denoted by $\Sigma_1^1\heartsuit$.

In Section 4.7, I show that \mathbf{D} can express a property expressible in $k+1$ -ary, existential, second-order logic that cannot be expressed in k -ary, existential, second-order logic, using the characterization result, see (a) above. Furthermore, I show that on linear ordered structures \mathbf{D} captures \mathbf{NP} .

In Section 4.8, I introduce a *model comparison game* for \mathbf{D} , à la Ehrenfeucht and Fraïssé.

In Section 4.9, I give an application of this game, leading to a non-expressibility result for \mathbf{D} which implies that \mathbf{D} is not closed under complementation, see (b). It also follows that on arbitrary finite structures $\mathbf{D} < \mathbf{NP}$.

In Section 4.10, I treat the descriptive complexity of queries from $\mathcal{L}(\mathbf{D})$ and $\mathcal{L}(\mathbf{H})$. I show that the upper bound of the expression complexity for $\mathcal{L}(\mathbf{H})$ is $\mathbf{P}_{||}^{\mathbf{NP}}$, and that $\mathcal{L}(\mathbf{H})$ and $\mathcal{L}(\mathbf{D})$ capture $\mathbf{P}_{||}^{\mathbf{NP}}$ on linear ordered structures.

Section 4.11 concludes the chapter.

4.2 GTS for partially ordered prefixes

Henkin quantifiers can be seen as blocks of IF quantifiers with highly regular independence schemes. I show that the imperfect information in semantic games of \mathbf{H} can be understood in terms of non-absentminded agents with a limited number of memory cells. In the same vein, I show that *function quantifiers* from (Krynicki and Väänänen 1989) are played by similar but absentminded agents. In this manner, we observe that Henkin quantifiers and function quantifiers are played by cognitively bounded agents. Recall that the imperfect information in IF games was explained in Section 3.3 in terms of multiple players and envelopes. I show that partially ordered connectives can be understood as limiting the array of actions available to the players.

For the sake of simplicity I shall restrict myself in this section to \mathbf{H} -formulae $\mathbf{H}_k^n \vec{x} \vec{y} \phi$ in which ϕ is a possibly negated atom. Furthermore, I assume that all variables in ϕ are bound by the Henkin quantifier $\mathbf{H}_k^n \vec{x} \vec{y}$. On this assumption, the game rules for the semantic game of $\mathbf{H}_k^n \vec{x} \vec{y} \phi$ are simply the ones for the semantic game of the first-order

$$\forall \vec{x}_1 \exists y_1 \dots \forall \vec{x}_n \exists y_n \phi,$$

which is a game of perfect information as before. The imperfect information is introduced by assuming that the semantic game of $\mathbf{H}_k^n \vec{x} \vec{y} \phi$ is being played by an agent $A_{\text{not-a}}^k$ who has exactly k memory cells and manages his cells in a *first in first out* fashion. This assumption implies that when $A_{\text{not-a}}^k$ is deciding on an object for y_i , $A_{\text{not-a}}^k$ “knows” only the objects picked up over the k previous rounds, that is, the objects assigned to $\vec{x}_i = x_{i1}, \dots, x_{ik}$. Furthermore, I postulate that $A_{\text{not-a}}^k$ is not *absentminded*. That is, if $A_{\text{not-a}}^k$ cannot distinguish two histories h and h' , then $\ell(h) = \ell(h')$. This postulate implies that when choosing an object to assign to y_i the agent $A_{\text{not-a}}^k$ “knows” that the object selected will be assigned to y_i .

In this manner, every \mathbf{H} -formula Φ and suitable structure \mathfrak{A} give rise to an extensive game with imperfect information, call it $Sem\text{-}game^{\mathbf{H}}(\Phi, \mathfrak{A})$. In this extensive game, two histories h and h' are indistinguishable, if the last k elements in h and h' coincide and the length of h is equal to the length of h' .

I refrain myself from a rigorous definition of these games, as they are highly similar to the ones defined for IF logic in Definition 3.2.7 on page 34.

4.2.1. PROPOSITION. *For every \mathbf{H} -formula Φ as in (4.1) and suitable structure \mathfrak{A} , the agent $A_{\text{not-a}}^k$ has a winning strategy in $Sem\text{-}game^{\mathbf{H}}(\Phi, \mathfrak{A})$ iff $\mathfrak{A} \models \Phi$.*

It is noteworthy that the “origin” of the imperfect information in IF semantic games is explained in Section 3.3 by means of special game rules and information hiding items, such as envelopes. By contrast, the imperfect information in the semantic games for Henkin quantifiers is explained by reference to the restricted cognitive capabilities of the agents. Limited memory is a case in point, but one may also study the consequences of dropping non-absentmindedness.

It was observed by Krynicki and Mostowski (1995, pg. 223) that many \mathbf{H} -formulae appearing in the literature express the existence of one single function on the universe at hand. As such many \mathbf{H} -formulae sit in a certain fragment of \mathbf{H} , that was studied by Krynicki and Väänänen (1989). This particular fragment is defined by the *function quantifier* \mathbf{F}_k^n , that binds the variables $x_{11}, \dots, x_{nk}, y_1, \dots, y_n$, just like the Henkin quantifier with dimensions k and n . (I will adhere to the same notational conventions that apply to Henkin quantifiers.) The logic \mathbf{F} is defined as the language containing all strings of the form

$$\mathbf{F}_k^n \vec{x}_1 \dots \vec{x}_n y_1 \dots y_n \phi(\vec{x}_1, \dots, \vec{x}_n, y_1, \dots, y_n), \quad (4.3)$$

where $\vec{x}_i = x_{i1}, \dots, x_{ik}$ as before and ϕ is a possibly negated atom. As for the semantics, any formula (4.3) is true on a suitable structure \mathfrak{A} iff there exists *one*

k -ary function f on A such that

$$\mathfrak{A} \models \forall \vec{x}_1 \dots \forall \vec{x}_n \phi(\vec{x}_1, \dots, \vec{x}_n, f(\vec{x}_1), \dots, f(\vec{x}_n)).$$

From a game-theoretic point of view, we will see that the move from Henkin quantifiers to function quantifiers resembles imposing absentmindedness on the agent $A_{\text{not-a}}^k$ playing according to the rules of the semantic game of $\forall \vec{x}_1 \exists y_1 \dots \forall \vec{x}_n \exists y_n \phi$ on \mathfrak{A} . Just as with \mathbf{H} , if Φ is a \mathbf{F} -formula let $\text{Sem-game}^{\mathbf{F}}(\Phi, \mathfrak{A})$ be the extensive game with imperfect information that models a k -memory cell, first in first out, absentminded agent A_a^k in the latter game. In this extensive game, crucially, two histories h and h' are indistinguishable, if the last k elements in h and h' coincide. However, h and h' need not be of equal length, due to the agent's absentmindedness.

4.2.2. PROPOSITION. *For every \mathbf{F} -formula Φ as in (4.3) and suitable structure \mathfrak{A} , the agent A_a^k has a winning strategy in $\text{Sem-game}^{\mathbf{F}}(\Phi, \mathfrak{A})$ iff $\mathfrak{A} \models \Phi$.*

In the same vein let us restrict the agent's powers to an even greater extent. Consider a poor sort of agent: absentminded and in possession of only one memory cell, that is, it recalls only the last move.¹ On top of this provide the agent with a fixed and finite array of actions. By contrast, recall that the number of actions in any semantic game on structure \mathfrak{A} available to the agent is unbound, since it equals the cardinality of \mathfrak{A} 's universe.

Would this agent be the refuse of the semantic game players society, frowned upon by the perfect memory and non-absentminded upper class? I think not. Imagine a figure skating dancing jury member—surely a highly respected citizen of the society. Her array of actions is fixed: holding up a sign with a mark in the range from 1 to 10. A professional member of the jury takes into account the performance of the skaters and nothing but the performance. So in particular, it should not matter to the objective judgment of the current performance what where the previous performances like, and neither should the ordering of the skaters be a factor in the evaluation. Henceforth, a *jury member* refers to a one-memory cell, absentminded player having a fixed and finite array of actions.

Moving from game theory to logic, the question is what logical language would give rise to extensive semantic games that are playable by jury members. Consider the *jury member prefix with actions I* as in

$$J_I^n x_1 \dots x_n i_1 \dots i_n \gamma(i_1, \dots, i_n)(x_1, \dots, x_n) \quad (4.4)$$

in which J_I^n prefixes the function $\gamma(i_1, \dots, i_n)(x_1, \dots, x_n)$ that maps every string of actions from $I^{\{i_1, \dots, i_n\}}$ to an atomic first-order formula over the variables x_1, \dots, x_n . Define

$$\mathfrak{A} \models J_I^n x_1 \dots x_n i_1 \dots i_n \gamma(i_1, \dots, i_n)(x_1, \dots, x_n)$$

¹Some authors would call this agent *memory-free*, as it obtains all its information from observing the last action taken in the game.

iff there exists one function f of type $A \rightarrow I$ such that

$$\mathfrak{A} \models \forall x_1 \dots \forall x_n \gamma(f(x_1), \dots, f(x_n))(x_1, \dots, x_n).$$

For one thing, jury members do an elegant job coloring graphs. Consider namely the following formula:

$$\Psi_3 = \mathbf{J}_{\{1,2,3\}}^2 x_1 x_2 i_1 i_2 \gamma(i_1, i_2)(x_1, x_2), \quad (4.5)$$

where

$$\gamma(i, j)(x_1, x_2) = \begin{cases} \top & \text{if } i \neq j \\ \neg R(x_1, x_2) & \text{if } i = j. \end{cases}$$

Then, Ψ_3 is true on a graph \mathfrak{G} iff there exists a function $f : G \rightarrow \{1, 2, 3\}$ such that

$$\mathfrak{G} \models \forall x_1 \forall x_2 (f(x_1) = f(x_2) \rightarrow \neg R(x_1, x_2)),$$

that is, if \mathfrak{G} is 3-colorable. Let \mathbf{J} be the language all of whose formulae are as in (4.4). Let the game rules of a semantic evaluation game for a formula $\Phi \in \mathbf{J}$ be defined as the game rules of the semantic game of the first-order formula

$$\Phi^\spadesuit = \forall x_1 \bigvee_{i_1 \in I} \dots \forall x_n \bigvee_{i_n \in I} \gamma(i_1, \dots, i_n)(x_1, \dots, x_n),$$

and let $\text{Sem-game}^{\mathbf{J}}(\Phi, \mathfrak{A})$ be the extensive game with imperfect information that models the semantic game starting from $\langle \Phi^\spadesuit, \mathfrak{A} \rangle$ being played by a jury member with actions I . Jury members have a properly defined logical language of their own:

4.2.3. PROPOSITION. *For every \mathbf{J} -formula Φ as in (4.4) and suitable structure \mathfrak{A} , a jury member with actions I has a winning strategy in the semantic game $\text{Sem-game}^{\mathbf{J}}(\Phi, \mathfrak{A})$ iff $\mathfrak{A} \models \Phi$.*

Interestingly, the jury member logic \mathbf{J} is akin to the logic studied in (Blass and Gurevich 1986; Sandu and Väänänen 1992). These papers study so-called *partially ordered connectives* whose semantic games are played by non-absentminded jury members. The study of partially ordered connectives on finite structures is taken up in subsequent sections in this chapter.

Let me conclude this paragraph with two remarks:

- In semantic evaluation games for first-order logic, Eloise is an agent with an indefinite amount of memory. The winning conditions for a semantic game for a first-order formula ϕ are expressible in first-order logic, namely by ϕ itself. Turning to semantic evaluation games for \mathbf{H} , the agent at hand has only a fixed number of memory cells. Amusingly, capturing the winning

conditions for this agent in a semantic game requires a logic with expressive power stronger than first-order logic, namely Σ_1^1 . The same phenomenon has a computational manifestation: the expression complexity of \mathbf{J} is \mathbf{NP} -complete, as the logic can express 3-COLORABILITY, witnessed by Ψ_3 . On the other hand, it is well-known that the expression complexity of fixed first-order is in deterministic logspace, cf. (Immerman 1999).

Although expressive power and expression complexity seem to show that imperfect information makes life harder, other measures of complexity may of course give different outcomes. In (van Benthem 2001) it is shown that to axiomatize the class of perfect information games one needs more axioms.

- Absentmindedness is a mind boggling concept in game theory. By contrast, as was observed by Krynicki and Mostowski (1995), many \mathbf{H} -formulae appearing in the literature happen to be implementations of \mathbf{F} -formulae. Interestingly, the formula Ψ_3 in (4.5) expresses 3-COLORABILITY in an elegant way. So what is dubious a notion in game theory, may be bon ton in logic and give rise to neat syntactic characterizations. This makes one wonder whether the semantic games thus introduced may be of any assistance in the clarification of the concept of absentmindedness in game theory.

4.3 Logics with partially ordered connectives

In this section, I introduce the syntax and semantics for two languages with partially ordered connectives, that will be the primary objects of investigation. The definitions are taken from (Sandu and Väänänen 1992) without substantial modifications.

4.3.1. DEFINITION. Let IND be the countable set of *indices*, let k be an integer, and let τ be a vocabulary. Define an *implicit matrix τ -formula* as a function of type $\{0, 1\}^k \rightarrow \mathbf{FO}(\tau)$. Let the language $\mathcal{L}(\mathbf{D})(\tau)$ be defined by the following grammar:

$$\Gamma ::= \gamma \mid \neg\Gamma \mid \Gamma \vee \Gamma \mid \exists x \Gamma \mid \left(\begin{array}{cccc} \forall x_{11} & \dots & \forall x_{1k} & \forall i_1 \\ \vdots & \ddots & \vdots & \vdots \\ \forall x_{1k} & \dots & \forall x_{nk} & \forall i_n \end{array} \right) \Gamma,$$

where γ ranges over the implicit matrix τ -formulae. For the sake of readability, I may abbreviate the matrix of quantifiers by $\mathbf{D}_k^n \vec{x}$ or even by \mathbf{D}_k^n , if no confusion threatens. \mathbf{D}_k^n is called a *partially ordered connective*, and k, n are called its *dimensions*.

Special attention will be with the fragment of $\mathcal{L}(\mathbf{D})(\tau)$ containing only implicit matrix τ -formulae that are prefixed by a single partially ordered connective.

Let $\mathbf{D}_k^n(\tau)$ be the fragment of $\mathcal{L}(\mathbf{D})(\tau)$ all of whose formulae are of the form $\mathbf{D}_k^n \gamma$. Finally, put

$$\begin{aligned}\mathbf{D}_k(\tau) &= \bigcup_n \mathbf{D}_k^n(\tau) \\ \mathbf{D}(\tau) &= \bigcup_k \mathbf{D}_k(\tau).\end{aligned}$$

Let the complementary language of $\mathbf{D}(\tau)$ be defined as the language that is obtained by negating every formula in $\mathbf{D}(\tau)$, written $\neg\mathbf{D}(\tau)$.

The notion of *free* and *bound* variable and index is defined with respect to $\mathcal{L}(\mathbf{D})(\tau)$ as usual. A *sentence* of $\mathcal{L}(\mathbf{D})(\tau)$ is a $\mathcal{L}(\mathbf{D})(\tau)$ -formula without free variables and indices.

4.3.2. DEFINITION. Let τ be a vocabulary, let \mathfrak{A} be a τ -structure, and let α be an assignment in \mathfrak{A} that is extended with respect to indices. That is, α maps every index from IND to $\{0, 1\}$, a set of objects disjoint from A . Then, define the satisfaction relation as follows:

$$\mathfrak{A} \models \mathbf{D}_k^n \vec{x}_1 \dots \vec{x}_n i_1 \dots i_n \Gamma(i_1, \dots, i_n)(\vec{x}_1, \dots, \vec{x}_n)[\alpha]$$

iff there exist k -ary functions f_1, \dots, f_n of type $A^k \rightarrow \{0, 1\}$ such that

$$\mathfrak{A} \models \forall \vec{x}_1 \dots \forall \vec{x}_n \Gamma(f_1(\vec{x}_1), \dots, f_n(\vec{x}_n))(\vec{x}_1, \dots, \vec{x}_n)[\alpha],$$

where $\vec{x}_j = x_{j1} \dots x_{jk}$.

In Sections 4.5 and 4.6, I relate the logics $\mathbf{D}_k(\tau)$ to fragments of Σ_1^1 . One of the main results in this regard is a characterization of $\mathbf{D}_k(\tau)$ in terms of existential, second-order logic. The relevant fragments are defined below.

4.3.3. DEFINITION. Let $\Sigma_{n,k}^1(\tau)$ be the fragment of $\Sigma_n^1(\tau)$ whose relation variables have arity k . Particular interest will be with the fragments $\Sigma_{1,k}^1(\tau)$, that are called *k-ary, existential, second-order logic*. If k equals one we arrive at *monadic, existential, second-order logic*: $\Sigma_{1,1}^1(\tau) = \mathbf{M}\Sigma_1^1(\tau)$.

The following proposition confirms the intuition that partially ordered connectives are restricted Henkin quantifiers. A similar result was provided also in (Sandu and Väänänen 1992) and (Hella and Sandu 1995).

4.3.4. PROPOSITION. $\mathbf{D} \leq \mathbf{H}$ and $\mathcal{L}(\mathbf{D}) \leq \mathcal{L}(\mathbf{H})$.

Proof. It suffices to see that

$$\left(\begin{array}{cccc} \forall x_{11} & \dots & \forall x_{1k} & \forall i_1 \\ \vdots & \ddots & \vdots & \vdots \\ \forall x_{1k} & \dots & \forall x_{nk} & \forall i_n \end{array} \right) \Gamma(\vec{i})(\vec{x})$$

is equivalent to

$$\left(\begin{array}{cccc} \forall x_{11} & \dots & \forall x_{1k} & \exists y_1 \\ \vdots & \ddots & \vdots & \vdots \\ \forall x_{1k} & \dots & \forall x_{nk} & \exists y_n \end{array} \right) (\phi \rightarrow \Phi_1) \wedge (\neg\phi \rightarrow \Phi_2),$$

where

$$\begin{aligned} \phi &= \forall z_0 \forall z_1 (z_0 = z_1) \\ \Phi_1 &= \left(\bigvee_{i_1 \dots i_n \in \{0,1\}^n} \Gamma(\vec{i})(\vec{x}) \right) \\ \Phi_2 &= \exists z_0 \exists z_1 \left(z_0 \neq z_1 \wedge \bigvee_{i_1 \dots i_n \in \{0,1\}^n} \left(y_1 = z_{i_1} \wedge \dots \wedge y_n = z_{i_n} \wedge \Gamma(\vec{i})(\vec{x}) \right) \right), \end{aligned}$$

The antecedent ϕ discriminates between structures with exactly one object and the rest. In case there are two or more objects in the structure, all variables y_1, \dots, y_n must be assigned objects from z_0 and z_1 to mimic indices being assigned objects from $\{0, 1\}$. Note that this translation is slightly different from the one given in (Sandu and Väänänen 1992, pg. 363), reappearing in (Hella and Sandu 1995, pg. 80). The translations given in the latter publications omit the antecedent ϕ , which renders them flawed. \square

4.4 Related research

In this section I review four studies that are of direct relevance and/or help to put to the content of this chapter into perspective. More has been published on partially ordered connectives though, and I invite the reader to consult (Krynicky and Mostowski 1995, pp. 227-8) for references.

4.4.1 D and complete problems

One of the reasons why $\mathbf{M}\Sigma_1^1$ drew attention is due to the fact that it expresses various natural **NP**-complete problems, including 3-COLORABILITY and SAT. Interestingly, also \mathbf{D}_1 was shown to express **NP**-complete problems in the paper *Henkin quantifiers and complete problems* by Blass and Gurevich (1986)—be it that they adopt a somewhat different vocabulary.

4.4.1. THEOREM (BLASS AND GUREVICH (1986)). *The expression complexity of \mathbf{D}_1 is NP-complete.*

The latter result was shown by reference to a formula of the form $D_1^3 \gamma$, that defines the class of boolean circuits built from binary NAND-gates that have output *true* for some inputs. Theorem 4.4.1 shows that already D_1 carries the computational interest of \mathbf{NP} , just like $\mathbf{M}\Sigma_1^1$.

Blass and Gurevich show that restricting the height of the partially ordered connectives in \mathbf{D} has considerable computational impact.

4.4.2. THEOREM (BLASS AND GUREVICH (1986)). *The expression complexity of the logic $\bigcup_k D_k^2$ is \mathbf{NL} -complete.²*

Among the applications given for partially ordered connectives, Blass and Gurevich show that if ϕ is a first-order τ -formula, then the *reflexive, transitive closure* of ϕ , denoted $TC(\phi)$, is definable in $\mathcal{L}(\mathbf{D})(\tau)$. In particular, on digraphs $\mathcal{L}(\mathbf{D})(\{R\})$ can define the reflexive, transitive closure of R . The formula that does so will return in Examples 4.6.2 and 4.6.9.

4.4.2 Partially ordered connectives coined

Partially ordered connectives were defined under this header in the paper *Partially ordered connectives* by Sandu and Väänänen (1992). In this publication the model-theoretic properties of several partially ordered connectives are studied, including the quantifiers

$$\left(\begin{array}{l} \forall x \quad \bigvee_{i \in I} \\ \forall y \quad \bigvee_{j \in J} \end{array} \right) \quad \text{and} \quad \left(\begin{array}{l} \forall x \quad \exists z \\ \forall y \quad \bigvee_{i \in I} \end{array} \right),$$

which are called $D_{1,1}(I, J)$ and $D_{(1),1}(I)$, respectively. (In case the semantics of these quantifiers is unclear, see Section 4.4.3 for a definition.) Note that Sandu and Väänänen present partially ordered connectives as a generalization of Henkin quantifiers, by allowing for partially ordered first-order quantifiers *and* connectives. By contrast, observe that in the definition of $\mathcal{L}(\mathbf{D})$ partially ordered connectives do not contain existential quantifiers. The sets I, J define the universes over which i, j range.³ One way to appreciate the authors' paper is to realize that already the slightest relaxation of the linear ordering increases the expressive power of the logic at hand. For instance, the authors give an implicit matrix formula γ , such that $\exists u D_{1,1} \gamma$ characterizes the standard model of arithmetic up to isomorphism.

The authors provide an *Ehrenfeucht-Fraïssé game* for logics with partially ordered connectives in order to prove non-expressibility results. The connection with second-order logics may become clearer by comparing the respective

²The result was stated in terms of \mathbf{coNL} , as it was unknown at the time that $\mathbf{NL} = \mathbf{coNL}$, see also Theorem 2.4.2 in Chapter 2.

³In this chapter, these sets are fixed to $\{0, 1\}$. We work with logics in which the height of the quantifiers is unbounded. It follows from this fact that the restriction to $\{0, 1\}$ does not affect the expressive power of the logics at hand.

Ehrenfeucht-Fraïssé games. I modify Sandu and Väänänen's games in Section 4.8 so as to apply them to the logic \mathbf{D} .

4.4.3 Normal form theorem for Henkin quantifiers

It was shown in *Hierarchies of finite partially ordered connectives and quantifiers* by Krynicky (1993) that every partially ordered quantifier can be expressed by a partially ordered quantifier with only two rows. For future usage let me state this theorem more precisely. To this end, let $\mathbf{V}_k \vec{x}\vec{y}zi$ denote a quantifier prefix of the form:

$$\left(\begin{array}{cccc} \forall x_1 & \dots & \forall x_k & \exists z \\ \forall y_1 & \dots & \forall y_k & \bigvee i \in \{0, 1\} \end{array} \right),$$

Let τ be a vocabulary. Then, let $\mathbf{V}(\tau)$ be the language generated by the following grammar:

$$\Gamma ::= \gamma \mid \mathbf{V}_k \gamma,$$

where γ ranges over the implicit matrix τ -formulae and k over the integers. Let \mathfrak{A} be a τ -structure and let α be an assignment in \mathfrak{A} . Then, the semantics of \mathbf{V}_k is defined such that

$$\mathfrak{A} \models \mathbf{V}_k \vec{x}\vec{y}zi \gamma(i)(\vec{x}, \vec{y}, z)[\alpha] \quad \text{iff} \quad \mathfrak{A} \models \forall \vec{x} \forall \vec{y} \gamma(f(\vec{y}))(\vec{x}, \vec{y}, g(\vec{x}))[\alpha],$$

for some $f : A^k \rightarrow \{0, 1\}$ and $g : A^k \rightarrow A$.

4.4.3. THEOREM (KRYNICKI (1993)). $\mathbf{V} = \mathbf{H}$.

Thus Krynicky gives a very strong normal form result for partially ordered quantifiers. The implications of this result for Σ_1^1 are explored in the concluding section of this chapter, Section 4.11.

4.4.4 Finite model theory for IF logic

In *The logic of informational independence and finite models* Sandu (1997) departs from the insight that IF logic and Σ_1^1 have equal expressive power. Therefore, every \mathbf{NP} -decidable property can be expressed by an IF-sentence. Sandu expresses several \mathbf{NP} -complete properties in IF logic, such as satisfiability for a Boolean circuit, 3-COLORABILITY, and the Hamiltonian path problem.

Interestingly, some of the characterizations are obtained by reference to a theorem, that states that certain monadic, second-order sentences have an equivalent IF-sentence, in which only disjunctions (and no existential quantifiers) are independent from universal quantifiers. Roughly speaking, the result holds for those

$\mathbf{M}\Sigma_1^1$ -sentences that state that there exists a partition of the universe into sets P_1, \dots, P_n , such that

$$\forall x_1 \dots \forall x_k \bigwedge_B \left(\bigwedge B \rightarrow \phi_B \right),$$

where B ranges over all subsets in $\{P_i(x_j) \mid 1 \leq i \leq n, 1 \leq j \leq k\}$ and the ϕ_B are first-order formulae.

The methodology adopted by Sandu (1997) resembles the one adopted in the current chapter, as I will isolate the expressively strongest fragment of second-order logic that can be translated into \mathbf{D}_k . The results put forward in this chapter can thus be seen as a strengthening of the one from (Sandu 1997).

4.5 Translating \mathbf{D}_k into $\Sigma_{1,k}^1$

In this section I give a translation from \mathbf{D}_k into $\Sigma_{1,k}^1$, that hinges on the insight that a function $f : A \rightarrow \{0, 1\}$ can be mimicked by the set $X = \{a \in A \mid f(a) = 1\}$.

4.5.1. DEFINITION. Let \vec{x} be a string of k variables from VAR and let $X \in \mathbf{R}\text{-VAR}$ be a k -ary relation variable. Then, $\langle X, \vec{x} \rangle$ is a *proto-literal* and the formulae $X(\vec{x})$, $\neg X(\vec{x})$ are the *literals* based on $\langle X, \vec{x} \rangle$. Likewise, if L is a set of proto-literals, then the *set of literals based on L* is defined as $\{X(\vec{x}), \neg X(\vec{x}) \mid \langle X, \vec{x} \rangle \in L\}$. If Φ is a second-order formula, then $L(\Phi)$ is the *set of proto-literals of Φ* , where

$$L(\Phi) = \{\langle X, x_1, \dots, x_k \rangle \mid X(x_1, \dots, x_k) \text{ appears in } \Phi\}.$$

Finally, for $\mathbf{D} = \mathbf{D}_k^n \vec{x}_1 \dots \vec{x}_n i_1 \dots i_n$ let $L(\mathbf{D})$ be defined as $\{\langle X_j, \vec{x}_j \rangle \mid 1 \leq j \leq n\}$.

4.5.2. DEFINITION. Let $L = \{\langle Y_1, \vec{y}_1 \rangle, \dots, \langle Y_m, \vec{y}_m \rangle\}$ be a set of proto-literals, and let $\gamma : \{0, 1\}^m \rightarrow \mathbf{FO}$ be an implicit matrix formula. Then, the *L -explication* of γ is defined as

$$T_L(\gamma) = \bigwedge_{i_1 \dots i_m \in \{0, 1\}^m} (\pm_{i_1} Y_1(\vec{y}_1) \wedge \dots \wedge \pm_{i_m} Y_m(\vec{y}_m) \rightarrow \gamma(i_1, \dots, i_m)(\vec{y}_1, \dots, \vec{y}_m)),$$

where $\pm_0 = \neg$ and $\pm_1 = \neg\neg$.

The *standard translation* T maps every $\mathbf{D}_k(\tau)$ -formula $\Gamma = \mathbf{D}_k^n \gamma$ to the $\Sigma_{1,k}^1(\tau)$ -formula $T(\Gamma)$, such that

$$T(\Gamma) = \exists X_1 \dots \exists X_n \forall \vec{x}_1 \dots \forall \vec{x}_n T_{L(\mathbf{D}_k^n)}(\gamma).$$

Observe that if \vec{x} are the free variables of the \mathbf{D} -formula Γ , then \vec{x} are exactly the free variables in $T(\Gamma)$.

4.5.3. PROPOSITION. *Every \mathbf{D}_k -formula Γ is equivalent to $T(\Gamma)$.*

Proof. Let $\Gamma = \mathbf{D}_k^n \gamma$ be a \mathbf{D}_k -formula and let $T(\Gamma)$ be its standard translation as defined above. To prove that they are equivalent, consider a suitable structure \mathfrak{A} , and an assignment α in \mathfrak{A} .

Suppose $\mathfrak{A} \models \Gamma$. Then, there exist functions f_1, \dots, f_n of type $A^k \rightarrow \{0, 1\}$ such that

$$\mathfrak{A} \models \forall \vec{x}_1 \dots \forall \vec{x}_n \gamma(f_1(\vec{x}_1), \dots, f_n(\vec{x}_n))[\alpha]. \quad (4.6)$$

It remains to be shown that $\mathfrak{A} \models T(\Gamma)$, that is, that

$$\mathfrak{A} \models \exists X_1 \dots \exists X_n \forall \vec{x}_1 \dots \forall \vec{x}_n T_{L(\mathbf{D}_k^n)}(\gamma)[\alpha].$$

To this end, choose the interpretations $X_1^{\mathfrak{A}}, \dots, X_n^{\mathfrak{A}}$ of the k -ary relation variables in such a way that

$$\langle a_1, \dots, a_k \rangle \in X_i^{\mathfrak{A}} \quad \text{iff} \quad f_i(a_1, \dots, a_k) = 1, \quad (4.7)$$

for every k -tuple $\langle a_1, \dots, a_k \rangle$ of objects from A . Let $\vec{x}_1^{\mathfrak{A}}, \dots, \vec{x}_n^{\mathfrak{A}}$ be an arbitrary assignment of the variables in A . It suffices to show that it is the case that

$$\mathfrak{A} \models T_{L(\mathbf{D}_k^n)}(\gamma)[\alpha.X_1^{\mathfrak{A}}, \dots, X_n^{\mathfrak{A}}, \vec{x}_1^{\mathfrak{A}}, \dots, \vec{x}_n^{\mathfrak{A}}]. \quad (4.8)$$

The formula $T_{L(\mathbf{D}_k^n)}(\gamma)$ is a conjunction of implications, whose antecedents exhaust the set $\bigcup_{i_1 \dots i_n \in \{0, 1\}^n} \{\pm_{i_1} X_1(\vec{x}_1) \wedge \dots \wedge \pm_{i_n} X_n(\vec{x}_n)\}$. Thus every assignment makes true exactly one of $T_{L(\mathbf{D}_k^n)}(\gamma)$'s antecedents. This antecedent's consequent is $\gamma^* = \gamma(t_1, \dots, t_n)$, where

$$t_j = \begin{cases} 1 & \text{if } \vec{x}_j^{\mathfrak{A}} \in X_j^{\mathfrak{A}} \\ 0 & \text{if } \vec{x}_j^{\mathfrak{A}} \notin X_j^{\mathfrak{A}}. \end{cases}$$

Hence, in order to show that (4.8) it suffices to show that γ^* is true on \mathfrak{A} under the present assignment.

The interpretations $X_j^{\mathfrak{A}}$ were based on f_j . Therefore derive from the fact that the latter functions are witnesses of (4.6) that

$$\mathfrak{A} \models \gamma^*[\alpha.\vec{x}_1^{\mathfrak{A}}, \dots, \vec{x}_n^{\mathfrak{A}}].$$

as required.

The converse direction is similar. □

4.6 A characterization of \mathbf{D}_k

In this section, I give a characterization of \mathbf{D}_k as a fragment of $\Sigma_{1,k}^1$. This characterization result will facilitate several expressibility results for \mathbf{D} in this chapter.

4.6.1. DEFINITION. Let τ be a vocabulary. Let Φ be a second-order τ -formula. Call Φ *sober* if in Φ no second-order quantifier appears and for every $X \in \mathbf{R}\text{-VAR}_n$, $X(x_1, \dots, x_n)$ occurring in Φ implies that the variables x_1, \dots, x_n are free in Φ . Let $\Sigma_{1,k}^1 \heartsuit(\tau)$ contain exactly those $\Sigma_{1,k}^1(\tau)$ -formulae without free relation variables, that are of the form $\exists X_1 \dots \exists X_m \forall x_1 \dots \forall x_n \Phi$, where Φ is a sober formula. Put $\Sigma_1^1 \heartsuit(\tau) = \bigcup_k \Sigma_{1,k}^1 \heartsuit(\tau)$.

4.6.2. EXAMPLE. Consider the following $\Sigma_1^1(\{R\})$ -formulae:

$$\exists X \forall x \forall x' (R(x, x') \rightarrow \neg(X(x) \leftrightarrow X(x'))) \quad (4.9)$$

$$\exists X \forall x \forall x' (X(u) \wedge \neg X(v) \wedge (X(x) \wedge R(x, x') \rightarrow X(x'))) \quad (4.10)$$

$$\exists S \exists X \forall x \forall x' \forall x'' (\text{“}S \text{ is a linear order”} \wedge \Phi_1 \wedge \Phi_2), \quad (4.11)$$

where

$$\Phi_1 = ((\forall y S(x, y)) \rightarrow X(x)) \wedge ((\forall y S(y, x)) \rightarrow \neg X(x))$$

$$\Phi_2 = (S(x, x') \wedge \forall y (S(y, x) \vee x=y \vee y=x' \vee S(x', y))) \rightarrow \neg(X(x) \leftrightarrow X(x')).$$

Observe that (4.9) and (4.10) are $\Sigma_1^1 \heartsuit(\{R\})$ -formulae. Note that (4.9) characterizes the 2-colorable $\{R\}$ -structures and that (4.10) is true on an $\{R\}$ -structure \mathfrak{A} under the assignment α in \mathfrak{A} iff there is no R -path in \mathfrak{A} leading from $\alpha(u)$ to $\alpha(v)$, cf. (Immerman 1999, pg. 129).

Although all first-order variables in (4.11) are bound by a universal quantifier, it is not a formula in $\Sigma_1^1 \heartsuit$. This is not due to the formula that expresses that S is a linear order, since S can be forced to be anti-reflexive, transitive, and total using only the quantifiers $\forall x, \forall x', \forall x''$ that follow $\exists S \exists X$. The formulae Φ_1 and Φ_2 ruin membership of $\Sigma_1^1 \heartsuit$ as they contain the atoms $S(x, y)$ and $S(y, x)$, that hold the variable y which is not bound by any of the quantifiers $\forall x, \forall x', \forall x''$. The quantifier $\forall y$ can be extracted from $((\forall y S(x, y)) \rightarrow X(x))$. But the result of this action is $\exists y (\neg S(x, y)) \wedge X(x)$, in which y existentially quantified.

Observe that (4.11) characterizes the finite structures \mathfrak{A} with even universe A . It does so by imposing a linear order S on the universe and stating that there exists a subset X of A that contains the S -minimal element, that does not contain the S -maximal element, such that for any two S -neighboring objects precisely one of them sits in X . \square

The characterization result, Theorem 4.6.7, says that $\mathbf{D}_k = \Sigma_{1,k}^1 \heartsuit$. The structure of the proof of Theorem 4.6.7 is as follows: The standard translation T accounts for the inclusion of \mathbf{D}_k in $\Sigma_{1,k}^1 \heartsuit$. Conversely, Lemma 4.6.5 shows

that a fragment of $\Sigma_{1,k}^1 \heartsuit$ can be translated to \mathbf{D}_k . In the proof of Theorem 4.6.7 we then see that this particular fragment of $\Sigma_{1,k}^1 \heartsuit$ is equally expressive as full $\Sigma_{1,k}^1 \heartsuit$.

Let S_L be the set of literals based on the set of proto-literals L , cf. Definition 4.5.1. Call $S \subseteq S_L$ a *maximally consistent subset* of S_L , if S does not contain both a literal and its negation, but adding any literal based on L to S would cause it to contain both a literal and its negation. Put differently, S is a maximally consistent subset of S_L , if for every $\langle X, \vec{x} \rangle \in L$, either $X(\vec{x})$ or $\neg X(\vec{x})$ is in S .

4.6.3. LEMMA. *Let τ be a vocabulary. Let Φ be a sober second-order τ -formula and let $L(\Phi)$ be the set of proto-literals of Φ . Then, Φ is equivalent to a formula of the form*

$$\bigwedge_S \left(\bigwedge S \rightarrow \phi_S \right),$$

where S ranges over the maximally consistent subsets of $S_{L(\Phi)}$ and the ϕ_S s are $\mathbf{FO}(\tau)$ -formulae. Call the latter formula the explicit matrix formula of Φ and let it be denoted by $M(\Phi)$.

Proof. Let Φ be as in the premise of the lemma and let $L(\Phi)$ be the set of literals based on $L(\Phi)$. Per maximally consistent subset S of $S_{L(\Phi)}$, obtain the formula ϕ_S from Φ by replacing every occurrence of $X(\vec{x})$ in Φ by \top , if $X(\vec{x}) \in S$ and by \perp if $X(\vec{x}) \notin S$. Now, consider Φ 's explicit matrix formula $M(\Phi) = \bigwedge_S (\bigwedge S \rightarrow \phi_S)$, and observe that it is of the desired form. It remains to be shown that (i) every ϕ_S is a $\mathbf{FO}(\tau)$ -formula and that (ii) $M(\Phi)$ and Φ are equivalent.

Ad (i): If Φ is a sober τ -formula and moreover contains no relation variables, then it is a first-order formula in the vocabulary τ . The formula ϕ_S is obtained from Φ by replacing all relation variables by the symbols \top and \perp . Hence ϕ_S is a $\mathbf{FO}(\tau)$ -formula.

Ad (ii): It suffices to show that for an arbitrary τ -structure \mathfrak{A} and assignment α in \mathfrak{A} , it is the case that $\mathfrak{A} \models \Phi[\alpha]$ iff $\mathfrak{A} \models M(\Phi)[\alpha]$. Note that if $X(\vec{x})$ occurs in Φ , where X is a relation variable, then all variables X, \vec{x} are free in Φ , and the variables \vec{x} are free in $M(\Phi)$.

Observe that there exists exactly one maximally consistent subset S of $S_{L(\Phi)}$ such that $\mathfrak{A} \models \bigwedge S[\alpha]$, namely the one that contains $X(\vec{x})$ if $\mathfrak{A} \models X(\vec{x})[\alpha]$, and $\neg X(\vec{x})$ if $\mathfrak{A} \not\models X(\vec{x})[\alpha]$. Since all other maximally consistent subsets S' from $L(\Phi)$ have it that $\mathfrak{A} \not\models \bigwedge S'[\alpha]$, to show that $M(\Phi)$ and Φ are equivalent, it suffices to show that $\mathfrak{A} \models \Phi[\alpha]$ iff $\mathfrak{A} \models \phi_S[\alpha]$. The latter equivalence can be proved by means of an inductive argument on the structure of Φ .

Base step. Suppose $\Phi = R(x_1, \dots, x_n)$. Then, $L(\Phi) = \emptyset$ and ϕ_S is simply Φ itself.

Suppose $\Phi = X(x_1, \dots, x_n)$. Observe the following string of equivalences:

$$\mathfrak{A} \models X(\vec{x})[\alpha] \quad \text{iff} \quad X(\vec{x}) \in L(\Phi) \quad \text{iff} \quad \phi_S = \top \quad \text{iff} \quad \mathfrak{A} \models \phi_S[\alpha].$$

Induction step. Trivial. \square

4.6.4. EXAMPLE. Let us go through an example involving 2-COLORABILITY to get intuitions straight. Consider the formula $\exists Y \exists Y' \forall x \forall x' \Xi$, that expresses 2-COLORABILITY over graphs—be it in a somewhat contrived manner—, where Ξ is specified as follows:

$$(x = x' \rightarrow (Y(x) \leftrightarrow Y'(x'))) \wedge (R(x, x') \rightarrow \neg(Y(x) \leftrightarrow Y'(x'))).$$

The first conjunct expresses that Y and Y' are given the same extension. Since Ξ does not contain quantifiers it is sober. Consider the set of proto-literals $L(\Xi) = \{\langle Y, x \rangle, \langle Y', x' \rangle\}$. There are four different maximally consistent subsets of $S_{L(\Xi)}$: $\{Y(x), Y'(x')\}$, $\{\neg Y(x), Y'(x')\}$, $\{Y(x), \neg Y'(x')\}$, and $\{\neg Y(x), \neg Y'(x')\}$. Per maximally consistent subset S of $S_{L(\Xi)}$, I write ξ_S (modulo truth-preserving rewriting):

$$\begin{aligned} \xi_{\{Y(x), Y'(x')\}} &= \neg R(x, x') \\ \xi_{\{\neg Y(x), Y'(x')\}} &= (x \neq x') \\ \xi_{\{Y(x), \neg Y'(x')\}} &= (x \neq x') \\ \xi_{\{\neg Y(x), \neg Y'(x')\}} &= \neg R(x, x'). \end{aligned}$$

\square

The following lemma shows that every $\Sigma_1^1 \heartsuit$ -formula that meets two conditions (A) and (B)—to be formulated—is equivalent to a \mathbf{D} -formula. Theorem 4.6.7 then shows that every $\Sigma_1^1 \heartsuit$ -formula has an equivalent $\Sigma_1^1 \heartsuit$ -formula that satisfies conditions (A) and (B).

4.6.5. LEMMA. *Let τ be a vocabulary. Let Φ be a sober τ -formula containing the relation variables X_1, \dots, X_n , such that*

(A) X_1, \dots, X_n are k -ary; and

(B) if $X_i(x_1, \dots, x_k)$ and $X_j(x'_1, \dots, x'_k)$ appear in Φ , then $i \neq j$ or $x_h = x'_h$, for every $1 \leq h \leq k$.

Then, (1) and (2) hold:

(1) There exists an implicit matrix τ -formula γ such that $T_{L(\Phi)}(\gamma)$ and Φ are equivalent.

(2) There exists a $\mathbf{D}_k(\tau)$ -formula that is equivalent to $\exists X_1 \dots \exists X_n \forall \vec{x}_1 \dots \forall \vec{x}_n \Phi$.

Proof. Let Φ meet the premise of the lemma and let $L(\Phi)$ be the set of proto-literals $\{\langle X_1, \vec{x}_1 \rangle, \dots, \langle X_n, \vec{x}_n \rangle\}$.

Ad (1): Since Φ is sober, derive from Lemma 4.6.3 that it can be rewritten into the explicit matrix formula $M(\Phi) = \bigwedge_S (\bigwedge S \rightarrow \phi_S)$, where the formulae ϕ_S are first-order. Now we need to obtain an implicit matrix formula γ , so that $T_{L(\Phi)}(\gamma)$ and $M(\Phi)$ are equivalent. This can be done by putting $\gamma(i_1, \dots, i_n) = \phi_{S_{i_1 \dots i_n}}$, where

$$S_{i_1 \dots i_n} = \{\pm_{i_1} X_1(\vec{x}_1), \dots, \pm_{i_n} X_n(\vec{x}_n)\},$$

for $\pm_1 = \neg \neg$ and $\pm_0 = \neg$. Following definition 4.5.2, $T_{L(\Phi)}(\gamma)$ equals

$$\bigwedge_{i_1 \dots i_n \in \{0,1\}^n} (\pm_{i_1} X_1(\vec{x}_1) \wedge \dots \wedge \pm_{i_n} X_n(\vec{x}_n) \rightarrow \gamma(i_1, \dots, i_n)). \quad (4.12)$$

Since every string $i_1 \dots i_n \in \{0,1\}^n$ corresponds thus with a maximally consistent set of proto-literals, and vice versa, (4.12) is a syntactic copy of the explicit matrix formula $M(\Phi)$. From Lemma 4.6.3 it follows that $T_{L(\Phi)}(\gamma)$ is equivalent to Φ .

Ad (2): Consider the $\Sigma_{1,k}^1(\tau)$ -formula $\Psi = \exists X_1 \dots \exists X_n \forall \vec{x}_1 \dots \forall \vec{x}_n \Phi$. By clause (1) there exists a matrix formula γ , such that $T_{L(\Phi)}(\gamma)$ and Φ are equivalent. Consider the formula $\Gamma = \mathbf{D}_k^n \vec{x}_1 \dots \vec{x}_n \vec{i} \gamma$ and its standard translation $T(\Gamma)$:

$$\exists X_1 \dots \exists X_n \forall \vec{x}_1 \dots \vec{x}_n T_{L(\Phi)}(\gamma)$$

From (A) and (B) it follows that $L(\mathbf{D}_k^n \vec{x}_1 \dots \vec{x}_n \vec{i}) = L(\Phi)$. Hence, $T(\Gamma)$ is a syntactic copy of Ψ , and $T(\Gamma)$ is equivalent to Γ in virtue of Proposition 4.5.3. \square

4.6.6. EXAMPLE. (Continuation of Example 4.6.4) Consider the explicit matrix formula of Ξ from Example 4.6.4:

$$M(\Xi) = \left[\begin{array}{l} Y(x) \wedge Y'(x') \rightarrow \neg R(x, x') \\ Y(x) \wedge \neg Y'(x') \rightarrow (x \neq x') \\ \neg Y(x) \wedge Y'(x') \rightarrow (x \neq x') \\ \neg Y(x) \wedge \neg Y'(x') \rightarrow \neg R(x, x') \end{array} \right].$$

Construct the matrix formula γ as described in Lemma 4.6.5 by setting

$$\begin{aligned} \gamma(1, 1) = \gamma(0, 0) &= \neg R(x, x') \\ \gamma(1, 0) = \gamma(0, 1) &= (x \neq x'). \end{aligned}$$

The formula $T_{\{(Y,x), (Y',x')\}}(\gamma)$ is a syntactic copy of $M(\Xi)$. Furthermore, observe that the formulae $\exists Y \exists Y' \forall x \forall x' T_{\{(Y,x), (Y',x')\}}(\gamma)$ and $\mathbf{D}_1^2 \gamma$ are equivalent, as the standard translation applied to the latter formula yields the former. Hence, $\mathbf{D}_1^2 \gamma$ expresses 2-COLORABILITY on graphs. \square

Theorem 4.6.7 shows that every $\Sigma_{1,k}^1 \heartsuit$ -formula has an equivalent $\Sigma_{1,k}^1 \heartsuit$ -formula meeting (B), and ties the previous results together.

4.6.7. THEOREM. *For every integer k , $\mathbf{D}_k = \Sigma_{1,k}^1 \heartsuit$. Hence, $\mathbf{D} = \Sigma_1^1 \heartsuit$.*

Proof. From left to right. This direction follows immediately from the standard translation as it maps every formula in $\mathbf{D}_k(\tau)$ to a formula in $\Sigma_{1,k}^1 \heartsuit(\tau)$. The standard translation was proved correct in Proposition 4.5.3.

From right to left. Consider a $\Sigma_{1,k}^1 \heartsuit$ -formula $\Phi = \exists X_1 \dots \exists X_m \forall x_1 \dots \forall x_n \Phi'$. By definition of $\Sigma_{1,k}^1 \heartsuit$, Φ' is sober and X_1, \dots, X_m are k -ary.

Let us make the following assumptions about Φ , that go without loss of generality:

- (i) Every relation variable X_1, \dots, X_m actually appears in Φ' , that is, none of the second-order quantifiers $\exists X_1, \dots, \exists X_m$ is vacuous in Φ .
- (ii) If $X_i(y_1, \dots, y_k)$ appears in Φ' , then for every variable it is the case that $y_j \in \{x_1, \dots, x_n\}$. To see that this goes without loss of generality observe that if the variable y is the argument of X_i but is not in $\{x_1, \dots, x_n\}$, then Φ' is equivalent to $\forall z (y = z \rightarrow \Phi'[y \mapsto z])$, where $\Phi'[y \mapsto z]$ is the formula that results from replacing every occurrence of y in Φ by z . The variable y does not appear in $\Phi'[y \mapsto z]$ anymore, so in particular it cannot be the argument of X_i . For an example see Example 4.6.9.

It suffices to show that there exists a $\Sigma_{1,k}^1 \heartsuit$ -formula Ψ , such that:

- (1) $\Psi = \exists Y_1 \dots \exists Y_l \forall \vec{y}_1 \dots \forall \vec{y}_l \Psi'$.
- (2) If $Y_i(y_{i1}, \dots, y_{ik})$ and $Y_j(y_{j1}, \dots, y_{jk})$ appear in Φ , then $i \neq j$ or $y_{ih} = y_{jh}$, for every $1 \leq h \leq k$.
- (3) Ψ is equivalent to Φ .

For if such a Ψ exists, then Ψ' is sober and the set of proto-literals of Ψ , $L(\Psi) = \{ \langle Y_i, \vec{y}_i \rangle \mid 1 \leq i \leq l \}$, meets the premises of Lemma 4.6.5, due to condition (2). Thus Lemma 4.6.5.2 applies and yields that there exists a $\mathbf{D}_k(\tau)$ -formula in equivalent to Ψ . In turn, Ψ is equivalent to Φ in virtue of (3).

I prove that such a Ψ exists by an inductive argument on the *relation variable rank* of Φ , denoted $rvr(\Phi)$, that is defined as the cardinality of $L(\Phi)$. Let R-VAR^Φ be the set of relation variables in Φ . By the assumption that Φ satisfies (i) it is the case that $rvr(\Phi) \geq \text{R-VAR}^\Phi$. In fact, if Ψ is a $\Sigma_{1,k}^1 \heartsuit$ -formula such that $rvr(\Psi) - \|\text{R-VAR}^\Psi\| = 0$, then Ψ satisfies condition (2).

Thus, it suffices to provide a truth preserving procedure that decreases the relation variable rank of every $\Sigma_{1,k}^1 \heartsuit$ -formula by one, while preserving properties (i) and (ii). That is, it is sufficient to prove that for every $\Sigma_{1,k}^1 \heartsuit$ -formula

Φ such that (i), (ii), and $rvr(\Phi) - \|\text{R-VAR}^\Phi\| > 0$, there exists a $\Sigma_{1,k}^1\heartsuit(\tau)$ -formula Θ equivalent to Φ , where Θ meets (i), (ii), and $rvr(\Theta) - \|\text{R-VAR}^\Theta\| = rvr(\Phi) - \|\text{R-VAR}^\Phi\| - 1$.

For the sake of the argument, suppose that $rvr(\Phi) - \|\text{R-VAR}^\Phi\| > 0$. Firstly, derive from the definition of $\Sigma_{1,k}^1\heartsuit$ and condition (ii) that $\text{R-VAR}^\Phi = \{X_1, \dots, X_m\}$. Hence, there exists at least one relation variable $X_i \in \text{R-VAR}^\Phi$ such that $X_i(\vec{z})$ and $X_i(\vec{z}')$ appear in Φ , where \vec{z}, \vec{z}' are two strings of k variables such that

- $\vec{z} = z_1, \dots, z_k$ and $\vec{z}' = z'_1, \dots, z'_k$;
- $z_1, \dots, z_k, z'_1, \dots, z'_k \in \{x_1, \dots, x_n\}$ (in virtue of (iii)); and
- for some $1 \leq i \leq k$, z_i and z'_i are different variables.

Assume there are exactly two such appearances of $X_i(\vec{z})$ and $X_i(\vec{z}')$ in Φ . The proof is easily extended to arbitrary numbers $X_i(\vec{z}_1), X_i(\vec{z}_2), \dots$, but in the interest of readability I content myself with this restriction.

We get rid of $X_i(\vec{z}_i)$ and $X_i(\vec{z}'_i)$ by replacing the string $X_i(z_1, \dots, z_k)$ by the string $Y_i(y_{i1}, \dots, y_{ik})$ and $X_i(z'_1, \dots, z'_k)$ by $Y'_i(y'_{i1}, \dots, y'_{ik})$, where all the variables $Y_i, Y'_i, y_{i1}, \dots, y_{ik}, y'_{i1}, \dots, y'_{ik}$ are *fresh*: they do not yet occur in Φ . Let $\vec{y}_i = y_{i1}, \dots, y_{ik}$ and let $\vec{y}'_i = y'_{i1}, \dots, y'_{ik}$.

This idea underlies the syntactic operation on Φ , yielding Θ :

- Replace $\exists X_i$ by $\exists Y_i \exists Y'_i$.
- Add $\forall y_{i1} \dots \forall y_{ik} \forall y'_{i1} \dots \forall y'_{ik}$ to the left of $\forall x_1 \dots \forall x_n$.
- Replace Φ' by $M \rightarrow (\Phi' \wedge N)$, where

$$\begin{aligned} M &= \bigwedge_{1 \leq j \leq k} (y_{ij} = z_j \wedge y'_{ij} = z'_j) \\ N &= (y_{i1} = y'_{i1} \wedge \dots \wedge y_{ik} = y'_{ik}) \rightarrow (Y_i(\vec{y}_i) \leftrightarrow Y'_i(\vec{y}'_i)). \end{aligned}$$

- Replace $X_i(z_1, \dots, z_k)$ by $Y_i(\vec{y}_i)$ and $X_i(z'_1, \dots, z'_k)$ by $Y'_i(\vec{y}'_i)$.

To prove that the formulae Φ and Θ are equivalent, let \mathfrak{A} be a suitable structure, and let α be an assignment in \mathfrak{A} .

Suppose $\mathfrak{A} \models \Phi[\alpha]$. Let $X_1^{\mathfrak{A}}, \dots, X_m^{\mathfrak{A}}$ be such that

$$\mathfrak{A} \models \forall x_1 \dots \forall x_n \Phi'[\alpha.X_1^{\mathfrak{A}}, \dots, X_m^{\mathfrak{A}}].$$

I shall prove that $\mathfrak{A} \models \Theta[\alpha]$, by showing that

$$\mathfrak{A} \models \forall x_1 \dots \forall x_n \forall \vec{y}_i \forall \vec{y}'_i M \rightarrow (\Phi' \wedge N)[\alpha.X_1^{\mathfrak{A}}, \dots, X_m^{\mathfrak{A}}, Y_i^{\mathfrak{A}}, (Y'_i)^{\mathfrak{A}}],$$

where $Y_i^{\mathfrak{A}}, (Y'_i)^{\mathfrak{A}} = X_i^{\mathfrak{A}}$. Arbitrarily assign the objects $x_1^{\mathfrak{A}}, \dots, x_n^{\mathfrak{A}}, y_{i1}^{\mathfrak{A}}, \dots, y_{ik}^{\mathfrak{A}}, (y'_{i1})^{\mathfrak{A}}, \dots, (y'_{ik})^{\mathfrak{A}} \in A$ to the respective universally quantified variables and suppose that they make M true. Since $Y_i^{\mathfrak{A}}, (Y'_i)^{\mathfrak{A}}$ were chosen equal to $X_i^{\mathfrak{A}}$, N is true. Finally, in order to see that Φ' is true set up an inductive argument that builds on the fact that

$$\mathfrak{A} \models \Phi'[\alpha.X_1^{\mathfrak{A}}, \dots, X_m^{\mathfrak{A}}, x_1^{\mathfrak{A}}, \dots, x_n^{\mathfrak{A}}].$$

As a matter of fact, the only non-trivial case lies with Φ being of the form $X_i(z_1, \dots, z_k)$ in which case one has that

$$\mathfrak{A} \models X_i(z_1, \dots, z_k)[\alpha.X_1^{\mathfrak{A}}, \dots, X_m^{\mathfrak{A}}, x_1^{\mathfrak{A}}, \dots, x_n^{\mathfrak{A}}].$$

Since $Y_i^{\mathfrak{A}} = X_i^{\mathfrak{A}}$ and $z_j^{\mathfrak{A}} = y_{ij}^{\mathfrak{A}}$, for every $1 \leq j \leq k$, derive that

$$\mathfrak{A} \models Y_i(y_{i1}, \dots, y_{ik})[\alpha.X_1^{\mathfrak{A}}, \dots, X_m^{\mathfrak{A}}, Y_i^{\mathfrak{A}}, (Y'_i)^{\mathfrak{A}}, x_1^{\mathfrak{A}}, \dots, x_n^{\mathfrak{A}}, y_{i1}^{\mathfrak{A}}, \dots, y_{ik}^{\mathfrak{A}}].$$

The same argument applies to Y'_i and the converse direction is similar.

Derive that for the objects $\vec{x}^{\mathfrak{A}} = x_1^{\mathfrak{A}}, \dots, x_n^{\mathfrak{A}}, \vec{y}_i^{\mathfrak{A}} = y_{i1}^{\mathfrak{A}}, \dots, y_{ik}^{\mathfrak{A}}$, and $(\vec{y}'_i)^{\mathfrak{A}} = (y'_{i1})^{\mathfrak{A}}, \dots, (y'_{ik})^{\mathfrak{A}}$, it is the case that

$$\mathfrak{A} \models M \rightarrow (\Phi' \wedge N)[\alpha.X_1^{\mathfrak{A}}, \dots, X_m^{\mathfrak{A}}, Y_i^{\mathfrak{A}}, (Y'_i)^{\mathfrak{A}}, \vec{x}^{\mathfrak{A}}, \vec{y}_i^{\mathfrak{A}}, (\vec{y}'_i)^{\mathfrak{A}}].$$

Since the objects $\vec{x}^{\mathfrak{A}}, \vec{y}_i^{\mathfrak{A}}, (\vec{y}'_i)^{\mathfrak{A}}$ were assigned arbitrarily, obtain that

$$\mathfrak{A} \models \forall \vec{x} \forall \vec{y}_i \forall \vec{y}'_i (M \rightarrow (\Phi' \wedge N))[\alpha.X_1^{\mathfrak{A}}, \dots, X_m^{\mathfrak{A}}, Y_i^{\mathfrak{A}}, (Y'_i)^{\mathfrak{A}}].$$

Introduction of existential quantifiers yields

$$\mathfrak{A} \models \exists X_1 \dots \exists X_{i-1} \exists Y_i \exists Y'_i \exists X_{i+1} \dots \exists X_m \forall \vec{x} \forall \vec{y}_i \forall \vec{y}'_i M \rightarrow (\Phi' \wedge N)[\alpha].$$

Hence, $\mathfrak{A} \models \Theta[\alpha]$ as required.

As to the converse direction, suppose that $\mathfrak{A} \models \Theta[\alpha]$. Let the interpretations $X_1^{\mathfrak{A}}, \dots, X_m^{\mathfrak{A}}, Y_i^{\mathfrak{A}}, (Y'_i)^{\mathfrak{A}}$ be witnesses of this fact. First, I show that $Y_i^{\mathfrak{A}} = (Y'_i)^{\mathfrak{A}}$. For the sake of contradiction, assume that on \mathfrak{A} the formula Θ holds but $Y_i^{\mathfrak{A}} \neq (Y'_i)^{\mathfrak{A}}$. Then, without loss of generality, there exist k objects $\vec{y}'_i = y'_{i1}, \dots, y'_{ik}$ from A such that $\vec{y}'_i \in Y_i^{\mathfrak{A}}$ and $\vec{y}'_i \notin (Y'_i)^{\mathfrak{A}}$. Hence, it is the case that

$$\mathfrak{A} \models \exists \vec{y}_i \exists \vec{y}'_i (y_{i1} = y'_{i1} \wedge \dots \wedge y_{ik} = y'_{ik} \wedge Y_i(\vec{y}_i) \wedge \neg Y'_i(\vec{y}'_i))$$

and consequently that $\mathfrak{A} \not\models M$. But this contradicts Θ holding on \mathfrak{A} , since if x_1, \dots, x_n are assigned objects so that M is made true (and such an assignment trivially exists), then the conjunction of Φ' and N is false. Therefore $Y_i^{\mathfrak{A}} = (Y'_i)^{\mathfrak{A}}$.

Set $X_i^{\mathfrak{A}} = Y_i^{\mathfrak{A}}$. To show that

$$\mathfrak{A} \models \forall x_1 \dots \forall x_n \Phi'[\alpha.X_1^{\mathfrak{A}}, \dots, X_m^{\mathfrak{A}}]$$

pick arbitrary objects $x_1^{\mathfrak{A}}, \dots, x_n^{\mathfrak{A}}$ from A . By the same kind of argument laid down for the converse direction one shows that $\mathfrak{A} \models \Phi'[\alpha.X_1^{\mathfrak{A}}, \dots, X_m^{\mathfrak{A}}, x_1^{\mathfrak{A}}, \dots, x_n^{\mathfrak{A}}]$. Hence, $\mathfrak{A} \models \Phi[\alpha]$.

The proof comes to an end, after concluding that $rvr(\Theta) - \|\mathbf{R}\text{-VAR}^\Theta\| = rvr(\Phi) - \|\mathbf{R}\text{-VAR}^\Phi\| - 1$. To see that this is the case, we observe that $\mathbf{R}\text{-VAR}^\Theta = \{X_1, \dots, X_{i-1}, Y_i, Y'_i, X_{i+1}, \dots, X_m\}$, whence $\|\mathbf{R}\text{-VAR}^\Theta\| = \|\mathbf{R}\text{-VAR}^\Phi\| + 1$. Secondly, we observe that

$$L(\Phi) - L(\Theta) = \{\langle X_i, \bar{z} \rangle, \langle X_i, \bar{z}' \rangle\} \quad \text{and} \quad L(\Theta) - L(\Phi) = \{\langle Y_i, \bar{y} \rangle, \langle Y'_i, \bar{y}' \rangle\}.$$

Therefore, $rvr(\Phi) = rvr(\Theta)$ and $rvr(\Theta) - \|\mathbf{R}\text{-VAR}^\Theta\| = rvr(\Phi) - \|\mathbf{R}\text{-VAR}^\Phi\| - 1$. \square

4.6.8. EXAMPLE. (Continuation of Examples 4.6.2 and 4.6.4) Consider formula (4.9) from Example 4.6.2, copied below:

$$\Phi = \exists X \forall x \forall x' (R(x, x') \rightarrow \neg(X(x) \leftrightarrow X(x'))).$$

Firstly observe that $L(\Phi) = \{\langle X, x \rangle, \langle X, x' \rangle\}$ and that $\mathbf{R}\text{-VAR}^\Phi = \{X\}$. Hence, $rvr(\Phi) = \|L(\Phi)\| > \|\mathbf{R}\text{-VAR}^\Phi\|$. Going through the syntactic operations of the proof one time yields $\Theta = \exists Y \exists Y' \forall y \forall y' \forall x \forall x' (M \rightarrow (\Theta' \wedge N))$, where

$$\begin{aligned} \Theta' &= (R(x, x') \rightarrow \neg(Y(y) \leftrightarrow Y'(y'))) \\ M &= (x = y \wedge x' = y') \\ N &= (y = y' \rightarrow (Y(y) \leftrightarrow Y'(y))). \end{aligned}$$

Observe that $L(\Theta) = \{\langle Y, x \rangle, \langle Y', x' \rangle\}$ and that $\mathbf{R}\text{-VAR}(\Theta) = \{Y, Y'\}$, so the syntactic operation comes to an end outputting Θ . But in this case, there is a formula equivalent to Θ' that has only two first-order variables instead of Θ' 's four:

$$(x = x' \rightarrow (Y(x) \leftrightarrow Y'(x'))) \wedge (R(x, x') \rightarrow \neg(Y(x) \leftrightarrow Y'(x'))).$$

This sentence, expressing 2-COLORABILITY was the starting point of Example 4.6.4. For a $\Sigma_1^1\mathfrak{Q}$ -formula expressing 3-COLORABILITY see Example 2.5.2. \square

4.6.9. EXAMPLE. (Continuation of Example 4.6.2) Consider the formula (4.10) from Example 4.6.2, that expresses that there is no R -path in the $\{R\}$ -structure \mathfrak{A} under α from $\alpha(u)$ to $\alpha(v)$:

$$\exists X \forall x \forall x' (X(u) \wedge \neg X(v) \wedge (X(x) \wedge R(x, x') \rightarrow X(x'))).$$

Firstly, observe that the variables u and v are free. As remarked in clause (ii) in the proof of Theorem 4.6.7, this does not cause any significant problems, as the latter formula is equivalent to

$$\exists X \forall x \forall x' \forall y \forall y' ((u = y \wedge v = y') \rightarrow (X(y) \wedge \neg X(y') \wedge (X(x) \wedge R(x, x') \rightarrow X(x')))).$$

Now, every variable occurring as the argument of the only relation variable X is bound by one of the quantifiers $\forall x \forall x' \forall y \forall y'$. To enhance readability, let me stick to the following equivalent variant of the previous formula:

$$\Phi = \exists X \forall x \forall x' ((u = x \rightarrow X(x)) \wedge (v = x \rightarrow \neg X(x)) \wedge (X(x) \wedge R(x, x') \rightarrow X(x'))).$$

Observe that $L(\Phi) = \{\langle X, x \rangle, \langle X, x' \rangle\}$ and that $\text{R-VAR}^\Phi = \{X\}$. Hence, it is the case that $\|L(\Phi)\| > \|\text{R-VAR}^\Phi\|$. Following the syntactic operation on Φ as described in the proof, obtain $\Theta = \exists Y \exists Y' \forall y \forall y' \forall x \forall x' (M \rightarrow (\Theta' \wedge N))$, where

$$\begin{aligned} \Theta' &= (u = y \rightarrow Y(y)) \wedge (v = y \rightarrow \neg Y(y)) \wedge (Y(y) \wedge R(x, x') \rightarrow Y'(y')) \\ M &= (x = y \wedge x' = y') \\ N &= (y = y' \rightarrow (Y(y) \leftrightarrow Y'(y'))). \end{aligned}$$

Notice that $L(\Theta) = \{\langle Y, y \rangle, \langle Y', y' \rangle\}$ and that $\text{R-VAR}^\Theta = \{Y, Y'\}$. Lemma 4.6.5, holds that the implicit matrix formula $M(\forall x \forall x' \Theta')$ is equivalent to $\forall x \forall x' \Theta'$ and that $\mathbf{D}_1^2 y y' M(\forall x \forall x' \Theta')$ is equivalent to Θ and Φ . \square

The characterization of \mathbf{D} in second-order terms speeds up the finding of interesting properties it enjoys, for second-order logic happens to be more intensively studied than partially ordered connectives. Concrete—and relevant—examples of this mode of research can be found in the next section.

4.7 Applications of Theorem 4.6.7

In this section, I obtain two results using the characterization of \mathbf{D} . In Section 4.7.1, it is shown that $\mathbf{D}_k < \mathbf{D}_{k+1}$. In Section 4.7.2, I show that on linear ordered structures $\mathbf{D} = \Sigma_1^1$.

4.7.1 Strict hierarchy result

Using a result of Ajtai's (1983), I show that $\mathbf{D}_k < \mathbf{D}_{k+1}$, making use of Theorem 4.6.7. Put differently, $\mathbf{D}(\tau)$ contains a strict, arity induced hierarchy, even over finite structures.

4.7.1. THEOREM. *Let $k \geq 2$ be an integer and let σ be a vocabulary with at least one k -ary relation symbol P and the linear order symbol $>$. Then, over σ -structures, $\mathbf{D}_{k-1}(\sigma) < \mathbf{D}_k(\sigma)$.*

Proof. From (Ajtai 1983) the following can be derived: Let Π_k be the property over σ -structures \mathfrak{A} such that

$$\Pi_k(\mathfrak{A}) = \text{true} \quad \text{iff} \quad \|P^{\mathfrak{A}}\| \text{ is even.}$$

Then, Π_k is not expressible in $\Sigma_{1,k-1}^1(\sigma)$, but it is expressible in $\Sigma_{1,k}^1(\sigma)$.⁴

To separate \mathbf{D}_k from \mathbf{D}_{k-1} , I show that Π_k is expressible by a formula in $\mathbf{D}_k(\sigma)$. This is sufficient an argument for the current end, since

$$\mathbf{D}_{k-1} = \Sigma_{1,k-1}^1 \heartsuit \leq \Sigma_{1,k-1}^1$$

and $\Sigma_{1,k-1}^1$ cannot express Π_k .

I show that $\mathbf{D}_k(\sigma)$ can express Π_k by giving a $\Sigma_{1,k}^1 \heartsuit(\sigma)$ -formula Υ_k that expresses Π_k . Intuitively, Υ_k lifts the binary linear order symbol $>$ to a linear order relation ψ_k over k -tuples of objects. With respect to this lifted linear order Υ_k expresses that there exists a subset Q of k -tuples of objects from the domain such that:

- (1) Q is a subset of $P^{\mathfrak{A}}$.
- (2) The $\psi_k^{\mathfrak{A}}$ -minimal k -tuple that is in $P^{\mathfrak{A}}$ is also in Q and the $\psi_k^{\mathfrak{A}}$ -maximal k -tuple that is in $P^{\mathfrak{A}}$ is not in Q .
- (3) If two k -tuples are in $P^{\mathfrak{A}}$ and there is no k -tuple in between them (in the ordering constituted by $\psi_k^{\mathfrak{A}}$) that is in $P^{\mathfrak{A}}$, then exactly one of the k -tuples is in Q .

Here, ψ_k is a formula with free variables $x_1, \dots, x_k, y_1, \dots, y_k$; $\psi_k^{\mathfrak{A}}$ denotes the set

$$\{(a_1, \dots, a_k, b_1, \dots, b_k) \in A^{2k} \mid \mathfrak{A} \models \psi_k[x_1/a_1, \dots, x_k/a_k, y_1/b_1, \dots, y_k/b_k]\}.$$

Essentially, the idea underlying (1)-(3) resembles the one underlying the $\Sigma_1^1 \heartsuit$ -formula (4.11) from Example 4.6.2 expressing evenness of the universe.

All in all, put

$$\Upsilon_k = \exists Q \forall \vec{x} \forall \vec{y} (\Phi_1 \wedge \Phi_2 \wedge \Phi_3),$$

where Φ_i is the formula that was informally described in clause (i) above and \vec{x} and \vec{y} are strings of k variables. In the light of these descriptions, the following specifications are more or less self-explanatory:

$$\begin{aligned} \Phi_1 &= Q(\vec{x}) \rightarrow P(\vec{x}) \\ \Phi_2 &= (\text{MIN}_P(\vec{x}) \rightarrow Q(\vec{x})) \wedge (\text{MAX}_P(\vec{x}) \rightarrow \neg Q(\vec{x})) \\ \Phi_3 &= \text{NEXT}_P(\vec{x}, \vec{y}) \rightarrow \neg(Q(\vec{x}) \leftrightarrow Q(\vec{y})), \end{aligned}$$

⁴The result essentially uses *hypergraphs*, that is, structures interpreting relation symbols of unbounded arity. As a consequence, the result does not imply that $\Sigma_{1,2}^1(\tau)$ is strictly weaker than $\Sigma_{1,3}^1(\tau)$, where τ a vocabulary that contains only unary and binary predicates, cf. (Durand, Lautemann, and Schwentick 1998).

where

$$\begin{aligned} MIN_P(\vec{x}) &= \forall \vec{z} (P(\vec{z}) \rightarrow (\psi_k(\vec{x}, \vec{z}) \vee \vec{z} = \vec{x})) \\ MAX_P(\vec{x}) &= \forall \vec{z} (P(\vec{z}) \rightarrow (\psi_k(\vec{z}, \vec{x}) \vee \vec{z} = \vec{x})) \\ NEXT_P(\vec{x}, \vec{y}) &= P(\vec{x}) \wedge P(\vec{y}) \wedge \psi_k(\vec{x}, \vec{y}) \wedge \\ &\quad \forall \vec{z} (P(\vec{z}) \rightarrow (\psi_k(\vec{z}, \vec{x}) \vee \vec{z} = \vec{x} \vee \psi_k(\vec{y}, \vec{z}) \vee \vec{y} = \vec{z})) \end{aligned}$$

and the k -dimensional lift of the linear order $>$ is inductively defined as

$$\begin{aligned} \psi_1(x, y) &= x < y \\ \psi_i(x_1, \dots, x_i, y_1, \dots, y_i) &= x_i < y_i \vee (x_i = y_i \wedge \psi_{i-1}(x_1, \dots, x_{i-1}, y_1, \dots, y_{i-1})). \end{aligned}$$

In the previous formulae, if \vec{x} and \vec{y} are strings of k variables, then “ $\vec{x} = \vec{z}$ ” abbreviates the conjunction $x_1 = y_1 \wedge \dots \wedge x_k = y_k$.

Observe that Υ_k is a formula in $\Sigma_{1,k}^1 \heartsuit(\sigma)$, hence the result follows. \square

4.7.2 On linear ordered structures $\mathbf{D} = \Sigma_1^1$

In this section, I show that on linear ordered structures $\mathbf{D} = \Sigma_1^1$. It will be shown in Section 4.8 that on arbitrary structures $\mathbf{D} < \Sigma_1^1$.

4.7.2. THEOREM. *On linear ordered structures, $\mathbf{D} = \Sigma_1^1$.*

Proof. In virtue of the result from (Krynicky 1993), cited in this thesis in Section 4.4.3, it suffices to show that for every \mathbf{V} -sentence Φ of the form

$$\left(\begin{array}{cccc} \forall x_1 & \dots & \forall x_k & \exists z \\ \forall y_1 & \dots & \forall y_k & \forall i \in \{0, 1\} \end{array} \right) \gamma(i)(\vec{x}, \vec{y}, z)$$

there is an equivalent \mathbf{D} -sentence Γ . Observe that Φ is equivalent to

$$\exists f \exists X \forall \vec{x} \forall \vec{y} (X(\vec{y}) \rightarrow \gamma(1)(\vec{x}, \vec{y}, f(\vec{x})) \wedge \neg X(\vec{y}) \rightarrow \gamma(0)(\vec{x}, \vec{y}, f(\vec{x}))),$$

where X is a k -ary relation variable and f is a k -ary function variable. In the remainder of the proof I show that one can mimic the function variable f by means of a $2(k+1)$ -ary relation variable Z . More precisely, I will provide a $\Sigma_1^1 \heartsuit$ -sentence Ψ with second-order quantifiers $\exists Z \exists X$ that is equivalent to Φ . The sentence Ψ will be using the k -dimensional lift ψ_k of the linear order symbol $>$, from the proof of Theorem 4.7.1. The $2k$ -ary predicate SUC is defined using ψ_k ; its interpretation on \mathfrak{A} contains all $2k$ -tuples $\langle \vec{a}, \vec{b} \rangle$ such that \vec{b} is the immediate $\psi_k^{\mathfrak{A}}$ -successor of \vec{a} on \mathfrak{A} .

Intuitively, in Ψ the relation variable Z will be defined such that on an arbitrary linear ordered structure \mathfrak{A} , it is the case that

- (1) $Z^{\mathfrak{A}}$ is a linear order of $(k+1)$ -tuples of the universe of \mathfrak{A} ; and
- (2) for all $\vec{a}, \vec{b} \in A^k$, if $\langle \vec{a}, \vec{b} \rangle \in \psi_k^{\mathfrak{A}}$, then for all $a', b' \in A$, $\langle \vec{a}, a', \vec{b}, b' \rangle \in Z^{\mathfrak{A}}$.

Thus, with every k -tuple \vec{a} we associate an \vec{a} -interval in A^{k+1} , to the effect that for two k -tuples \vec{a} and \vec{b} , if $\langle \vec{a}, \vec{b} \rangle \in \psi_k^{\mathfrak{A}}$ then every object in the \vec{a} -interval $Z^{\mathfrak{A}}$ -precedes every tuple in the \vec{b} -interval.

Let $\vec{a} \in A^k$ and let $a' \in A$. Then, if for all $a'' \in A$ it is the case that $\langle \vec{a}, a', \vec{a}, a'' \rangle \in Z^{\mathfrak{A}}$, then a' is called the $Z^{\mathfrak{A}}$ -minimal object of \vec{a} . In the same vein, call a' the $Z^{\mathfrak{A}}$ -maximal object of \vec{a} , if for all $a'' \in A$ it is the case that $\langle \vec{a}, a'', \vec{a}, a' \rangle \in Z^{\mathfrak{A}}$.

Although $Z^{\mathfrak{A}}$ is a relation, it will be used to the effect of a k -ary function f_Z by letting $f_Z(\vec{a})$ be the $Z^{\mathfrak{A}}$ -minimal object of \vec{a} . But—for reasons that will become clear in due course—if \vec{a} is the $\psi_k^{\mathfrak{A}}$ -minimal tuple, then $f_Z(\vec{a})$ is the $Z^{\mathfrak{A}}$ -maximal object of \vec{a} .

For instance, consider the following ordering Z of $\{1, 2, 3\}^2$, observing the 1-, 2-, and 3-interval:

$$\underbrace{\langle 1, 2 \rangle Z \langle 1, 3 \rangle Z \langle 1, 1 \rangle}_{1\text{-interval}} Z \underbrace{\langle 2, 2 \rangle Z \langle 2, 1 \rangle Z \langle 2, 3 \rangle}_{2\text{-interval}} Z \underbrace{\langle 3, 1 \rangle Z \langle 3, 3 \rangle Z \langle 3, 1 \rangle}_{3\text{-interval}}.$$

Then, Z gives rise to the function f_Z , such that

$$\begin{aligned} f_Z(1) &= 1 \\ f_Z(2) &= 2 \\ f_Z(3) &= 1. \end{aligned}$$

In the implementation of Z the $Z^{\mathfrak{A}}$ -minimal object of \vec{a} will be recognized as the object a' such that there exists a tuple \vec{b} and an object b' where $\langle \vec{b}, \vec{a} \rangle \in SUC^{\mathfrak{A}}$ and $\langle \vec{b}, b', \vec{a}, a' \rangle \in Z^{\mathfrak{A}}$. If \vec{a} is the $\psi_k^{\mathfrak{A}}$ -minimal tuple then it cannot be recognized in this manner, since there is no \vec{b} such that $\langle \vec{b}, \vec{a} \rangle \in SUC^{\mathfrak{A}}$. It is for this reason that if \vec{a} is the $\psi_k^{\mathfrak{A}}$ -minimal tuple then $f_Z(\vec{a})$ is the $Z^{\mathfrak{A}}$ -maximal object of \vec{a} . The $Z^{\mathfrak{A}}$ -maximal object of \vec{a} is recognized as the object a' such that there exists a \vec{b} and a b' such that $\langle \vec{a}, \vec{b} \rangle \in SUC^{\mathfrak{A}}$ and $\langle \vec{a}, a', \vec{b}, b' \rangle \in Z^{\mathfrak{A}}$.

Let Ψ be the following sentence:

$$\begin{aligned} \exists Z \exists X \forall \vec{x} \forall \vec{y} \forall \vec{z} \forall u \forall u' \forall u'' \quad & \text{“}Z \text{ is a linear order of } (k+1)\text{-tuples”} \wedge \\ & \psi_k(\vec{x}, \vec{y}) \rightarrow Z(\vec{x}, u, \vec{y}, u') \wedge \\ & (X(\vec{y}) \rightarrow \delta(1)) \wedge (\neg X(\vec{y}) \rightarrow \delta(0)), \end{aligned}$$

where “ Z is a linear order of $(k+1)$ -tuples” abbreviates the conjunction of

$$\begin{aligned} & \neg Z(\vec{x}, u, \vec{x}, u) \\ & Z(\vec{x}, u, \vec{y}, u') \vee (\vec{x} = \vec{y} \wedge u = u') \vee Z(\vec{y}, u', \vec{x}, u) \\ & Z(\vec{x}, u, \vec{y}, u') \wedge Z(\vec{y}, u', \vec{z}, u'') \rightarrow Z(\vec{x}, u, \vec{z}, u'') \end{aligned}$$

and $\delta(i)$, for $i \in \{0, 1\}$, abbreviates the conjunction of

$$\begin{aligned} \neg MIN(\vec{y}) \wedge SUC(\vec{z}, \vec{y}) \wedge Z(\vec{z}, u'', \vec{y}, u') &\rightarrow \gamma(i)(\vec{x}, \vec{y}, u') \\ MIN(\vec{y}) \wedge SUC(\vec{y}, \vec{z}) \wedge Z(\vec{y}, u', \vec{z}, u'') &\rightarrow \gamma(i)(\vec{x}, \vec{y}, u'). \end{aligned}$$

In the δ -formulae, MIN is the predicate that holds only of the ψ_k -minimal tuple.

In view of the discussion of the underlying intuition, Ψ is reasonably self-explanatory. I leave it to the reader to check that Ψ is indeed equivalent to Φ .

To prove that there is a \mathbf{D} -sentence that is equivalent to Φ , it suffices to show that Ψ is a $\Sigma_1^1\heartsuit$ -formula, in virtue of Theorem 4.6.7. To this end observe that one can define ψ_k , SUC , and MIN using only the binary relation symbol $>$. So in particular it follows that these predicates can be defined without the help of relation variables. Finally, observe that every argument of the relation variable Z and X is quantified by the universal quantifiers $\forall \vec{x} \forall \vec{y} \forall \vec{z} \forall u \forall u' \forall u''$. \square

Equivalently, the theorem holds that on linear ordered structures \mathbf{D} captures **NP**.

4.8 Ehrenfeucht-Fraïssé game for \mathbf{D}

I recall the standard model comparison games, or *Ehrenfeucht-Fraïssé* games, for first-order logic and monadic, second-order, existential logic in Sections 4.8.1 and 4.8.2. These games will not be used themselves and serve mainly to appreciate the difference between Σ_1^1 and \mathbf{D} , from a game-theoretic perspective. The reader familiar with the model comparison games for first-order logic and second-order logic may safely skip these parts. This section's contribution to the chapter is a model comparison game for \mathbf{D} . Model comparison games are usually employed to prove that some property is not expressible in a certain logic. I shall use the game for \mathbf{D} to this end in Section 4.9.

The *quantifier rank* of a \mathbf{D} -formula is defined as the maximum number of nested quantifiers in its implicit τ -formulae γ as follows:

$$\begin{aligned} qr(R(\vec{x})) &= 0, \quad \text{for } R \in \tau \\ qr(\neg\phi) &= qr(\phi) \\ qr(\phi \vee \psi) &= \max\{qr(\phi), qr(\psi)\} \\ qr(\exists x \phi) &= qr(\phi) + 1 \\ qr(\gamma) &= \max\{qr(\gamma(\vec{i})) \mid \vec{i} \in \{0, 1\}^k\}, \quad \text{for } \gamma \text{ of type } \{0, 1\}^k \rightarrow \mathbf{FO} \\ qr(\mathbf{D}_k^n \phi) &= qr(\phi). \end{aligned}$$

For any two τ -structures $\mathfrak{A}, \mathfrak{B}$ and r -tuples of objects $\vec{x}^{\mathfrak{A}} \in A^r, \vec{x}^{\mathfrak{B}} \in B^r$, I write

$$\langle \mathfrak{A}, \vec{x}^{\mathfrak{A}} \rangle \equiv_m^{\text{FO}} \langle \mathfrak{B}, \vec{x}^{\mathfrak{B}} \rangle$$

to indicate that for every first-order τ -formula ϕ whose free variables are amongst \vec{x} , if $qr(\phi) \leq m$ then $\mathfrak{A} \models \phi[\vec{x}^{\mathfrak{A}}]$ iff $\mathfrak{B} \models \phi[\vec{x}^{\mathfrak{B}}]$. Likewise, for any language $\mathbf{L}(\tau) \in \{\mathcal{L}(\mathbf{D}_k^n(\tau)), \mathbf{D}(\tau)\}$, I write

$$\mathfrak{A} \equiv_m^{\mathbf{L}} \mathfrak{B}$$

to indicate that every $\mathbf{L}(\tau)$ -sentence Γ , such that $qr(\Gamma) \leq m$, is true on \mathfrak{A} iff it is true on \mathfrak{B} .

A partial function p from A to B is a *partial isomorphism* between \mathfrak{A} and \mathfrak{B} , if it meets the following conditions:

- p is injective; and
- for every k -ary relation symbol $R \in \tau$ and all $a_1, \dots, a_k \in \text{dom}(p)$ it is the case that $\langle a_1, \dots, a_k \rangle \in R^{\mathfrak{A}}$ iff $\langle p(a_1), \dots, p(a_k) \rangle \in R^{\mathfrak{B}}$.

Note that vocabularies do not contain constants, for which reason no condition for them is included. Sometimes it will be handy to treat p as the set $\bigcup_{a \in \text{dom}(p)} \{ \langle a, p(a) \rangle \}$.

4.8.1 Comparison game for FO

Let τ be a vocabulary and let m be an integer. Let $\mathfrak{A}, \mathfrak{B}$ be τ -structures and let $\vec{x}^{\mathfrak{A}} = \langle x_1^{\mathfrak{A}}, \dots, x_r^{\mathfrak{A}} \rangle \in A^r$, and $\vec{x}^{\mathfrak{B}} = \langle x_1^{\mathfrak{B}}, \dots, x_r^{\mathfrak{B}} \rangle \in B^r$. Then, the *m -round Ehrenfeucht-Fraïssé game* on \mathfrak{A} and \mathfrak{B} , denoted by

$$EF_m^{\text{FO}}(\langle \mathfrak{A}, \vec{x}^{\mathfrak{A}} \rangle, \langle \mathfrak{B}, \vec{x}^{\mathfrak{B}} \rangle),$$

is an m -round game proceeding as follows: There are two players, Spoiler (male) and Duplicator (female). On the i th round, where $1 \leq i \leq m$, Spoiler first chooses a structure \mathfrak{A} (or \mathfrak{B}) and an element called c_i (or d_i) from the universe of the chosen structure. Duplicator replies by choosing an element d_i (or c_i) from the universe of the other structure \mathfrak{B} (or \mathfrak{A}). Duplicator wins the play $\langle c_1, d_1 \rangle, \dots, \langle c_m, d_m \rangle$, if the relation

$$\{ \langle x_i^{\mathfrak{A}}, x_i^{\mathfrak{B}} \rangle \mid 1 \leq i \leq r \} \cup \{ \langle c_j, d_j \rangle \mid 1 \leq j \leq m \}$$

is a partial isomorphism between \mathfrak{A} and \mathfrak{B} ; otherwise, Spoiler wins the play. If against any sequence of moves by Spoiler, Duplicator is able to make her moves so as to win the resulting play, say that Duplicator has a *winning strategy* in $EF_m^{\text{FO}}(\langle \mathfrak{A}, \vec{x}^{\mathfrak{A}} \rangle, \langle \mathfrak{B}, \vec{x}^{\mathfrak{B}} \rangle)$. The notion of winning strategy for Spoiler is defined

analogously. By the Gale-Stewart Theorem, cited as Theorem 2.2.1 in this dissertation, Ehrenfeucht-Fraïssé games are determined, that is, precisely one of the players has a winning strategy. The effectiveness of these games is established in the following seminal result.

4.8.1. THEOREM (FRAÏSSÉ (1954) AND EHRENFUCHT (1961)). *The following are equivalent for every integer m :*

- *Duplicator has a winning strategy in $EF_m^{\mathbf{FO}}(\langle \mathfrak{A}, \vec{x}^{\mathfrak{A}} \rangle, \langle \mathfrak{B}, \vec{x}^{\mathfrak{B}} \rangle)$.*
- $\langle \mathfrak{A}, \vec{x}^{\mathfrak{A}} \rangle \equiv_m^{\mathbf{FO}} \langle \mathfrak{B}, \vec{x}^{\mathfrak{B}} \rangle$.

4.8.2 Comparison game for $\mathbf{M}\Sigma_1^1$

In the Ehrenfeucht-Fraïssé game for first-order logic, Spoiler is free to choose the structure from which he picks his object. This freedom reflects the fact that first-order logic is closed under complementation. That is, if a property Π is expressible in first-order logic (say by ϕ) then also its complement can be expressed in first-order logic (namely by $\neg\phi$).

Yet sometimes one wishes to show that a language is not closed under complementation. The grand question as to whether \mathbf{NP} equals \mathbf{coNP} is a case in point: $\mathbf{NP} = \mathbf{coNP}$ iff Σ_1^1 is closed under complementation. Although the latter question is still open, the problem has been solved for the monadic fragment of Σ_1^1 : $\mathbf{M}\Sigma_1^1$ is *not* closed under complementation. This was proved by Fagin (1974), who showed that $\mathbf{CONNECTED}$ is not expressible in $\mathbf{M}\Sigma_1^1$, whereas its complement $\overline{\mathbf{CONNECTED}}$ is expressible in $\mathbf{M}\Sigma_1^1$. $\mathbf{CONNECTED}$ is the graph property such that $\mathbf{CONNECTED}(\mathfrak{G}) = \mathit{true}$ iff for every pair of vertices v, v' in the graph \mathfrak{G} , there is a path from v to v' . Fagin's proof uses model comparison games.

Let τ be a vocabulary, let $\mathfrak{A}, \mathfrak{B}$ be τ -structures, and let m, n be integers. Then, the m -round, n -color $\mathbf{M}\Sigma_1^1$ -Ehrenfeucht-Fraïssé game on \mathfrak{A} and \mathfrak{B} , denoted as

$$EF_{n,m}^{\mathbf{M}\Sigma_1^1}(\mathfrak{A}, \mathfrak{B})$$

has two *phases* and proceeds as follows. First there is the *coloring phase*. The game commences by Spoiler choosing n monadic relations $X_1^{\mathfrak{A}}, \dots, X_n^{\mathfrak{A}}$ on A , whereupon Duplicator chooses n monadic relations $X_1^{\mathfrak{B}}, \dots, X_n^{\mathfrak{B}}$ on B . Crucially, Spoiler picks up sets from the universe A ; he does not have the liberty to pick the structure of his liking. Monadic relations are just predicates or *colorings* of the universe, hence the name. Next, the *first-order phase* starts, during which the players play the m -round Ehrenfeucht-Fraïssé on the expanded structures, that is, they play

$$EF_m^{\mathbf{FO}}(\langle \mathfrak{A}, X_1^{\mathfrak{A}}, \dots, X_n^{\mathfrak{A}} \rangle, \langle \mathfrak{B}, X_1^{\mathfrak{B}}, \dots, X_n^{\mathfrak{B}} \rangle).$$

The winner of this game is the winner of $EF_{n,m}^{\mathbf{M}\Sigma_1^1}(\mathfrak{A}, \mathfrak{B})$. The effectiveness of these games is established as follows: Let $\Phi = \exists X_1 \dots \exists X_n \Psi$ be an $\mathbf{M}\Sigma_1^1(\tau)$ -sentence

where Ψ contains no second-order quantifiers and $qr(\Psi) \leq m$. Then, for any two τ -structures \mathfrak{A} and \mathfrak{B} the following are equivalent:

- Duplicator has a winning strategy in $EF_{n,m}^{\mathbf{M}\Sigma_1^1}(\mathfrak{A}, \mathfrak{B})$.
- $\mathfrak{A} \models \Phi$ implies $\mathfrak{B} \models \Phi$.

Note the asymmetry in the definition of the model comparison game—Spoiler is forced to start picking from A —and the one-directional implication in the second statement.

Model comparison games for $\Sigma_{1,k}^1$ can be defined analogous to the one described for $\mathbf{M}\Sigma_1^1$, by letting the players pick up sets from A^k and B^k rather than A and B . At present, these games have not lead to separation results for $\Sigma_{1,k}^1$ and $\Pi_{1,k}^1$, for any $k \geq 2$.

4.8.3 Comparison games for \mathbf{D}

In essence, the model comparison game for \mathbf{D} explicated below is a modification of the comparison game introduced in (Sandu and Väänänen 1992).

Like the model comparison game for Σ_1^1 , the game for \mathbf{D} has two phases: a *watercoloring phase* and a *first-order phase*.

Let τ be a vocabulary, let \mathfrak{A} and \mathfrak{B} be τ -structures and let m be an integer. Then, the *m -round, watercolor \mathbf{D}_k^n -Ehrenfeucht-Fraïssé game* on \mathfrak{A} and \mathfrak{B} , denoted as

$$EF_m^{\mathbf{D}_k^n}(\mathfrak{A}, \mathfrak{B})$$

is an $m+1$ -round game proceeding as follows: During the watercoloring phase, Spoiler picks for every $1 \leq i \leq n$ a subset X_i from A^k . Duplicator picks a subset B_i of B^k , for every $1 \leq i \leq n$. Next, Spoiler chooses a tuple $\vec{x}_i^{\mathfrak{B}} \in B^k$, for every $1 \leq i \leq n$, and Duplicator replies by choosing a tuple $\vec{x}_i^{\mathfrak{A}} \in A^k$. If for every $1 \leq i \leq n$ the selected tuples satisfy $\vec{x}_i^{\mathfrak{A}} \in A_i$ iff $\vec{x}_i^{\mathfrak{B}} \in B_i$, then the game proceeds to the first-order phase as $EF_m^{\mathbf{FO}}(\langle \mathfrak{A}, \vec{x}^{\mathfrak{A}} \rangle, \langle \mathfrak{B}, \vec{x}^{\mathfrak{B}} \rangle)$; otherwise, Duplicator loses right away. The winner of the first-order game is the winner of $EF_m^{\mathbf{D}_k^n}(\mathfrak{A}, \mathfrak{B})$.

Important to note that in the first-order Ehrenfeucht-Fraïssé game that is started up after the watercolor phase, the actual colorings are immaterial. The watercolors fade away quickly, so to say.

4.8.2. PROPOSITION. *Let τ be a vocabulary, let \mathfrak{A} and \mathfrak{B} be τ -structures, and let k, m, n be integers. Let $\Gamma = \mathbf{D}_k^n \gamma$ be any $\mathbf{D}_k^n(\tau)$ -sentence with $qr(\gamma) \leq m$. Then, (1) and (2) hold:*

- (1) *Statement (a) implies (b):*

(a) Duplicator has a winning strategy in $EF_m^{\mathbf{D}^n}(\mathfrak{A}, \mathfrak{B})$.

(b) $\mathfrak{A} \models \Gamma$ implies $\mathfrak{B} \models \Gamma$.

(2) If (a) holds for arbitrary k, n , then (b) holds for every \mathbf{D} -sentence Γ , with $qr(\Gamma) \leq m$.

Proof. Ad (1): I shall prove the case in which $k = 1$, as this will enhance readability of the argument and it facilitates a gentler comparison with the game for $\mathbf{M}\Sigma_1^1$ from Section 4.8.2. This limitation does not affect the general result, though.

So let $\Gamma = \mathbf{D}_1^n \gamma$ be a $\mathbf{D}_1^n(\tau)$ -sentence with $qr(\gamma) \leq m$. Assume \mathfrak{A} is a τ -structure such that $\mathfrak{A} \models \Gamma$ and such that Duplicator has a winning strategy in $EF_m^{\mathbf{D}_1^n}(\mathfrak{A}, \mathfrak{B})$. That is, Duplicator has a strategy that guarantees a win no matter Spoiler's behavior during the game.

The fact that $\mathfrak{B} \models \Gamma$ will be derived by considering the case in which Spoiler kicks off by picking the sets $X_1^{\mathfrak{A}}, \dots, X_n^{\mathfrak{A}}$ on $A^k = A$ in such a way that they are witnesses of the fact that $\mathfrak{A} \models \Gamma$. By “ $X_1^{\mathfrak{A}}, \dots, X_n^{\mathfrak{A}}$ being witnesses of $\mathfrak{A} \models \Gamma$,” I mean that they witness that $\mathfrak{A} \models T(\Gamma)$, that is,

$$\mathfrak{A} \models \forall x_1 \dots \forall x_n T_{L(\mathbf{D}_1^n)}(\gamma)[X_1^{\mathfrak{A}}, \dots, X_n^{\mathfrak{A}}],$$

where $T_{L(\mathbf{D}_1^n)}(\gamma)$ denotes the $L(\mathbf{D}_1^n)$ -explication of γ , see Definition 4.5.2 on page 87.

Let Duplicator respond by picking the sets $X_1^{\mathfrak{B}}, \dots, X_n^{\mathfrak{B}}$ on B that are prescribed by her winning strategy. Thereupon, suppose Spoiler picks arbitrary objects $x_1^{\mathfrak{B}}, \dots, x_n^{\mathfrak{B}}$ from B and let Duplicator choose the objects $x_1^{\mathfrak{A}}, \dots, x_n^{\mathfrak{A}}$ from A that are prescribed by her winning strategy. By making his choice, Spoiler implicitly selects one of the first-order τ -formulae in the matrix formula γ . In particular, Spoiler selects $\gamma^* = \gamma(t(x_1^{\mathfrak{B}}, X_1^{\mathfrak{B}}), \dots, t(x_n^{\mathfrak{B}}, X_n^{\mathfrak{B}}))$, where

$$t(a, A) = \begin{cases} 1 & \text{if } a \in A \\ 0 & \text{if } a \notin A. \end{cases}$$

Since Duplicator selected the objects $x_1^{\mathfrak{A}}, \dots, x_n^{\mathfrak{A}}$ using her winning strategy, it must be the case that $x_i^{\mathfrak{A}} \in X_i^{\mathfrak{A}}$ iff $x_i^{\mathfrak{B}} \in X_i^{\mathfrak{B}}$, for every $1 \leq i \leq n$. Hence, $\gamma(t(x_1^{\mathfrak{B}}, X_1^{\mathfrak{B}}), \dots, t(x_n^{\mathfrak{B}}, X_n^{\mathfrak{B}})) = \gamma^* = \gamma(t(x_1^{\mathfrak{A}}, X_1^{\mathfrak{A}}), \dots, t(x_n^{\mathfrak{A}}, X_n^{\mathfrak{A}}))$.

By the assumption that $\mathfrak{A} \models \Gamma$ and that $X_1^{\mathfrak{A}}, \dots, X_n^{\mathfrak{A}}$ are witnesses thereof it follows that $\mathfrak{A} \models \gamma^*[\vec{x}^{\mathfrak{A}}]$.

Because Duplicator has a winning strategy and she made all her previous moves in accordance with one winning strategy, she can keep playing this winning strategy in the first-order phase of the game— $EF_m^{\mathbf{FO}}(\langle \mathfrak{A}, \vec{x}^{\mathfrak{A}} \rangle, \langle \mathfrak{B}, \vec{x}^{\mathfrak{B}} \rangle)$ —and win it. A direct application of the effectiveness of Ehrenfeucht-Fraïssé games, Theorem 4.8.1 in this chapter, yields that $\langle \mathfrak{A}, \vec{x}^{\mathfrak{A}} \rangle \equiv_m^{\mathbf{FO}} \langle \mathfrak{B}, \vec{x}^{\mathfrak{B}} \rangle$. By assumption, $qr(\gamma) \leq m$, so in particular, $qr(\gamma^*) \leq m$. It follows directly that $\mathfrak{B} \models \gamma^*[\vec{x}^{\mathfrak{B}}]$.

Since $\vec{x}^{\mathfrak{B}}$ was chosen arbitrarily, so was γ^* . Thus we get that

$$\mathfrak{B} \models \forall x_1 \dots \forall x_n \bigwedge_{\vec{i} \in \{0,1\}^n} ((\pm_{i_1} X_1(x_1) \wedge \dots \wedge \pm_{i_n} X_n(x_n)) \rightarrow \gamma(\vec{i}))[X_1^{\mathfrak{B}}, \dots, X_n^{\mathfrak{B}}].$$

The latter is the case iff

$$\mathfrak{B} \models \forall x_1 \dots \forall x_n T_{L(\mathbf{D}_1^n)}(\gamma)[X_1^{\mathfrak{B}}, \dots, X_n^{\mathfrak{B}}].$$

Introduction of existential quantifiers yields

$$\mathfrak{B} \models \exists X_1 \dots \exists X_n \forall x_1 \dots \forall x_n T_{L(\mathbf{D}_1^n)}(\gamma).$$

The latter formula is simply the standard translation $T(\Gamma)$ of Γ , see Definition 4.5.2. Hence, by adequacy of T proved in Proposition 4.5.3 it is the case that $\mathfrak{B} \models \Gamma$.

Ad (2): Suppose that Duplicator has a winning strategy in $EF_m^{\mathbf{D}_k^n}(\mathfrak{A}, \mathfrak{B})$, for every k, n , and that $\mathfrak{A} \models \Gamma$, for an arbitrary $\mathbf{D}(\tau)$ -sentence Γ . By definition, $\mathbf{D}(\tau) = \bigcup_{k,n} \mathbf{D}_k^n(\tau)$, so there are k, n , such that Γ is in fact a $\mathbf{D}_k^n(\tau)$ -sentence. Furthermore, there is an m such that $qr(\Gamma) = m$. From the assumption and the previous implication it follows that $\mathfrak{B} \models \Gamma$, as required. \square

To appreciate the difference between model comparison games for Σ_1^1 and \mathbf{D} , recall that $\mathbf{M}\Sigma_1^1$ -Ehrenfeucht-Fraïssé games have two phases. During the second-order phase only relations over the universe are selected. Thereafter, the first-order phase begins on the extended models. By contrast, in the watercolor phase in \mathbf{D} -Ehrenfeucht-Fraïssé games first relations are selected and then objects. If the chosen objects satisfy the imposed constraint, then the first-order phase is started up but only the objects that were chosen in the watercolor phase persist. The relations are immaterial during the first-order phase.

4.9 Non-expressibility result for \mathbf{D}

Model comparison games are typically used to prove that some property Π is not expressible in a certain language. In this manner, I will show that \mathbf{D} is not closed under complementation and that $\mathbf{D} < \Sigma_1^1$.

Using the model comparison games for \mathbf{D} developed in Subsection 4.8.3, I show that \mathbf{D} is not closed under complementation. That is, there exists a class of finite graphs that is characterizable in \mathbf{D} but the complement of this class is not. This result may be interesting because it concerns a fragment of Σ_1^1 that is not bounded by the arity of the relation variables and has a non-empty intersection with k -ary, existential, second-order logic, for arbitrary k , see Theorem 4.8.3.

Let τ be a vocabulary. For any two τ -structures \mathfrak{A} and \mathfrak{B} with non-intersecting universes, let $\mathfrak{A} \cup \mathfrak{B}$ denote the τ -structure with universe $A \cup B$ and $R^{\mathfrak{A} \cup \mathfrak{B}} = R^{\mathfrak{A}} \cup R^{\mathfrak{B}}$, for every $R \in \tau$.

4.9.1. THEOREM. $\overline{2\text{-COLORABILITY}}$ is not expressible in \mathbf{D} .

Proof. For the sake of contradiction, suppose $\overline{2\text{-COLORABILITY}}$ were expressible in \mathbf{D} . Then, there would be a particular \mathbf{D} -sentence that characterizes $\overline{2\text{-COLORABILITY}}$, call it Γ . Γ would have a partially ordered connective with dimensions k, n prefixing a implicit matrix τ -formula of quantifier rank m . Now consider two graphs \mathfrak{A} and \mathfrak{B} such that (i) \mathfrak{A} is not 2-colorable and \mathfrak{B} is 2-colorable and (ii) Duplicator has a winning strategy in $EF_m^{\mathbf{D}^n}(\mathfrak{A}, \mathfrak{B})$. Since Γ characterizes $\overline{2\text{-COLORABILITY}}$, I derive from (i) that $\mathfrak{A} \models \Gamma$ and $\mathfrak{B} \not\models \Gamma$. But from (ii) and $\mathfrak{A} \models \Gamma$ it follows by Proposition 4.8.2, that $\mathfrak{B} \models \Gamma$. Contradiction. Hence, no \mathbf{D} -sentence expresses $\overline{2\text{-COLORABILITY}}$.

It remains to be shown that for arbitrary k, m, n , there exist graphs \mathfrak{A} and \mathfrak{B} meeting (i) and (ii). To this end, fix integers k, m, n and consider the graphs \mathfrak{C} and \mathfrak{D} , where

$$\begin{aligned} C &= \{c_1, \dots, c_N\} \\ R^{\mathfrak{C}} &= \{\langle c_i, c_{i+1} \rangle, \langle c_{i+1}, c_i \rangle \mid 1 \leq i \leq N-1\} \cup \{\langle c_N, c_1 \rangle, \langle c_1, c_N \rangle\} \\ D &= \{d_1, \dots, d_{N+1}\} \\ R^{\mathfrak{D}} &= \{\langle d_i, d_{i+1} \rangle, \langle d_{i+1}, d_i \rangle \mid 1 \leq i \leq N\} \cup \{\langle d_{N+1}, d_1 \rangle, \langle d_1, d_{N+1} \rangle\} \end{aligned}$$

and $N = 2^{m+(k \cdot n)}$. So \mathfrak{C} and \mathfrak{D} are cycles of even and odd length, respectively. A cycle is 2-colorable iff it is of even length; whence \mathfrak{D} is not 2-colorable whereas \mathfrak{C} is. Obviously, the structure $\mathfrak{C} \cup \mathfrak{D}$ is not 2-colorable either.

I show that Duplicator has a winning strategy in $EF_m^{\mathbf{D}^n}(\mathfrak{C} \cup \mathfrak{D}, \mathfrak{C})$. Suppose Spoiler selects the set $X_i \subseteq (C \cup D)^k$, for every $1 \leq i \leq n$. Let Duplicator respond with X_i restricted to the universe of \mathfrak{C} , that is, $Y_i = X_i \cap C^k$, for every $1 \leq i \leq n$. Suppose Spoiler selects the tuple $\vec{x}_i^{\mathfrak{C}} \in C^k$, for every $1 \leq i \leq n$. Let Duplicator respond by simply copying these tuples on $(C \cup D)^k$, that is, choosing $\vec{x}_i^{\mathfrak{C} \cup \mathfrak{D}} = \vec{x}_i^{\mathfrak{C}}$. The game advances to the first-order phase, since $\vec{x}_i^{\mathfrak{C} \cup \mathfrak{D}} \in X_i$ iff $\vec{x}_i^{\mathfrak{C}} \in Y_i$. A standard argument suffices to see that Duplicator has a winning strategy in

$$EF_m^{\mathbf{FO}}(\langle \mathfrak{C} \cup \mathfrak{D}, \vec{x}_1^{\mathfrak{C} \cup \mathfrak{D}}, \dots, \vec{x}_n^{\mathfrak{C} \cup \mathfrak{D}} \rangle, \langle \mathfrak{C}, \vec{x}_1^{\mathfrak{C}}, \dots, \vec{x}_n^{\mathfrak{C}} \rangle),$$

compare (Ebbinghaus and Flum 1999, pg. 23). □

4.9.2. COROLLARY. In Example 4.6.6 on page 92 I concluded that the graph property 2-COLORABILITY is expressible in \mathbf{D} . Theorem 4.9.1 shows that the complement of this class is not expressible in \mathbf{D} . Therefore, on finite graphs, $\mathbf{D} \neq \neg\mathbf{D}$, that is, \mathbf{D} is not closed under complementation.

4.9.3. COROLLARY. *Since $\mathfrak{C} \cup \mathfrak{D}$, from the proof of Theorem 4.9.1, is not connected but \mathfrak{C} is, $\overline{\text{CONNECTED}}$ is not expressible in \mathbf{D} . However, $\overline{\text{CONNECTED}}$ is expressible in $\mathbf{M}\Sigma_1^1$, cf. (Fagin 1975). So $\mathbf{D} < \Sigma_1^1$.*

Actually, it should not come as a big surprise, that on finite graphs $\mathbf{D} < \Sigma_1^1$. For suppose the converse were true, that is, suppose that $\mathbf{D} = \Sigma_1^1$. Then, by definition it would be the case that $\neg\mathbf{D} = \Pi_1^1$. Corollary 4.9.2, however, holds that $\mathbf{D} \neq \neg\mathbf{D}$ and would thus imply that $\Sigma_1^1 \neq \Pi_1^1$, or equivalently, that $\mathbf{NP} \neq \mathbf{coNP}$.

4.10 Descriptive complexity of $\mathcal{L}(\mathbf{D})$ and $\mathcal{L}(\mathbf{H})$

In this section I will take up the descriptive complexity of the logics $\mathcal{L}(\mathbf{D})$ and $\mathcal{L}(\mathbf{H})$. This will give us an algorithmic view on Henkin quantifiers. Furthermore it teaches us the way partially ordered quantifiers manifest themselves in the theory of computing. A more general variant of Theorem 4.10.4 from this section appeared in an excellent paper by Gottlob (1997), pointed out to me by Mostowski. By the time I investigated these issues I was unaware of this publication, and obtained a proof different from Gottlob's. That is, the references I use do not build on any of Gottlob's results nor on his main references. An independent proof that is.

In Section 4.10.1, I will study the descriptive complexity of $\mathcal{L}(\mathbf{D})$ and $\mathcal{L}(\mathbf{H})$ and show that they capture the complexity class $\mathbf{P}_{\parallel}^{\mathbf{NP}}$ on linear ordered structures. These results are put into perspective in Section 4.10.2.

4.10.1 $\mathcal{L}(\mathbf{D})$ and $\mathcal{L}(\mathbf{H})$ capture $\mathbf{P}_{\parallel}^{\mathbf{NP}}$

For future reference I show that $\mathcal{L}(\mathbf{H})$ has a Prenex normal form. If I want to distinguish two different Henkin quantifiers without referring to their dimensions, I may index them with numerals like $\mathbf{H}_{(i)}$.

4.10.1. PROPOSITION. *Let τ be a vocabulary. Then, every $\mathcal{L}(\mathbf{H})(\tau)$ -formula Φ is equivalent to an $\mathcal{L}(\mathbf{H})(\tau)$ -formula of the following form:*

$$\pm_1 \mathbf{H}_{(1)} \vec{x}_1 \dots \pm_n \mathbf{H}_{(n)} \vec{x}_n \phi,$$

where $\mathbf{H}_{(i)}$ is a Henkin prefix of arbitrary dimensions, $\pm_i \in \{\neg, \neg\neg\}$, and ϕ is a $\mathbf{FO}(\tau)$ -formula.

Proof. A standard inductive proof suffices, the only non-trivial case being the conjunction. But also this case is easily dealt with: $\mathbf{H}_{(1)} \vec{x} \phi_1(\vec{x}) \wedge \mathbf{H}_{(2)} \vec{y} \phi_2(\vec{y})$ is equivalent to $\mathbf{H}_{(1)} \vec{x} \mathbf{H}_{(2)} \vec{z} (\phi_1(\vec{x}) \wedge \phi_2(\vec{z}))$, where \vec{z} is a string of variables none of which appear in \vec{x} . \square

The main observation of this section concerns the descriptive complexity of $\mathcal{L}(\mathbf{H})$ and $\mathcal{L}(\mathbf{D})$, that is associated with the complexity class $\mathbf{P}_{\parallel}^{\mathbf{NP}}$.⁵ $\mathbf{P}_{\parallel}^{\mathbf{NP}}$ denotes the class of properties decidable in deterministic polynomial time with the help of an \mathbf{NP} -oracle that can be asked a polynomial number of queries in parallel only once. The action of querying the oracle takes one time step.

4.10.2. THEOREM. *The expression complexity of $\mathcal{L}(\mathbf{D})$ and $\mathcal{L}(\mathbf{H})$ is in $\mathbf{P}_{\parallel}^{\mathbf{NP}}$.*

Proof. It suffices to show that for an arbitrary $\mathcal{L}(\mathbf{H})$ -sentence Φ , deciding whether Φ is true on a finite, suitable structure \mathfrak{A} can be done in $\mathbf{P}_{\parallel}^{\mathbf{NP}}$. First I describe an algorithm that computes whether Φ is true on \mathfrak{A} . Thereafter, I observe that this algorithm can be implemented on a Turing machine that works in $\mathbf{P}_{\parallel}^{\mathbf{NP}}$. This will also settle the argument for $\mathcal{L}(\mathbf{D})$, in virtue of Proposition 4.3.4.

As for the algorithm, due to Proposition 4.10.1 one may assume without loss of generality that Φ has the form:

$$\pm_1 \mathbf{H}_{(1)} \vec{x}_1 \dots \pm_n \mathbf{H}_{(n)} \vec{x}_n \psi(\vec{x}),$$

where ψ is a first-order formula over the variables $\vec{x} = \vec{x}_1, \dots, \vec{x}_n$.

If S is a set and $V = \{\vec{v}\}$ is a set of variables, let S^V denote the set of assignments from V to S . I may write $S^{\vec{v}}$ instead of S^V .

Let the algorithm start off by writing down all variable assignments in $A^{\vec{x}}$, and label every assignment $\alpha \in A^{\vec{x}}$ with *true* if $\mathfrak{A} \models \psi(\vec{x})[\alpha]$, and *false* otherwise. Note that ψ 's truth conditions on \mathfrak{A} are completely spelled out thereafter.

Put $i = n$ and $\Xi_{i+1} = \psi$. For every i from n to 1 , proceed as follows for $\pm_i \mathbf{H}_{(i)} \vec{x}_i$ in Φ :

- Write down all assignments in $A^{\vec{x}_1, \dots, \vec{x}_{i-1}}$.
- For every assignment $\alpha \in A^{\vec{x}_1, \dots, \vec{x}_{i-1}}$ ask the oracle if $\mathfrak{A} \models \mathbf{H}_{(i)} \vec{x}_i \Xi_{i+1}[\alpha]$.
- Label α with *true* if the answer of the oracle was positive and $\pm_i = \neg\neg$ or the answer was negative and $\pm_i = \neg$; otherwise label it *false*.
- Erase all labeled assignments from $A^{\vec{x}_1, \dots, \vec{x}_i}$ and let the current list of assignments fully specify the truth conditions of $\Xi_i(\vec{x}_1, \dots, \vec{x}_{i-1})$; that is, let Ξ_i be the formula that holds of an assignment α on \mathfrak{A} if and only if α is labeled *true*.

⁵Gottlob's (1997) version of the theorem is cast in terms of $\mathbf{L}^{\mathbf{NP}}$, that is, the class of problems decidable in logarithmic space with an \mathbf{NP} -oracle. Recall that $\mathbf{L}^{\mathbf{NP}} = \mathbf{P}_{\parallel}^{\mathbf{NP}}$, due to (Wagner 1990).

Finally, upon arriving at $n = 0$, if the empty assignment is labeled *true* the algorithm accepts the input; otherwise, it rejects it. By means of an elementary inductive argument this algorithm can be shown correct.

Apart from consulting the oracle, this algorithm runs in polynomial deterministic time: enumerating all assignments over n iterations takes at most $n \cdot \|A^{\vec{x}}\|$ steps, which is clearly polynomial in the size of the input, $\|\mathfrak{A}\|$. Since \mathbf{H} captures \mathbf{NP} it is sufficient (and necessary) to employ an \mathbf{NP} -oracle. This renders the algorithm in $\mathbf{P}^{\mathbf{NP}}$, since the number of queries are bounded by the polynomially many different assignments. Yet, this result can be improved, since per iteration the oracle can harmlessly be consulted in parallel. So the algorithm needs a constant number of n parallel queries to the \mathbf{NP} -oracle. In (Buss and Hay 1991) it was shown that a constant number of rounds of polynomially many queries to an \mathbf{NP} -oracle is equivalent to one round of parallel queries. Therefore, the algorithm sits in $\mathbf{P}_{\parallel}^{\mathbf{NP}}$. \square

Let $\mathbf{H}^+(\tau)$ be the *first-order closure* of $\mathbf{H}(\tau)$. That is, the closure of $\mathbf{H}(\tau)$ under boolean operations and existential quantification (but not under application of Henkin quantifiers). More formally, $\mathbf{H}^+(\tau)$ is generated by the following grammar:

$$\Phi ::= \Psi \mid \neg\Phi \mid \Phi \vee \Phi \mid \exists x \Phi,$$

where Ψ ranges over the $\mathbf{H}(\tau)$ -formulae. Let $\mathbf{D}^+(\tau)$ be defined similarly. The first-order closure of (fragments of) Σ_1^1 was taken up in (Ajtai, Fagin, and Stockmeyer 2000). In the latter publication, the authors observe that the first-order closure of Σ_1^1 captures $\mathbf{P}_{\parallel}^{\mathbf{NP}}$, on linear ordered structures. Since $\mathbf{H} = \Sigma_1^1$, the following result follows.

4.10.3. PROPOSITION. *On linear ordered structures, \mathbf{D}^+ and \mathbf{H}^+ capture $\mathbf{P}_{\parallel}^{\mathbf{NP}}$.*

Proof. The case of \mathbf{H}^+ follows from the observation from (Ajtai, Fagin, and Stockmeyer 2000). In Theorem 4.7.2, it was proved that on linear ordered structures, $\mathbf{D} = \Sigma_1^1$. Hence, on linear ordered structures, $\mathbf{D} = \mathbf{H}$. An inductive arguments settles that $\mathbf{D}^+ = \mathbf{H}^+$, on linear ordered structures. \square

In the proof of the following theorem, we will use the fact the logic \mathbf{H}^+ is a fragment of $\mathcal{L}(\mathbf{H})$.

4.10.4. THEOREM. *On linear ordered structures, $\mathcal{L}(\mathbf{D})$ and $\mathcal{L}(\mathbf{H})$ capture $\mathbf{P}_{\parallel}^{\mathbf{NP}}$.*

Proof. By Theorem 4.10.2, $\mathcal{L}(\mathbf{H})$'s expression complexity is in $\mathbf{P}_{\parallel}^{\mathbf{NP}}$ on the class of all finite structure. It remains to be proved therefore that $\mathcal{L}(\mathbf{H})$ captures at least $\mathbf{P}_{\parallel}^{\mathbf{NP}}$, on linear ordered structures. To this end, let Π be an arbitrary $\mathbf{P}_{\parallel}^{\mathbf{NP}}$ -decidable property on the linear ordered structures. In virtue of Proposition

4.10.3, obtain that there is a sentence Φ from \mathbf{H}^+ that expresses Π over the linear ordered structures. As I concluded at the outset of this theorem, Φ is a sentence in $\mathcal{L}(\mathbf{H})$ as well. Whence, Π is expressible in $\mathcal{L}(\mathbf{H})$ as well.

Idem for $\mathcal{L}(\mathbf{D})$. □

4.10.2 Aftermath

I wish to warn the reader who is about to jump into conclusions about parallel computation and partially ordered quantification. Admittedly, the complexity class $\mathbf{P}_{\parallel}^{\mathbf{NP}}$ is based on parallel Turing machines and it is captured by $\mathcal{L}(\mathbf{H})$, on linear ordered structures. However, this does not mean that model checking a single formula $\mathbf{H}\vec{x} \phi \in \mathbf{H}$ can be done by parallel means, as this requires “simply” an \mathbf{NP} -machine. The parallel way of computing comes in effect only when one computes the semantic value of several \mathbf{H} -formulae at the same moment in time. For instance, if $\mathbf{H}_k^n \vec{x} \phi(y)$ is an \mathbf{H} -formula with one free variable y , then model checking all of

$$\mathfrak{A} \models \mathbf{H}_k^n \vec{x} \phi(y)[y/a_1] \quad \dots \quad \mathfrak{A} \models \mathbf{H}_k^n \vec{x} \phi(y)[y/a_m]$$

for objects $a_1, \dots, a_m \in A$, can be done by one round of m parallel queries to an \mathbf{NP} -oracle. It is this principle that underlies the fact that $\mathcal{L}(\mathbf{H})$'s expression complexity is in $\mathbf{P}_{\parallel}^{\mathbf{NP}}$. On the other hand, it is noteworthy that a polynomial number of parallel queries suffice is due to the fact that $\mathcal{L}(\mathbf{H})$ -formulae only contain first-order variables. This, namely, causes it sufficient to spell out all variable assignments, simply being tuples of objects, and to compute the formula's semantic value with respect to this list. By contrast, if one wishes to verify a second-order formula like $\exists X \forall Y \exists Z \phi$ on a structure, spelling out variable assignments amounts to checking triples of *subsets* of tuples of objects. Interesting to note here that full second-order logic captures the Polynomial Hierarchy, whereas the full logic $\mathcal{L}(\mathbf{H})$ gets stuck at $\mathbf{P}_{\parallel}^{\mathbf{NP}}$. In this sense Theorem 4.10.2 provides the computational upper-bound of partially ordered, yet first-order, quantification.

One way to appreciate the fact that the logics \mathbf{H}^+ and $\mathcal{L}(\mathbf{H})$ coincide on linear ordered structures is by means of the *Henkin depth* of $\mathcal{L}(\mathbf{H})$ -formulae:

$$\begin{aligned} hd(\Phi) &= 0, \quad \text{for first-order } \Phi \\ hd(\neg\Phi) &= hd(\Phi) \\ hd(\Phi \vee \Psi) &= \max\{hd(\Phi), hd(\Psi)\} \\ hd(\exists x \Phi) &= hd(\Phi) \\ hd(\mathbf{H}\vec{x} \Phi) &= hd(\Phi) + 1, \end{aligned}$$

reading $\mathbf{H}_n^0 x_1 \dots x_n$ as $\exists x_1 \dots \exists x_n$.

Clearly every \mathbf{H}^+ -sentence has Henkin depth at most one. Therefore, by Theorem 4.10.4 it is the case that for every $\mathcal{L}(\mathbf{H})$ -sentence Φ there exists an \mathbf{H}^+ -sentence Ψ , such that $hd(\Psi) \leq 1$ and that on the class of linear ordered structures Φ and Ψ define the same property. Put differently, on linear ordered structures granting Henkin quantifiers to nest does not yield greater expressive power. Gottlob (1997) proves an even stronger normal form for $\mathcal{L}(\mathbf{H})$ on linear ordered structures. In Gottlob's terminology, an $\mathcal{L}(\mathbf{H})$ -sentence Φ is in *Stewart normal form*, if it is of the form

$$\exists \vec{x} \left(\mathbf{H}_{(1)} \vec{y} \phi_1(\vec{x}, \vec{y}) \wedge \neg \mathbf{H}_{(2)} \vec{z} \phi_2(\vec{x}, \vec{z}) \right),$$

where ϕ_1 and ϕ_2 are first-order. This normal form is inspired by (Stewart 1993a; Stewart 1993b), hence the name. Clearly the Henkin depth of every formula in Stewart normal form is at most one. Gottlob proves that on the class of linear ordered structures for every $\mathcal{L}(\mathbf{H})$ -sentence Φ there exists an $\mathcal{L}(\mathbf{H})$ -sentence Ψ in Stewart normal form, that expresses the same property.

This result cries out for an effective translation procedure from $\mathcal{L}(\mathbf{H})$ into \mathbf{H}^+ of course, but unfortunately I cannot provide it. The translation hinges on finding a way of reducing the number of Henkin prefixes in a quantifier block. It gives some insight in the problem to show that

$$\left(\begin{array}{cc} \forall u_1 & \exists v_1 \\ \forall u_2 & \exists v_2 \end{array} \right) \left(\begin{array}{cc} \forall x_1 & \exists y_1 \\ \forall x_2 & \exists y_2 \end{array} \right) \phi$$

is equivalent to

$$\left(\begin{array}{cccc} & & \forall u_1 & \exists v_1 \\ & & \forall u_2 & \exists v_2 \\ \forall u_1 & \forall u_2 & \forall x_1 & \exists y_1 \\ \forall u_1 & \forall u_2 & \forall x_2 & \exists y_2 \end{array} \right) \phi,$$

see also (Blass and Gurevich 1986). But the real challenge is to find a way to handle negations appearing in between Henkin prefixes, making use of the finiteness of the structure and its linear order.

Dawar, Gottlob, and Hella (1998) raise the question whether $\mathcal{L}(\mathbf{H})$ captures $\mathbf{P}_{\parallel}^{\text{NP}}$ over *unordered* structures. Surprisingly, it turns out that $\mathcal{L}(\mathbf{H})$ does not capture $\mathbf{P}_{\parallel}^{\text{NP}}$ in the absence of a linear order, unless the *Exponential Boolean Hierarchy* collapses, amongst other hierarchies. In complexity theory the collapse of this hierarchy is considered to be highly unlikely.

Furthermore, a study by Hyttinen and Sandu (2000) implies that essentially one has to make use of the finiteness of the structures. Consider the logical languages

$$\begin{aligned} \mathbf{H}_1 &= \mathbf{H} \\ \mathbf{H}_m^+ &= \text{first-order closure of } \mathbf{H}_m \\ \mathbf{H}_{m+1} &::= \Phi \mid \mathbf{H}_k^n \vec{x} \Phi, \end{aligned}$$

where Φ ranges over the \mathbf{H}_m^+ -formulae and k, n are integers. Clearly the Henkin depth of any \mathbf{H}_m -sentence is k , and $\bigcup_m \mathbf{H}_m = \mathcal{L}(\mathbf{H})$. The authors prove that on the standard model of arithmetic the language \mathbf{H}_{m+1} has strictly stronger expressive power than \mathbf{H}_m^+ , for every $m \geq 1$.

4.11 Concluding remarks

In this chapter I showed that Henkin quantifiers can be seen to describe games whose imperfect information is brought about by limiting the number of memory cells of the involved agents. The idea of restricting the agents proved fruitful: function quantifiers and partially ordered connectives could be explained by absentmindedness and finite action arrays, respectively. The logic \mathbf{D} was shown to be a natural, though weaker, fragment of Σ_1^1 , cf. Theorem 4.6.7 and Corollary 4.9.3. Theorem 4.7.1 shows that \mathbf{D} contains a strict arity induced hierarchy, and Corollary 4.9.2 concludes that it is not closed under complementation. I observed that although on linear ordered structures, $\mathbf{D} = \Sigma_1^1$, on arbitrary structures it is the case that $\mathbf{D} < \Sigma_1^1$.

I gave another proof of Gottlob's result holding that $\mathcal{L}(\mathbf{H})$ captures $\mathbf{P}_{\parallel}^{\mathbf{NP}}$ on linear ordered structures, and derived the same theorem for $\mathcal{L}(\mathbf{D})$.

In Theorem 4.6.7 it was shown that \mathbf{D} comprises a fragment of Σ_1^1 whose sentences do not allow for a single existential variable to appear as the argument of a predicate variable. Hereby we arrive at a more refined way of dividing Σ_1^1 in syntactic fragments, than the division in *prefix classes*. For instance, the prefix class $\Sigma_1^1 \forall^*$ is the fragment of Σ_1^1 all of whose formulae have the form $\exists X_1 \dots \exists X_m \forall x_1 \dots \forall x_n \phi$, where ϕ quantifier-free. $\Sigma_1^1 \heartsuit$ extends the prefix class $\Sigma_1^1 \forall^*$ by replacing the constraint "quantifier-free" by "sober". It is interesting to compare the meta-theoretical properties of $\Sigma_1^1 \heartsuit$ and $\Sigma_1^1 \forall^*$, and other logics that stand in the same relationship (i.e., quantifier-free vs. sober). This research may put the program outlined in (Eiter, Gottlob, and Gurevich 2000; Gottlob 2004) in a broader context.

The discussion about the imperfect information games definable by partially ordered quantifiers gave some nice characterizations but was rather unsystematic. I meandered from limited cell agents to absentmindedness, without sketching the underlying framework. The reason being that no framework has been developed in which the interaction of game theory and (partially ordered) quantification theory can be studied. Setting up such a framework may thus lead to new quantifiers that in turn may lead to fully game-theoretic characterizations of the conundrums from complexity theory such as the $\mathbf{NP} = \mathbf{coNP}$ -problem.

Finally I mention a game-theoretic gap that needs to be filled in the interest of logic and descriptive complexity. We used a computational result from (Buss and Hay 1991) saying that every constant series of parallel queries can be reduced

to one session of parallel queries. The logical face of this theorem is the *flatness result*, holding that on linear ordered structures a $\mathcal{L}(\mathbf{H})$ -sentence of arbitrary Henkin depth has an equivalent $\mathcal{L}(\mathbf{H})$ -sentence of Henkin depth at most one. The question arises what would be the game-theoretic face of the aforementioned flatness result, in particular in the realm of *model comparison games*. Model comparison games are typically used to prove that some property is not expressible in a logic. As such they are tools par excellence to separate \mathbf{NP} from \mathbf{coNP} , for instance. A fertile approach to prove non-expressibility results is by simplifying model comparison games, in order to develop a library of intuitive tools for separating logics, cf. (Ajtai and Fagin 1990; Arora and Fagin 1997). Along these lines the flatness result concerning Henkin quantifiers may give rise to less complicated, but powerful, games to separate logics.

Chapter 5

Branching quantifiers

In this chapter I will focus on so-called *branching quantifiers* as they are studied in theoretical linguistics (Barwise 1979). Technically, branching quantifiers are akin to partially ordered quantifiers. I treat branching quantifiers in a chapter of their own, firstly because I use *strategic* games to give them a game-theoretic semantics, rather than extensive games. In particular, I show that branching quantifiers define a class of strategic games, such as Rock, Paper, Scissors and the Prisoner’s Dilemma. To do so I develop a framework in which logical expressions are evaluated in terms of Nash equilibria and may have any truth value in the interval between 0 and 1. This approach has not been taken before, although it was hinted at by various researchers (Blass and Gurevich 1986; van Benthem 2004; Ajtai 2005).

Secondly, the computational questions that I raise aim to bear relevance to the study of natural language semantics. I compare the complexity of branching quantifiers with non-branching quantifiers from the literature on natural language semantics, using an observation by van Benthem (1986) to show that they are “very tractable”. Surprisingly it turns out that branching quantifiers—including the ones whose semantics are generally unquestioned—have an **NP**-complete complexity, which in all likelihood means that they are intractable. I show that other quantifiers in natural language also have high complexity, suggesting that branching quantifiers are not in an isolated position in this respect.

5.1 Introduction

One of the best-known applications of partially ordered quantifiers is found in linguistics. It all began with Hintikka (1974), who claimed that the sentences (a)-(c) below have a logical form that cannot be described in first-order logic.

- (a) Some relatives of each villager and some relatives of each townsman hate each other.

- (b) Some book by every author is referred to in some essay by every critic.
- (c) Every writer likes a book of his almost as much as every critic dislikes some book he has reviewed.

Sentences like (a)-(c) are called *Hintikka sentences*. Hintikka claims that the logical form of (a) is

$$\left(\begin{array}{l} \forall x_1 \quad \exists y_1 \\ \forall x_2 \quad \exists y_2 \end{array} \right) (V(x_1) \wedge T(x_2) \rightarrow (R(x_1, y_1) \wedge R(x_2, y_2) \wedge H(y_1, y_2))), \quad (5.1)$$

containing a true partially ordered quantifier which is not first-order definable. In linguistics, partially ordered quantifiers are usually known as *branching quantifiers*. Henceforth I will refer to the doctrine holding that sentence (a) has a logical interpretation as in (5.1) as *Hintikka's Thesis*. Readers unfamiliar with Hintikka's claims may have a hard time grasping the meaning ascribed to (a)-(c); and he or she would not be the first one to disagree with Hintikka on the intended meaning of these sentences. Barwise (1979) summed up the arguments for and against branching as a construct in natural language. Barwise takes a sympathetic view of the idea of introducing partially ordered quantifiers to the linguist's toolbox. Barwise argues that the notation of partially ordered quantifiers should be used more broadly in formalizing the meaning of natural language, even if the logical form of the sentence can be given without partially ordered quantifiers. For instance, Barwise proposes to express the sentence

- (d) Some relatives of each townsman and every villager hate each other.

by

$$\left(\begin{array}{l} \forall x \quad \exists y \\ \quad \forall z \end{array} \right) (T(x) \wedge V(z) \rightarrow (R(x, y) \wedge H(y, z))).$$

Although the latter is logically equivalent to

$$\forall x \exists y \forall z (T(x) \wedge V(z) \rightarrow (R(x, y) \wedge H(y, z)))$$

it does not account for the wide scope of every in (d). In this manner, Barwise argues that branching quantifiers may give a more binding scope-sensitive representation. Remember that this is one of the motivations of Independence-friendly logic (Sandu 1993; Hintikka 1996; Hintikka and Sandu 1997), see also Chapter 3 of this dissertation. Branching quantifiers have been used in the analysis of natural language, not only to formalize Hintikka sentences, cf. (Boolos 1981; Gil 1982; van Benthem 1983).

As regards the Hintikka sentences (a)-(c), Barwise (1979) rejects Hintikka's Thesis on empirical grounds. In Barwise's empirical tests, subjects were asked to judge the truth value of a certain Hintikka sentence in a given model. It turned out that the subjects' behavior was more in line with the proposition that

Hintikka sentences have a certain first-order reading than with the proposition that they have a branching reading.

Yet, Barwise (1979) supplied other sentences claiming that their logical form essentially relies on the partially ordered quantification. Barwise's examples are as follows:

- (e) Most philosophers and most linguists agree with each other about branching quantification.
- (f) Most of the boys and most of the girls dated each other.

I call sentences (e) and (f) *branching most* sentences. According to Barwise (1979) the contention of (f) is that the majority of the boys and the majority of the girls have all dated each other pairwise.

The linguistic debate that was aroused by Hintikka's claims seems to have settled on the conviction that sentences like (a)-(c) have a "plain" first-order logical form, but that (e) and (f) are examples of essential branching.

In this chapter I take up the game-theoretic and computational analysis of branching quantifiers, with a keen eye on linguistic applications.

A recurring complaint concerning Hintikka's game-theoretic semantics is that it only accounts for nested quantifiers \forall and \exists . Therefore, a game-theoretic analysis of branching most sentences would counter this complaint in both respects: it deals with *branching* instead of nesting, and with the quantifier *most*.

Somewhat surprisingly, complexity issues of natural language quantifiers have received only little attention in the literature although it was recognized that computational complexity "[carries] the promise of a new field of *computational semantics*, which, in addition to questions of logical and mathematical interest, has applications to *language learning* and to mental *processing* of natural language" (Westerståhl 1989, pg. 115, Westerståhl's italics).

By "complexity of a quantifier" I refer to its expression complexity, in a sense that shall be formalized below.¹ In my view, it is natural to study this problem, as it matches the everyday problem of checking whether a sentence is true in a given situation. As such this study may be of interest to disciplines related to generalized quantifiers such as cognitive psychology, cf. (McMillan et al. 2005).

In Section 5.2, I set the stage by introducing the basics of generalized quantifier theory and branching quantifiers. I define a quantifier's expression complexity.

In Section 5.3, I give a game-theoretic account of branching quantifiers in terms of strategic games. Thereby, I give an argument against the view that game-theoretic semantics is only capable of accounting for the nesting of quantifiers \forall and \exists . Thus branching quantifiers define games in which the imperfect

¹The problem of deciding whether a set of generalized quantifier expressions is *satisfiable*, was studied in a publication by Pratt-Hartmann (2004).

information is introduced by parallel playing. Several researchers have hinted at the congenial mathematical structure that strategic game theory has in reserve for logic theory, but this topic has not been taken up until now. I conclude this section with some directions for further research that are opened up by the strategic framework.

Little research has been done on the interface of computability and generalized quantifiers. In Section 5.4, I recall van Benthem's work on finite automata and Mostowski and Wojtyniak's work on a computational analysis of Hintikka sentences.

In Section 5.5, I take up the complexity of natural language determiners and the quantifiers that one can construct from them by means of "traditional machinery", such as boolean operations and iterations (nesting) of quantifiers. Their complexity will be seen to be "very tractable" (**L**-computable).

In Section 5.6, I prove that in branching most sentences have **NP**-complete expression complexity, according to Barwise's reading. Given that $\mathbf{P} \neq \mathbf{NP}$, this means that the branching tool increases complexity even beyond what is tractable.

In Section 5.7, I consider three supposedly natural language quantifiers, including *Every ... a different* and *A few ... all*, and map out their expression complexity. I do so to point out that branching most's **NP**-complete complexity is not the only quantifier which has such a high complexity.

In Section 5.8, I conclude the chapter, with a small empirical experiment of my own concerning the actual usage of the branching of most sentences in natural language.

5.2 Prerequisites

In this section, I introduce the syntax and semantics of generalized quantifiers and define their expression complexity. Also I provide the Barwise reading of the branching most quantifier.

Syntax. Let Q be a *generalized quantifier symbol*. Every quantifier symbol has a *type*, that is a tuple of integers. Let Q be a generalized quantifier symbol of type $\langle k_1, \dots, k_n \rangle$. Then:

(GQ) If R_1, \dots, R_n are relation symbols of arity k_1, \dots, k_n and $\vec{x}_1, \dots, \vec{x}_n$ are strings of variables of length k_1, \dots, k_n , then

$$Q\vec{x}_1, \dots, \vec{x}_n (R_1(\vec{x}_1), \dots, R_k(\vec{x}_n))$$

is a Q -expression.

The symbol \top counts as a relation symbol of arbitrary arity.

Semantics. With every generalized quantifier symbol Q of type $\langle k_1, \dots, k_n \rangle$ is associated a function Q_S which assigns to each set S a subset of $\wp(S^{k_1}) \times \dots \times \wp(S^{k_n})$.

Let $Q\vec{x}_1, \dots, \vec{x}_n (R_1(\vec{x}_1), \dots, R_n(\vec{x}_n))$ be a Q -expression. Let \mathfrak{S} be a structure interpreting R_1, \dots, R_n . Then,

$$\mathfrak{S} \models Q\vec{x}_1, \dots, \vec{x}_n (R_1(\vec{x}_1), \dots, R_n(\vec{x}_n)) \quad \text{iff} \quad \langle R_1^{\mathfrak{S}}, \dots, R_n^{\mathfrak{S}} \rangle \in Q_S,$$

where $\top^{\mathfrak{S}} = S^n$, for every appropriate integer n .

The generalized quantifiers **All** and **Some** are familiar from first-order logic. Yet, in natural language one typically does not find the type $\langle 1 \rangle$ quantifiers \forall and \exists , but rather their *relativized* brothers of type $\langle 1, 1 \rangle$. The functions associated with some generalized quantifiers symbols can be found below:

$$\begin{aligned} \forall_S &= \{S\} \\ \exists_S &= \{X \in \wp(S) \mid X \neq \emptyset\} \\ R_S &= \{X \in \wp(S) \mid \|X\| > \|S - X\|\} \\ \text{All}_S &= \{\langle X, Y \rangle \in \wp(S)^2 \mid X \subseteq Y\} \\ \text{Some}_S &= \{\langle X, Y \rangle \in \wp(S)^2 \mid X \cap Y \neq \emptyset\} \\ \text{Most}_S &= \{\langle X, Y \rangle \in \wp(S)^2 \mid \|X \cap Y\| > \|X - Y\|\} \\ \text{A few}_S &= \{\langle X, Y \rangle \in \wp(S)^2 \mid \|X \cap Y\| < \|X - Y\|\}. \end{aligned}$$

The quantifier **R** is also known as the *Rescher quantifier*.

Complexity. Let $\Omega = Q\vec{x}_1, \dots, \vec{x}_n (R_1(\vec{x}_1), \dots, R_n(\vec{x}_n))$ be a Q -expression. Let \mathcal{F} be the class of all finite $\{R_1, \dots, R_n\}$ -structures. Then, Ω gives rise to the set

$$\{\mathfrak{S} \in \mathcal{F} \mid \mathfrak{S} \models \Omega\}. \quad (5.2)$$

Let \mathbf{C} be a complexity class. Let structures be encoded as in Chapter 4 in line with (Immerman 1999). I say that the *expression complexity* of Ω is in \mathbf{C} , if (5.2) is \mathbf{C} -computable. The expression complexity of Ω said to be \mathbf{C} -complete if (5.2) is \mathbf{C} -complete. A quantifier Q is called \mathbf{C} -computable, if all Q -expressions are \mathbf{C} -computable, and \mathbf{C} -complete if at least one Q -expression is \mathbf{C} -complete.

Branching quantifiers. In first-order logic, one may meaningfully combine quantifiers by *nesting* them. For instance, $\forall x A(x)$ and $\exists y B(y)$ are sentences with a properly defined semantics and so is $\forall x \exists y R(x, y)$. Nesting of first-order quantifiers can thus be seen as an operator that maps two type $\langle 1 \rangle$ quantifiers on one type $\langle 2 \rangle$ quantifier. In the literature on natural language quantifiers several operators are known, and they are addressed in Section 5.5 of this chapter. I will not repeat the computational terminology for compound quantifiers, but it should be clear.

Before leaving this section let me introduce the branching operator. To this end, a type $\langle 1, 1 \rangle$ quantifier Q is called *monotone increasing* ($\text{MON}\uparrow$), if for every set $A \subseteq S$, if $B \subseteq C \subseteq S$ and $\langle A, B \rangle \in Q_S$ then $\langle A, C \rangle \in Q_S$. The notion of a *monotone decreasing* quantifier is defined analogously, for $B \supseteq C$. Let Q, Q' be two $\text{MON}\uparrow$ quantifiers of type $\langle 1, 1 \rangle$. Define the *branching* of the quantifier symbols Q and Q' as the type $\langle 1, 1, 2 \rangle$ quantifier symbol $\text{Br}(Q, Q')$. On the assumption that $R^\mathfrak{S} \subseteq A^\mathfrak{S} \times B^\mathfrak{S}$, define its semantics as follows on structures \mathfrak{S} that interpret A, B, R :

$$\langle A^\mathfrak{S}, B^\mathfrak{S}, R^\mathfrak{S} \rangle \in \text{Br}(Q, Q')_S$$

iff

$$(\exists X \subseteq A^\mathfrak{S})(\exists Y \subseteq B^\mathfrak{S}) (\langle X, A^\mathfrak{S} \rangle \in Q_S \text{ and } \langle Y, B^\mathfrak{S} \rangle \in Q'_S \text{ and } X \times Y \subseteq R^\mathfrak{S}).$$

In this manner, (f) has as its logical form

$$\text{Br}(\text{Most}, \text{Most})xyz z' (BOY(x), GIRL(y), DATE(z, z')).$$

Branching quantifiers can also be defined for monotone decreasing quantifiers, and for pairs of quantifiers such as **Exactly** n that are neither monotone increasing nor monotone decreasing, see (van Benthem 1983). Mathematical generalizations of Br that cover these cases are explored in (Westerstahl 1987).

5.3 Strategic games and branching quantifiers

In Section 5.3.1, I develop a game-theoretic semantics for branching type $\langle 1 \rangle$ quantifiers, in terms of strategic games. This treatment is general in that it provides a game-theoretic interpretation for all branched quantifiers Q and Q' , including but not restricted to \forall and \exists . The underlying idea can be generalized to apply to quantifiers of higher types. Since the strategic approach is new in logic, I contemplate on further research in Section 5.3.2. The game-theoretic semantics for branching quantifiers, laid out in Section 5.3.1, gives an approach for solving a problem that faces the programme of game-theoretic semantics. Because, although game-theoretic semantics works out intuitively for first-order logic, it is unclear how to extend it for quantifiers other than \forall and \exists , and operations other than iteration.

5.3.1 GTS for branching quantifiers

Let me first adapt the previous definitions so that they apply to type $\langle 1 \rangle$ quantifiers Q . Q is called *universal*, if for some set S , $Q_S = \wp(S)$. Q is called *monotone increasing*, if for every set S , if $A \subseteq B$ and $A \in Q_S$ then $B \in Q_S$. Note that \exists, \forall, R are all $\text{MON}\uparrow$ quantifiers, but neither of them is universal.

Let \mathbf{Q}, \mathbf{Q}' be two $\text{MON}\uparrow$ quantifiers of type $\langle 1 \rangle$. The *branching* of \mathbf{Q} and \mathbf{Q}' is the type $\langle 2 \rangle$ quantifier $\text{Br}(\mathbf{Q}, \mathbf{Q}')$, such that for every structure \mathfrak{S} interpreting R , $R^{\mathfrak{S}} \in \text{Br}(\mathbf{Q}, \mathbf{Q}')_{\mathfrak{S}}$ iff

$$(\exists X \subseteq S)(\exists Y \subseteq S) (X \in \mathbf{Q}_{\mathfrak{S}} \text{ and } Y \in \mathbf{Q}'_{\mathfrak{S}} \text{ and } X \times Y \subseteq R^{\mathfrak{S}}).$$

Branching quantifiers admit of a highly regular *independence scheme*. For instance, the expression $\text{Br}(\mathbf{R}, \mathbf{R})xy R(x, y)$ would have $\mathbf{R}x(\mathbf{R}y/\{x\}) R(x, y)$ as its information-friendly rendering; and $\text{Br}(\forall, \exists)xy R(x, y)$ would read as the IF-formula $\forall x(\exists y/\{x\}) R(x, y)$. We see that the two simple quantifiers in a branching quantifier are *independent* of each other. In this section the independence scheme constituted by branching quantifier will be given a game-theoretic semantics in terms of parallel playing agents. In this manner, the expression $\text{Br}(\forall, \exists)xy R(x, y)$ indicates that Abelard and Eloise must pick their objects in parallel. If the chosen pair of objects is R -related, Eloise wins and otherwise Abelard wins. This one-shot, parallel way of playing is closely connected with *strategic games* that are one-shot games in which the players pick up their strategies in parallel, and the payoff is returned immediately afterward on the basis of the selected strategies. Surprisingly little has been said on the strategic viewpoint on semantic games. It was touched on in (Blass and Gurevich 1986) and (van Benthem 2004, pg. 193-4), but has not been taken up, cf. (Ajtai 2005).

In strategic game theory things get especially interesting when resorting to *mixed strategies*, that are probability distributions over *pure strategies*. In (Blass and Gurevich 1986, pg. 15-6) the authors hint at using mixed strategies, dwelling on the truth value of

$$\mathfrak{S} \models \text{Br}(\forall, \exists)xy (x = y). \quad (5.3)$$

They observe that the expression in 5.3 is neither true nor false in case $\|S\|$ has two or more objects. They suggest to let the formula in (5.3) have truth value $1/n$ where n is the cardinality of S . To account for their intuitions, Blass and Gurevich suggest to use mixed strategies in a game-theoretic framework. I will formalize matters and return to the truth value of (5.3) to show that the proposed formalization indeed meets the stated intuitions.²

Let \mathfrak{S} be a finite structure interpreting R and let \mathbf{Q}, \mathbf{Q}' be two $\text{MON}\uparrow$ type $\langle 1 \rangle$ quantifiers. Put $\Theta = \text{Br}(\mathbf{Q}, \mathbf{Q}')xy R(x, y)$. Let the *pre-strategic evaluation game* of Θ on \mathfrak{S} be the tuple $\langle N, \langle A_i \rangle_{i \in N}, \langle U'_i \rangle_{i \in N} \rangle$, where

- $N = \{\mathbf{Q}, \mathbf{Q}'\}$ are the players;
- $A_i = S$ is the set of *pure strategies* for player $i \in N$; and

²Importantly, my formalization is substantially different from the one proposed in (Blass and Gurevich 1986). In the latter publication, the authors sketch an approach that uses von Neumann's minimax Theorem. As far as I can see, this account can only handle branching quantifiers $\text{Br}(\mathbf{Q}, \mathbf{Q}')$, for $\mathbf{Q}, \mathbf{Q}' \in \{\forall, \exists\}$. My aim is to develop a framework that can deal also with branching most expressions.

- $U'_i : A_Q \times A_{Q'} \rightarrow \{0, 1\}$ is the *utility function* for player $i \in N$ such that

$$U'_i(a, a') = \begin{cases} 1 & \text{if } \langle a, a' \rangle \in R^\mathfrak{S} \\ 0 & \text{if } \langle a, a' \rangle \notin R^\mathfrak{S}. \end{cases}$$

Note that U'_Q is defined independent from Q . For this reason I will henceforth simply write U' , suppressing the subscript that indicate the player to whom the utility function belongs.

Let $\Delta(S)$ denote the class of probability distributions over a finite set S . A *probability distribution* over a finite set S is a function δ of type $S \rightarrow [0, 1]$, such that $(\sum_{a \in S} \delta(a)) = 1$. Observe that $\Delta(\emptyset)$ is ill-defined. If $\delta \in \Delta(S)$ and $a \in S$, let $\delta(a) \in \mathbb{R}$ be the probability assigned to a by δ . Let the *support* of $\delta \in \Delta(S)$, symbolically $Supp(\delta)$, be the set of objects in S to which δ assigns a non-zero probability. Call a probability distribution $\delta \in \Delta(S)$ *uniform*, if it assigns to every object in its support equal probability.

Let the *strategic evaluation game* of Θ on \mathfrak{S} , denoted $Str\text{-}game(\Theta, \mathfrak{S})$, be the tuple $\langle N, \langle \Delta(S)_i \rangle_{i \in N}, \langle U_i \rangle_{i \in N} \rangle$, where

- $N = \{Q, Q'\}$ is as in the pre-strategic evaluation game of Θ on \mathfrak{S} ;
- $\Delta(S)_C$ is the *flat set of mixed strategies* for player $C \in N$, defined

$$\{\delta \in \Delta(S) \mid Supp(\delta) \in C_S \text{ and } \delta \text{ is uniform}\};$$

- $U_i : \Delta(S)_Q \times \Delta(S)_{Q'} \rightarrow \{0, 1\}$ is the *utility function* for player $i \in N$ such that

$$U_i(\delta, \delta') = \sum_{\langle a, a' \rangle \in S^2} \delta(a) \cdot \delta'(a') \cdot U'(a, a').$$

Again, the utility functions are defined without reference to the player i , for which reason I omit the subscript. So it is not the *goals* of the game that discriminate the players. Instead it is the strategies that are available to the players.

It needs notice that $\Delta(S)_C$ does not contain all mixed strategies of S —it only contains those that are uniform and whose support is appropriate with respect to C . To avoid confusion, I coined the set $\Delta(S)_C$ “flat”.

Intuitively, one may envisage the protocol of a strategic game as follows: both players $i \in N$ pick a strategy δ_i from their set of flat mixed strategies $\Delta(S)_i$. This ends the active part of the game after which they receive the payoff $U(\delta_Q, \delta_{Q'})$. See (Osborne and Rubinstein 1994) for other conceptualizations of strategic games.

Like semantic games, I am not so much interested in particular runs of the game, but rather in statements we make about them.

5.3.1. DEFINITION. Let $G = \langle \{1, 2\}, \langle A_i \rangle_{i \in N}, \langle U_i \rangle_{i \in N} \rangle$ be a strategic game. Then, call the pair $\langle a_1, a_2 \rangle \in A_1 \times A_2$ a *Nash equilibrium* in G , if for every $a'_1 \in A_1$ and $a'_2 \in A_2$,

$$U_1(a'_1, a_2) \leq U_1(a_1, a_2) \quad \text{and} \quad U_2(a_1, a'_2) \leq U_2(a_1, a_2).$$

Let $\langle a_1, a_2 \rangle$ be a Nash equilibrium in G . Call $\langle a_1, a_2 \rangle$ a *Pareto optimal Nash equilibrium* in G , if there is no Nash equilibrium $\langle a'_1, a'_2 \rangle$ in G such that for all players i , $U_i(a_1, a_2) \leq U_i(a'_1, a'_2)$ and for some player j , $U_j(a_1, a_2) < U_j(a'_1, a'_2)$.

In every strategic game G , if the players have the same utility function, then there is at least one Pareto optimal Nash equilibrium. In particular it follows that in strategic evaluation games for branching quantifier expressions there is at least one Pareto optimal Nash equilibrium guaranteed. This insight justifies the following definition.

5.3.2. DEFINITION. Let \mathbf{Q}, \mathbf{Q}' be two $\text{MON}\uparrow$ type $\langle 1 \rangle$ quantifiers, let Θ be the expression $\text{Br}(\mathbf{Q}, \mathbf{Q}')xy R(x, y)$, and let \mathfrak{S} be a finite structure interpreting R . Then, let $U(\delta, \delta')$ be the *truth value* of Θ on \mathfrak{S} , where $\langle \delta, \delta' \rangle$ is a Pareto optimal Nash equilibrium in $\text{Str-game}(\Theta, \mathfrak{S})$.

Armed with this formal machinery, return to Blass and Gurevich's formula $\Omega = \text{Br}(\forall, \exists)xy (x = y)$. In the following result, that may also be appreciated as an example, I prove that the strategic machinery accounts for their intuition.

5.3.3. PROPOSITION. *Let \mathfrak{S} be a finite structure. Then, the truth value of Ω on \mathfrak{S} is $1/\|S\|$.*

Proof. Consider the flat set of mixed strategies of \forall and \exists . $\Delta(S)_\forall = \{\delta_\forall\}$, where δ_\forall is the uniform probability distribution whose support is S . Hence, for every $a \in S$, $\delta_\forall(a) = 1/\|S\|$. More interestingly, $\Delta(S)_\exists$ contains all uniform probability distributions δ such that $\|Supp(\delta)\| \neq \emptyset$. Fix any $\delta_\exists \in \Delta(S)_\exists$ and consider

$$\begin{aligned} U(\delta_\forall, \delta_\exists) &= \sum_{\langle a, a' \rangle \in S^2} \delta_\forall(a) \cdot \delta_\exists(a') \cdot U'(a, a') \\ &= \sum_{a \in S} \sum_{a' \in S} \delta_\forall(a) \cdot \delta_\exists(a') \cdot U'(a, a'). \end{aligned}$$

Since $U'(a, a') = 1$ iff $a = a'$, it is the case that $U(\delta_\forall, \delta_\exists)$ is in fact equal to

$$\begin{aligned} \sum_{a \in S} \delta_\forall(a) \cdot \delta_\exists(a) &= \sum_{a \in S} (1/\|S\|) \cdot \delta_\exists(a) \\ &= 1/\|S\|. \end{aligned}$$

The support of δ_\exists turns out to be immaterial to these calculations: For every probability distribution δ from $\Delta(S)_\exists$, it is the case that $U(\delta_\forall, \delta) = 1/\|S\|$. Therefore, $\langle \delta_\forall, \delta_\exists \rangle$ is a Pareto optimal Nash equilibrium in $\text{Str-game}(\Omega, \mathfrak{S})$, and consequently the truth value of Ω on \mathfrak{S} is $1/\|S\|$. \square

Proposition 5.3.3 shows that the theoretic possibility of truth values not being equal to 0 or 1, does materialize in strategic games. In fact, the proposition shows that for every integer n , there is a semantic game whose truth value is $1/n$. Proposition 5.3.3 does not relate truth values to the standard truth definition of branching expression. That is, it is observed that the truth value of Ω on a structure with n objects is $1/n$, yet from this mere fact one cannot deduce whether Ω is *true* under the customary notion of \models on the structure at hand. The connection between truth values and truth is established in Theorem 5.3.4.

5.3.4. THEOREM. *Let Q, Q' be two non-universal $\text{MON}\uparrow$ type $\langle 1 \rangle$ quantifiers. Let R be a binary relation symbol and let \mathfrak{S} be a finite structure interpreting R . Then, $\mathfrak{S} \models \text{Br}(Q, Q')xy R(x, y)$ iff the value of the strategic evaluation game $\text{Str-game}(\text{Br}(Q, Q')xy R(x, y), \mathfrak{S})$ is 1.*

Proof. Suppose $\mathfrak{S} \models \text{Br}(Q, Q')xy R(x, y)$, then by definition

$$(\exists X \subseteq S)(\exists Y \subseteq S) (X \in Q_S \text{ and } Y \in Q'_S \text{ and } X \times Y \subseteq R^{\mathfrak{S}}).$$

Since Q and Q' are $\text{MON}\uparrow$ but not universal, derive that X and Y are non-empty. For suppose otherwise and X were \emptyset . Then by monotonicity every subset of S would sit in Q_S . Hence, Q_S would be equal to $\wp(S)$ contradicting the assumption of Q 's not being universal.

Consider the uniform probability distributions δ_X and δ_Y that have support X and Y , respectively. Since X and Y are non-empty, these distributions are properly defined. Furthermore, observe that $\delta_X \in \Delta(S)_Q$ and that $\delta_Y \in \Delta(S)_{Q'}$, since the support of δ_Z is Z , for $Z \in \{X, Y\}$. As regards the payoff in case Q and Q' play δ_X and δ_Y , consider the equivalences

$$\begin{aligned} U(\delta_X, \delta_Y) &= \sum_{\langle a, a' \rangle \in S^2} \delta_X(a) \cdot \delta_Y(a') \cdot U'(a, a') \\ &= \sum_{a \in S} \sum_{a' \in S} \delta_X(a) \cdot \delta_Y(a') \cdot U'(a, a') \\ &= \sum_{a \in \text{Supp}(\delta_X)} \sum_{a' \in \text{Supp}(\delta_Y)} \delta_X(a) \cdot \delta_Y(a') \cdot U'(a, a') \\ &= \sum_{a \in X} \sum_{a' \in Y} \delta_X(a) \cdot \delta_Y(a') = 1, \end{aligned}$$

since $X \times Y \subseteq R^{\mathfrak{S}}$. The maximal value is clearly 1, whence $\langle \delta_X, \delta_Y \rangle$ is a Pareto optimal Nash equilibrium. Therefore the value of the strategic game is 1 and the claim follows.

The converse direction runs along similar lines. □

Theorem 5.3.4 reports the adequacy of strategic evaluation games for modeling the notion of truth for two arbitrary, type $\langle 1 \rangle$ quantifiers. It is fairly easy to

see that a stronger result can be established in at least two respects. No definition hinges essentially on the fact that there are only two players, and every definition can be extended to facilitate the game $Str\text{-}game(\text{Br}(\mathbf{Q}_1, \dots, \mathbf{Q}_n)\vec{x} R(\vec{x}), \mathfrak{S})$, for arbitrary n . Furthermore, extensions toward type $\langle 1, 1 \rangle$ quantifiers are readily envisioned by letting the mixed strategies be probability distributions over pairs of objects. A more substantial obstacle is the restriction to monotone increasing quantifiers. Barwise (1979) observed that the following natural language expression makes perfect sense, containing two monotone decreasing quantifiers:

(g) Few boys and at most three girls have all dated each other.

My game-theoretic analysis runs into problems at this point, for consider the type $\langle 1 \rangle$ quantifier **Few** that is the non-relativized cousin of the type $\langle 1, 1 \rangle$ quantifier **A few**. For convenience, put $\text{Few}_S = \wp(S) - \mathbf{R}_S$. Clearly, **Few** is monotone decreasing and for every non-empty set S , Few_S contains \emptyset . For this reason, $\Delta(S)_{\text{Few}}$ cannot be defined analogous to the flat set of mixed strategies $\Delta(S)_{\mathbf{C}}$ for $\text{MON}\uparrow$ quantifier \mathbf{C} , since there is no probability distribution whose support is the empty set.

As it happens, branching monotone decreasing quantifiers are the odd one out from a model-theoretic point of view. Westerståhl (1995, Proposition 1.9.3) shows, namely, that the truth conditions of $\text{Br}(\mathbf{Q}, \mathbf{Q}')$ boil down to the truth conditions of the *cumulation* of \mathbf{Q} and \mathbf{Q}' , given that \mathbf{Q}, \mathbf{Q}' are monotone decreasing quantifiers of type $\langle 1, 1 \rangle$. (The cumulation of two quantifiers is defined in Section 5.5 of this chapter.) It would be interesting to see if the failure of the above introduced game-theoretic apparatus can be connected with branching coinciding with cumulation.

5.3.2 Contemplations on the strategic framework

Theorem 5.3.4 shows that strategic game theory can define a game-theoretic semantics for branching quantifiers. The question arises if one can set up also game-theoretic interpretations of operations other than branching in the strategic framework. For instance, can the operations of cumulation and iteration, that will be addressed in Section 5.5, also be understood from a strategic viewpoint? I will not address these questions here and leave them as challenges to the strategic framework.

It be noted that single $\langle 1, 1 \rangle$ quantifiers can be tackled using the very strategic framework I laid down. Strategic evaluation games for $\mathbf{Q}xy R(x, y)$ would thus be single player strategic games, i.e., perfect information optimization problems.

The other way around—departing from game theory—one wonders whether other solution concepts give rise to interesting concepts of truth. Truth for branching quantifiers was characterized in Theorem 5.3.4 using the solution concept of Nash equilibrium, but game theory provides a garden-variety of solution concepts, each of which may give rise to interesting notions of truth. By example, what

notions of truth are defined by the concept of dominant strategy and iterated removal of dominated strategies?

The very fact that branching quantifiers may not only have truth value 0 (false) and 1 (true), but any value in the interval $[0, 1]$ suggests another series of questions. In this respect, namely, Theorem 5.3.4 only scratched the surface of the structure that strategic viewpoint has to offer, as it characterizes truth in terms of having truth value exactly 1. By this token, more exciting notions of truth come to mind. Consider for instance the satisfaction relation \models_ε , where ε is a real in $[0, 1]$, such that $\mathfrak{S} \models_\varepsilon \text{Br}(\mathbf{Q}, \mathbf{Q}')xy R(x, y)$ iff the value of $\text{Str-game}(\text{Br}(\mathbf{Q}, \mathbf{Q}')xy R(x, y), \mathfrak{S}) \geq \varepsilon$.

In this manner, Theorem 5.3.4 contributes that the original definition of \models for branching expressions coincides with \models_1 . Clearly for any integer $n \geq 2$ it is the case that \models and $\models_{1/n}$ differ. Observe for instance that:

$$\mathfrak{S} \models_{1/n} \text{Br}(\forall, \exists)xy (x = y) \quad \text{and} \quad \mathfrak{S} \not\models_1 \text{Br}(\forall, \exists)xy (x = y), \quad (5.4)$$

for every structure \mathfrak{S} with $\|S\| \geq n$. Interesting and non-trivial questions pop up right away. Suppose one switches from \models_1 to $\models_{1-\varepsilon}$, for a very small ε . Intuitively, this means that one “relaxes” the notion of truth one entertains: for a branching quantifier expression to be true under $\models_{1-\varepsilon}$ it suffices there exists a Pareto optimal Nash equilibrium whose value is $1 - \varepsilon$, instead of the customary 1. Now, what are the model-theoretic differences between $\models_{1-\varepsilon}$ and \models_1 ?

The latter question is model-theoretically motivated, but also probabilistic approaches to logic come to mind, such as 0-1 laws for first-order logic and extensions thereof.

In my opinion the strategic framework sheds a fresh light on the concept of dependence in logic. Let me take in mind branching quantifier expressions or, more ambitiously, formulae from IF logic. Under game-theoretic semantics for IF logic, the IF-sentence Φ is true on a structure \mathfrak{S} iff Eloise has a winning strategy in the associated game. The latter condition, now, is equivalent to Eloise having a strategy returning utility 1 *independent* of the strategy played by Abelard. In this manner, we see that the notion of winning strategy that underlies game-theoretic semantics for IF logic is a notion that ipse facto ignores the possible dependence between the players’ strategies. By contrast, the satisfaction relation \models_ε is defined in terms of Nash equilibria, a notion that hinges on dependence between strategies. This observation justifies the conclusion that studying \models_ε also in the context of IF logic, may broaden our understanding of dependence not only on the level of quantifier-variable pairs, but also on the level of strategies.

The broad variety of these questions shows that the strategic viewpoint provides a promising view on various topics in logic. This said, I will now turn to the computational aspects of natural language quantifiers, and branching quantifier expression.

5.4 Related research

In the remainder of this chapter I will be concerned with the computational aspects of natural language quantifiers in general and branching quantifiers in particular. There has been little interaction between the study of natural language quantifiers and the theory of complexity. In the subsequent two paragraphs I recall relevant research that relates computational devices with natural language quantifiers.

5.4.1 Van Benthem’s semantic automata

The approach taken by van Benthem (1986) is described by the author as follows:

“An attractive, but never very central idea in modern semantics has been to regard linguistic expressions as denoting certain ‘procedures’ performed within models for the language. [...] Viewed procedurally, the quantifier has to decide which truth value to give when presented with an enumeration of the individuals in [the universe of the structure] marked for their (non-)membership of A and B .” (van Benthem 1986, pg. 151)

Van Benthem proceeds to establish the correspondence between *semantic automata*, recognizing the truth of a generalized quantifier expression on a structure, and the language in which the quantifier is definable. A type $\langle 1, 1 \rangle$ quantifier Q is said to be definable in a logical language \mathbf{L} , if there exists an \mathbf{L} -formula ϕ in the vocabulary $\{A, B\}$, such that for every $\{A, B\}$ -structure \mathfrak{S} it is the case that $\langle A^{\mathfrak{S}}, B^{\mathfrak{S}} \rangle \in Q_S$ iff $\mathfrak{S} \models \phi$.

Van Benthem’s automata are given bit strings in which 0 represents an $A \wedge \neg B$ object and 1 an $A \wedge B$ object. Under the assumption that the quantifiers at stake satisfy a set of natural constraints (CONS, EXT, and ISOM),³ the quantity and quality of not- A objects are irrelevant to the truth of $Qxy (A(x), B(y))$. See Figure 5.1.a and 5.1.b for the self-explanatory semantic automata that compute the quantifiers **All** and **An even number of**.

Van Benthem (1986) proves the following two theorems for type $\langle 1, 1 \rangle$ quantifiers, that are CONS, EXT, and ISOM.

³A type $\langle 1, 1 \rangle$ quantifier Q is

- CONS if $\langle A, B \rangle \in Q_S$ and $A \cap B = A \cap C$ implies $\langle A, C \rangle \in Q_S$;
- EXT if $\langle A, B \rangle \in Q_S$ and $S \subseteq S'$ implies $\langle A, B \rangle \in Q_{S'}$;
- ISOM if \mathfrak{S} and \mathfrak{T} are isomorphic implies $\langle A^{\mathfrak{S}}, B^{\mathfrak{S}} \rangle \in Q_S$ iff $\langle A^{\mathfrak{T}}, B^{\mathfrak{T}} \rangle \in Q_T$.

See (Westerståhl 1989) for intuitions behind these conditions.

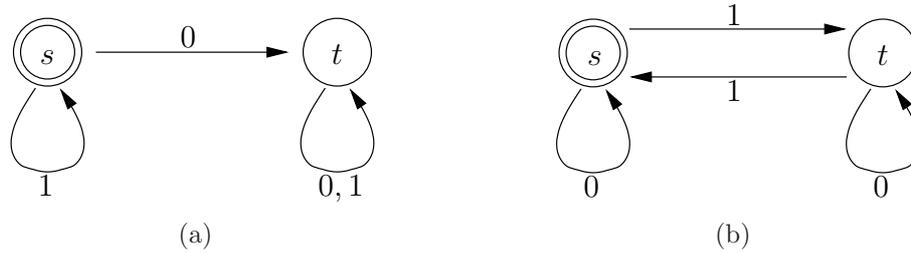


Figure 5.1: In both automata state s is the starting state and the only accepting state. Automaton (a) computes **All** and (b) computes **An even number of**. Note that the latter automaton has a *loop*, whereas the former is *acyclic*.

5.4.1. THEOREM (VAN BENTHEM (1986)). *The first-order definable quantifiers are precisely those which can be recognized by permutation-invariant acyclic finite state machines.*

Let *first-order additive logic*, denoted $\mathbf{FO}(+)$, be first-order logic extended with ternary $+$ relation and two constants a and b . Any formula from $\mathbf{FO}(+)$ will be interpreted as a standard arithmetical statement, where a is interpreted as the number of zeroes and b as the number of ones. For instance, **An even number of** would require the formula $\exists x (b = x + x)$, cf. (van Benthem 1986, pg. 162).

5.4.2. THEOREM (VAN BENTHEM (1986)). *The first-order additively definable quantifiers are precisely those which can be recognized by push-down automata.*

Theorems 5.4.1 and 5.4.2 can be read as results from descriptive complexity as they reveal the correspondence between a class of computational devices and linguistic means.⁴ Van Benthem argues that Theorem 5.4.2 is particularly valuable, since “on the whole, one finds natural language quantifiers, even the higher order ones, within the context-free realm. Thus, they are essentially ‘additive’ [...] There is some foundational significance to this observation, as *additive arithmetic* is still an axiomatizable (indeed, decidable) fragment of mathematics” (van Benthem 1986, pg. 154).

It is not addressed in the analysis of van Benthem’s whether results like Theorem 5.4.1 and 5.4.2 can be obtained for quantifiers with types other than $\langle 1, 1 \rangle$. I.e., what would a semantic automaton look like that computes *for every A, there is a B that...*? For a start, what alphabet should encode the input?—surely one cannot rely on a binary alphabet. These issues are not touched on in (van Benthem 1986), and it is not at all straightforward to see what the answer would be.

⁴Also, the characterization results have inspired cognitive psychologists to test the hypothesis holding that first-order definable quantifiers are processed differently than quantifiers definable in first-order additive logic, see (McMillan et al. 2005).

5.4.2 Expression complexity of Hintikka sentences

Recall the Hintikka sentence, cited earlier as (a), that is commonly known as the *townsman sentence*:

- (h) Some relative of each villager and some relative of each townsman hate each other,

whose logical reading contains a non-first-order definable, partially ordered quantifiers, given Hintikka's Thesis is true. In (Mostowski and Wojtyniak 2004) and a reading slightly different from Hintikka's proposal (5.1) is given, in that it gives a different treatment of relativization of the variables. Under the reading proposed in (Mostowski and Wojtyniak 2004) the logical reading of (h) is

$$\begin{aligned} \exists S_1 \exists S_2 \quad & (\forall x_1 \in V)(\exists y_1 \in S_1) R(x, y) \wedge \\ & (\forall x_2 \in V)(\exists y_2 \in S_1) R(x, y) \wedge \\ & (\forall y_1 \in S_1)(\forall y_2 \in S_2) H(y_1, y_2), \end{aligned} \tag{5.5}$$

where $(\forall x \in X) \phi$ abbreviates $\forall x (X(x) \rightarrow \phi(x))$ and $(\exists x \in X) \phi$ abbreviates $\exists x (X(x) \wedge \phi(x))$. In (Mostowski and Wojtyniak 2004) it is shown that the expression complexity of the generalized quantifier whose truth conditions are reflected in (5.5) is **NP**-complete. This result entails that verifying a Hintikka sentence on an arbitrary model is intractable, provided Hintikka's Thesis is true.

5.5 Complexity of natural language quantifiers

Van Benthem observed that on the whole, quantifiers found in natural language are definable in first-order additive logic. This observation bears consequences on the expression complexity of the type $\langle 1, 1 \rangle$ quantifiers in natural language. It was proved, namely, that first-order logic extended with the Rescher quantifier and the arithmetical operations of addition and multiplication captures the complexity class **TC**₀ on an appropriate class of structures, see (Immerman 1999). The class **TC**₀ was introduced in (Barrington et al. 1988) and has *circuits* as model of computation. Research on the complexity class **TC**₀ was "motivated by analogies with neural computing" (Barrington et al. 1988, pg. 48). I refrain myself from giving a formal exposure of this interesting complexity class, but let us take notice of the fact that **TC**₀ is contained in **L**, and that it is unknown whether it is strictly contained in **NP**.

Therefore, van Benthem's observation that natural language type $\langle 1, 1 \rangle$ quantifiers are typically definable in first-order additive logic, entails that a fortiori their expression complexity is in **L**.

5.5.1. PROPOSITION. *Let Q be a type $\langle 1, 1 \rangle$ quantifier that is definable in first-order additive logic. Then, Q 's expression complexity is in **L**.*

Proposition 5.5.1 only addresses type $\langle 1, 1 \rangle$ quantifiers (determiners). As I pointed out in Section 5.4.1, it is not straightforward to generalize van Benthem's framework so as to cover quantifiers of higher types. By contrast, the computational devices from complexity theory—including boolean circuits and Turing machines—are well-suited to deal with quantifiers of arbitrary types. Thus a more general theory of computational semantics may be defined naturally in terms of boolean circuits or Turing machines. As a by-product of these investigations, one is likely to obtain other measures of complexity for quantifiers than the ones stemming from van Benthem's automata framework.

In particular in this section I will focus on quantifiers one can obtain from combining two type $\langle 1, 1 \rangle$ natural language quantifiers Q and Q' through operations such as branching, although this operation will be the focus of discussion in Section 5.6. Note that the operations to come are usually defined for an arbitrary number of quantifiers, rather than two. For generalizations in this respect I refer the reader to (Westerstahl 1995). Note that the computational claims are not affected by this restriction.

Boolean operations. Two kinds of quantifier negation have been put forward: the *inner* negation, $Q\neg$, and the *outer* negation, $\neg Q$. The former resembles *sentence negation*, whereas the latter corresponds to *verb phrase negation*, as present in, respectively:

- (i) It is not the case that some philosopher walks.
- (j) Some philosopher does not walk.

Let the semantics of both negations thus be specified:

$$(Q\neg)_S = \{\langle X, Y \rangle \in \wp(S)^2 \mid \langle X, S - Y \rangle \in Q_S\} \quad (5.6)$$

$$(\neg Q)_S = \wp(S)^2 - Q_S. \quad (5.7)$$

There is the possibility of taking the conjunction or disjunction of quantifiers as in

- (k) Less than two or more than five philosophers run.

The respective semantics are naturally defined as follows:

$$\begin{aligned} (Q \vee Q')_S &= Q_S \cup Q'_S \\ (Q \wedge Q')_S &= Q_S \cap Q'_S. \end{aligned}$$

Iteration. The nesting of first-order quantifiers is an instance of *iteration*, but may be applied to any generalized quantifier:

- (l) Some soccer player read most books on oriental philosophy.

The iteration operator takes two type $\langle 1, 1 \rangle$ generalized quantifiers to one quantifier of type $\langle 1, 1, 2 \rangle$ and its semantics are specified in such a way that for every structure \mathfrak{S} ,

$$\langle A^{\mathfrak{S}}, B^{\mathfrak{S}}, R^{\mathfrak{S}} \rangle \in \text{It}(\mathbf{Q}, \mathbf{Q}')_S \quad \text{iff} \quad \langle A^{\mathfrak{S}}, \{a \in S \mid \langle B^{\mathfrak{S}}, R_a^{\mathfrak{S}} \rangle \in \mathbf{Q}'_S \} \rangle \in \mathbf{Q}_S,$$

where $R_a^{\mathfrak{S}}$ abbreviates $\{b \in S \mid \langle a, b \rangle \in R^{\mathfrak{S}}\}$.

Cumulation. A natural reading of many sentences involve *cumulated* quantifiers, first studied in (Scha 1981). The sentence, cited from (Westerståhl 1995, pg. 377),

(m) Sixty professors taught seventy courses at the summer school.

implies that 4200 courses would have been taught when one analyzes it by means of an iteration of the quantifiers **Sixty** and **Seventy**. Instead, a reading in which seventy courses in toto were taught is more plausible. This idea underlies the semantics of the *cumulation* operator, that is defined for every structure \mathfrak{S} such that $\langle A^{\mathfrak{S}}, B^{\mathfrak{S}}, R^{\mathfrak{S}} \rangle \in \text{Cum}(\mathbf{Q}, \mathbf{Q}')_S$ iff

$$\langle A^{\mathfrak{S}}, B^{\mathfrak{S}}, R^{\mathfrak{S}} \rangle \in \text{It}(\mathbf{Q}, \text{Some})_S \quad \text{and} \quad \langle B^{\mathfrak{S}}, A^{\mathfrak{S}}, R^{\mathfrak{S}} \rangle \in \text{It}(\mathbf{Q}', \text{Some})_S.$$

Resumption. Resumed quantifiers allow one to quantify over pairs of objects, rather than objects solely, and can be used to analyze sentences like

(n) Most lovers will eventually hate each other.

$\text{Res}^2(\mathbf{Q})$ is a quantifier of type $\langle 2, 2 \rangle$. Its truth conditions are as follows, for every structure \mathfrak{S} :

$$\langle R_1^{\mathfrak{S}}, R_2^{\mathfrak{S}} \rangle \in \text{Res}^2(\mathbf{Q})_S \quad \text{iff} \quad \langle R_1^{\mathfrak{S}}, R_2^{\mathfrak{S}} \rangle \in \mathbf{Q}_{S^2}.$$

I claim without proof that these operation do not make the computational complexity increase.

5.5.2. PROPOSITION. *Let \mathbf{Q} and \mathbf{Q}' be type $\langle 1, 1 \rangle$ quantifiers that are definable in first-order additive logic. Then, the expression complexity of*

$$\neg\mathbf{Q}, \mathbf{Q}\neg, \mathbf{Q} \vee \mathbf{Q}', \mathbf{Q} \wedge \mathbf{Q}', \text{It}(\mathbf{Q}, \mathbf{Q}'), \text{Cum}(\mathbf{Q}, \mathbf{Q}'), \text{Res}^2(\mathbf{Q}, \mathbf{Q}')$$

is in \mathbf{L} .

In fact, a stronger claim can be made—I conjecture that \mathbf{TC}_0 is also closed under taking the above operations on quantifiers, but I skip a rigorous argumentation to this effect.⁵ In any case, the expression complexity of natural language

⁵The argument proceeds by showing that if two families of circuits compute \mathbf{Q} and \mathbf{Q}' , respectively, then there is a way to combine those with appropriate means and to obtain a family of circuits that computes $\text{It}(\mathbf{Q}, \mathbf{Q}')$, $\text{Cum}(\mathbf{Q}, \mathbf{Q}')$, etc.

quantifiers and the quantifiers one construes from them with the previous operators are “very tractable”.

Proposition 5.5.2 reports that, from the current computational viewpoint, the class \mathbf{L} is closed under the addressed operations. But, naturally, there are more viewpoints on this matter serving different agenda’s. The logician’s approach, for instance, would typically aim at understanding the operators’ impact on expressive power. Using expressive power as a measure of complexity, it may well turn out that the previous operators do increase complexity. This very subject was addressed in (Hella, Väänänen, and Westerståhl 1997) for, amongst others, the operators of branching and resumption.

5.6 Branching quantifiers and NP

In this section I show that the expression complexity of branching quantifiers $\text{Br}(\mathbf{Q}, \mathbf{Q}')$ can be \mathbf{NP} -complete, even for \mathbf{L} -computable \mathbf{Q} and \mathbf{Q} . Before I come to the proof of Theorem 5.6.2, let me show that \mathbf{NP} is closed under branching.

5.6.1. PROPOSITION. *Let $\mathbf{Q}_1, \mathbf{Q}_2$ be two $\text{MON}\uparrow$ type $\langle 1, 1 \rangle$ generalized quantifiers whose expression complexity is in \mathbf{NP} . Then, the expression complexity of $\text{Br}(\mathbf{Q}_1, \mathbf{Q}_2)$ is in \mathbf{NP} .*

Proof. Consider $\Theta = \text{Br}(\mathbf{Q}_1, \mathbf{Q}_2)xy (A(x), B(y), R(x, y))$, for quantifier symbols $\mathbf{Q}_1, \mathbf{Q}_2$ that are in accordance with the premise of the proposition. The truth definition of Θ on an arbitrary finite structure \mathfrak{S} reads as follows:

$$(\exists X \subseteq A^{\mathfrak{S}})(\exists Y \subseteq B^{\mathfrak{S}}) (\langle A^{\mathfrak{S}}, X \rangle \in (\mathbf{Q}_1)_S \text{ and } \langle B^{\mathfrak{S}}, Y \rangle \in (\mathbf{Q}_2)_S \text{ and } X \times Y \subseteq R^{\mathfrak{S}}) \quad (5.8)$$

Since \mathbf{Q}_i ’s expression complexity is in \mathbf{NP} , for $i \in \{1, 2\}$, it follows from Fagin’s Theorem that there exists a Σ_1^1 -formula Ψ_i , such that on an arbitrary structure \mathfrak{S} that interprets A, B, R , the following holds

$$\langle A^{\mathfrak{S}}, B^{\mathfrak{S}}, R^{\mathfrak{S}} \rangle \in (\mathbf{Q}_i)_S \text{ iff } \mathfrak{S} \models \Psi_i.$$

Therefore, (5.8) is equivalent to

$$(\exists X \subseteq A^{\mathfrak{S}})(\exists Y \subseteq B^{\mathfrak{S}}) (\mathfrak{S} \models \Psi_1 \text{ and } \mathfrak{S} \models \Psi_2 \text{ and } X \times Y \subseteq R^{\mathfrak{S}}) \quad (5.9)$$

It is straightforward to see that the expression in (5.9) can be characterized by a Σ_1^1 -sentence in the vocabulary $\{A, B, R\}$. Hence, the expression complexity of Θ is in \mathbf{NP} . \square

Proposition 5.6.1 is not particularly strong. It shows that \mathbf{NP} is the computational upper-bound for branching two quantifiers that have \mathbf{NP} expression complexity themselves. The following theorem shows that the expression complexity

for the branching most quantifier is **NP**-complete, even though the expression complexity of **Most** is in **L** (in fact, in **TC**₀).⁶

5.6.2. THEOREM. *Br(Most, Most) has NP-complete expression complexity.*

Proof. Membership of **NP** follows from Proposition 5.6.1, since **Most**'s expression complexity is in **NP**. In the remainder of this proof I reduce an **NP**-complete problem to the verification of a **Br(Most, Most)**-expression. To this end, I define a graph problem called **HALF SEMI-CLIQUE**, that questions whether in a digraph $\mathfrak{G} = \langle G, R^\mathfrak{G} \rangle$, there exist two sets of vertices $G_1, G_2 \subseteq G$ such that

$$2\|G_1\|, 2\|G_2\| > \|G\| \quad \text{and} \quad G_1 \times G_2 \subseteq R^\mathfrak{G}.$$

It is obvious that the following sentence expresses whether a graph \mathfrak{G} is in **HALF SEMI-CLIQUE**:

$$\mathfrak{G} \models \text{Br}(\text{Most}, \text{Most})xy (\top(x), \top(y), R(x, y)), \quad (5.10)$$

where \top is the “unary tautology”. To establish **NP**-completeness for the problem **HALF SEMI-CLIQUE** I reduce from the problem **BCBG**, which stands for **BALANCED COMPLETE BIPARTITE GRAPH** and can be found in (Garey and Johnson 1979, pg. 196). **BCBG** is the problem that has an integer k and a *bipartite* digraph $\mathfrak{G} = \langle G, R^\mathfrak{G} \rangle$ as instance. A digraph $\langle G, R^\mathfrak{G} \rangle$ is bipartite, if there exists a partition V_1, V_2 of its vertices G , such that $R^\mathfrak{G} \subseteq V_1 \times V_2$. The pair \mathfrak{G} and k is in **BCBG** iff there exist sets $W_i \subseteq V_i$ ($i \in \{1, 2\}$) both of size exactly k such that $W_1 \times W_2 \subseteq R^\mathfrak{G}$.

I show that a bipartite digraph \mathfrak{G} and an integer k sit in **BCBG** iff the digraph \mathfrak{H} is in **HALF SEMI-CLIQUE**. \mathfrak{H} is constructed from \mathfrak{G} and k as follows: Let $m = \|G\|$ and let $d = m - 2k + 1$. Then, $\mathfrak{H} = \langle H, R^\mathfrak{H} \rangle$ is as follows:

$$\begin{aligned} H &= G \cup U, \text{ where} \\ U &= \{u_1, \dots, u_d\}, \text{ containing only new objects} \\ R^\mathfrak{H} &= R^\mathfrak{G} \cup (V_1 \times U) \cup (U \times U) \cup (U \times V_2). \end{aligned}$$

Now suppose that a certain pair of \mathfrak{G} and k is in **BCBG**. Let V_1, V_2 be witnesses of the fact that \mathfrak{G} is bipartite; that is, $R^\mathfrak{G} \subseteq V_1 \times V_2$. Then, in \mathfrak{G} live two sets of vertices $W_i \subseteq V_i$ ($i \in \{1, 2\}$) both of size k , such that $W_1 \times W_2 \subseteq R^\mathfrak{G}$. Consider the sets $X_i = W_i \cup U$. By construction of $R^\mathfrak{H}$ it holds that $X_1 \times X_2 \subseteq R^\mathfrak{H}$. As to the size of X_i observe that it contains exactly $\|W_i\| + \|U\|$ elements, that is, $k + d = m - k + 1$. $\|H\|$ equals $m + d = 2m - 2k + 1$. Hence, $2\|X_i\| > \|H\|$.

Conversely, suppose \mathfrak{H} has two sets of vertices W_1 and W_2 , such that $W_1 \times W_2 \subseteq R^\mathfrak{H}$ and

$$2\|W_i\| > \|H\| = m + d = 2m - 2k + 1.$$

⁶The fact that the expression complexity of the branching most quantifier was unknown was brought to my attention by Jakub Szymanik, whom I gratefully acknowledge.

Therefore, $\|W_i\| \geq m - k$. Since \mathfrak{G} does not contain the nodes from U , consider the sets $V_i = W_i \cap (H - U)$, for $i \in \{1, 2\}$. Since U contains d elements, V_i minimally contains $(m - k) - d = k$ objects. Now harmlessly remove arbitrary elements from V_1 and V_2 until both are of size exactly k . By construction of $R^{\mathfrak{G}}$ and the fact that $W_1 \times W_2 \subseteq R^{\mathfrak{G}}$, conclude that $V_1 \times V_2 \subseteq R^{\mathfrak{G}}$. \square

Theorem 5.6.2 shows that verifying a branching most expression on an arbitrary structure is intractable. In Section 5.7.3 I show that under a certain restriction the expression complexity of the branching most quantifier is **P**-computable.

I conclude with two notes.

- The proof of Theorem 5.6.2 actually reveals a stronger result. The type $\langle 2 \rangle$ quantifier $\text{Br}(\mathbf{R}, \mathbf{R})$, namely, has **NP**-complete expression complexity as well. For one derives from the \top s that (5.10) holds iff $\mathfrak{G} \models \text{Br}(\mathbf{R}, \mathbf{R})xy R(x, y)$. Hence, the claim follows.
- There is nothing special to branching **Most**—any branching of proportional determiners as in

$\text{Br}(\text{More than } p \text{ percent, More than } q \text{ percent})$

has **NP**-complete expression complexity, where $0 < p, q < 1$ are rationals. The proof is analogous to the one of Theorem 5.6.2, yet one may find it convenient to reduce from the generalization of **BCBG**, that enjoys two parameters k_1 and k_2 that constrain the size of W_1 and W_2 , respectively. This variant is clearly **NP**-complete as it has $k_1 = k_2$ as a special case, which was complete for **NP**.

5.7 More complexity of quantifiers

Among operations like iteration, cumulation, and resumption, branching is the odd one out, when it comes to expression complexity. These observations may give rise to the question whether the semantics ascribed to branching should not be reconsidered, as its computational behavior takes such an isolated position. I avoid any such discussion here, but I give three more quantifiers that have reasonably high model checking complexity, relative to **L**.

5.7.1 Every... a different...

Consider the type $\langle 1, 1, 2 \rangle$ quantifier that allows expressions of the form

$$\text{Every... a different... }xyz z' (A(x), B(y), R(z, z')), \quad (5.11)$$

and is true on an $\{A, B, R\}$ -structure \mathfrak{G} iff there exists a bijection f from $A^{\mathfrak{G}}$ to $B^{\mathfrak{G}}$, such that for every $a \in A^{\mathfrak{G}}$, $\langle a, f(a) \rangle \in R^{\mathfrak{G}}$. This interpretation follows

a proposal by van Benthem (1983). In my experience people tend not to give this as the preferred reading. But as I did not find much consensus on this topic anyway, why not treat this as one amongst equal.

It is easy to see that **Every... a different...** is **NP**-computable. As a lower bound, I show that the problem of **BIPARTITE MATCHING** reduces to the verification of **Every... a different...** It is well-known that **BIPARTITE MATCHING** is in **P**, although to the best of today's knowledge no completeness proof is known with respect to this class. This state of affairs may be taken to imply that it probably does not sit in $\mathbf{L} \subseteq \mathbf{NL}$. **BIPARTITE MATCHING** is the graph property holding all bipartite graphs $\langle G_1, G_2, R^\mathfrak{G} \rangle$, such that $R^\mathfrak{G} \subseteq G_1 \times G_2$. A bipartite graph \mathfrak{G} belongs to **BIPARTITE MATCHING** iff there is a bijection f from G_1 to G_2 such that for every $a \in G_1$, $\langle a, f(a) \rangle \in R^\mathfrak{G}$.

5.7.1. PROPOSITION. *The expression complexity of **Every... a different...** is at least as hard as **BIPARTITE MATCHING**.*

Proof. Reducing **BIPARTITE MATCHING** to the verification of the quantifier **Every... a different...** on an bipartite graph comes almost for free: For every bipartite graph $\mathfrak{G} = \langle G_1, G_2, R^\mathfrak{G} \rangle$, it is the case that

$$\langle G_1, G_2, R^\mathfrak{G} \rangle \in \text{Every... a different...}_{G_1 \cup G_2} \quad \text{iff} \quad \mathfrak{G} \in \text{BIPARTITE MATCHING}.$$

This concludes the proof. □

5.7.2 A few... all...

Consider the following sentence:

- (o) Less than twenty percent of the world population consumes more than ninety percent of the natural resources.

Turning to discrete—and more cheerful—environments, every professional Dutch soccer team can win at most three national prizes over one season: *de Johan Cruyff-schaal*, *de Beker*, and *kampioenschap van de Eredivisie*. In this context, consider the following sentence:

- (p) Two (or less) teams won all prices.

Under the iterated reading this sentence means that either one team won all three prices, or two teams both won all three prices. In the latter case, there would be two *kampioenen van de Eredivisie*, which cannot be of course. So iteration does not give the intended meaning to (p). Under the cumulative reading (p) is truth equivalent to:

- (q) Two (or less) teams won some price, and every price was won by a team.

Since every price is always won by exactly one team, this renders (q) truth equivalent to saying that two (or less) teams won some price. But this expressed weaker a proposition than (p), so also the cumulative reading fails to give an intuitive account.

In fact, I have not found any operator in the literature that yields a sound reading when taking **A few** and **All** as arguments, although there is some similarity with Keenan's (1992, pg. 209) *comparative dependent determiners*. For the current purposes it suffices to introduce a $\langle 1, 1, 2 \rangle$ quantifier **A few ... All ...**, rather than an operator that would map the quantifiers **A few** and **All** on a quantifier with the same semantics. Thus, I define as follows: Let \mathfrak{S} be an $\{A, B, R\}$ -structure, then $\langle A^{\mathfrak{S}}, B^{\mathfrak{S}}, R^{\mathfrak{S}} \rangle \in \mathbf{A\ few\ \dots\ All\ \dots}_S$ iff

$$(\exists U \subseteq S) ((U, A^{\mathfrak{S}}) \in \mathbf{A\ few}_S \text{ and } (\forall x \in B^{\mathfrak{S}})(\exists y \in U) R^{\mathfrak{S}}(x, y)).$$

Observe that **A few ... All ...** $xyz z'$ ($TEAM(x), PRICE(y), WIN(z, z')$) gives the intended meaning to (p), modulo the semantic difference between **A few** and **Two or less**.

On syntactic grounds, the truth condition of **A few ... All ...** is Σ_1^1 . In fact, the quantifier enjoys the strong computational behavior of existential, second-order logic, as it has **NP**-complete expression complexity. This I prove by reducing the problem **HALF COVER** to the verification of **A few ... All ...** on arbitrary finite structures. **HALF COVER** is the following problem: Let X be a set and let $\mathcal{Y} \subseteq \wp(X)$ be a collection of subsets of X . The pair constituted X and \mathcal{Y} sits in **HALF COVER** iff there exists a subset \mathcal{Y}' of \mathcal{Y} , such that $2\|\mathcal{Y}'\| < \|\mathcal{Y}\|$ and $\bigcup \mathcal{Y}' = X$; or using the natural language quantifier at stake, iff a few sets from \mathcal{Y} contain all elements in X .

HALF COVER can be proved **NP**-complete by reducing **MINIMUM COVER** to it, see (Garey and Johnson 1979, pg. 222). **MINIMUM COVER** is similar to **HALF COVER**, except for the fact that it requires $\|\mathcal{Y}'\| \leq k$, for some fixed parameter k . The reduction is straightforward for which reason I omit it.

5.7.2. PROPOSITION. *Expression complexity of **A few ... All ...** is **NP**-complete.*

Proof. To reduce **HALF COVER** to the verification of an **A few ... All ...** expression on an arbitrary finite structure, consider an instance constituted by X and \mathcal{Y} as above. Let the tuple $\mathfrak{Z} = \langle Z, ELT^{\mathfrak{Z}}, SET^{\mathfrak{Z}}, IN^{\mathfrak{Z}} \rangle$ be a structure that encodes X and \mathcal{Y} , in such a way that

$$\begin{aligned} Z &= X \cup \mathcal{Y} \\ ELT^{\mathfrak{Z}} &= X \\ SET^{\mathfrak{Z}} &= \mathcal{Y} \\ IN^{\mathfrak{Z}} &= \{\langle x, y \rangle \in X \times \mathcal{Y} \mid x \in y\}. \end{aligned}$$

The predicate SET^3 contains all sets in Z and ELT^3 all elements in X . I claim that

$$\langle X, \mathcal{Y} \rangle \in \text{HALF COVER} \quad \text{iff} \quad \langle SET^3, ELT^3, IN^3 \rangle \in \text{A few} \dots \text{All} \dots_Z.$$

Spelling out the truth condition of the quantifier on \exists yields

$$(\exists U \subseteq Z) ((U, SET^3) \in \text{A few}_Z \text{ and } (\forall x \in ELT^3)(\exists y \in U) IN^3(x, y)),$$

which is simply a reformulation of HALF COVER in pseudo-formal notation. \square

5.7.3 Disjoint halves

In Section 5.6, I showed that the quantifier expression

$$\text{Br}(\text{Most}, \text{Most})xyz z' (A(x), B(y), R(z, z'))$$

is **NP**-complete. I will show that if A and B are known to be complementary, then the expression complexity is **P**-solvable. The resulting quantifier can still be considered a branching quantifier, but I prefer to treat it in its own right, as it may have linguistic relevance.

Consider for instance the sentence

(r) One half of the professors hate the other half.

According our intuitions the intended reading of this sentence relies on the type $\langle 1, 2 \rangle$ quantifier **Disjoint halves**, that is defined such that for every $\{A, R\}$ -structure \mathfrak{S} , $\langle A^{\mathfrak{S}}, R^{\mathfrak{S}} \rangle \in \text{Disjoint halves}_S$ iff there exists a partition of $A^{\mathfrak{S}}$ into X, Y , such that $\|X\| = \|Y\|$ and $X \times Y \subseteq R^{\mathfrak{S}}$.

5.7.3. PROPOSITION. *The expression complexity of Disjoint halves is in P.*

Proof. Consider an $\{A, R\}$ -structure \mathfrak{S} . Whether $\langle A^{\mathfrak{S}}, R^{\mathfrak{S}} \rangle \in \text{Disjoint halves}_S$ is computed through the following steps:

- Remove all non- $A^{\mathfrak{S}}$ objects from S and compute $R^{\mathfrak{S}}$ restricted to $A^{\mathfrak{S}}$: $R|_A = R^{\mathfrak{S}} \cap (A^{\mathfrak{S}})^2$.
- It is easy to see that if two objects are not in the $R|_A$ -relation then they have to be in the same witness set. E.g., if one professor does not hate another, then they should be put in the same partition. Formally, for any two a, a' if $\langle a, a' \rangle \notin R|_A$ or $\langle a', a \rangle \notin R|_A$, then they should be in the same witness set. Here I make use of the requirement that X and Y partition $A^{\mathfrak{S}}$. Furthermore, if it is clear that the two couples a, a' and a', a'' have to be in the same witness set, then so must a and a'' . Therefore, compute the symmetric and transitive closure of the inverse of $R|_A$, denoted \overline{R}^* .

- \overline{R}^* constitutes an equivalence relation on A . Compute the equivalence sets: $\mathcal{A} = \{A_1, \dots, A_m\}$, such that for every $A_i \in \mathcal{A}$ it is the case that $a, a' \in A_i$ iff $\langle a, a' \rangle \in \overline{R}^*$.
- All sets in \mathcal{A} contain objects that must sit in the same witness set. But if two objects are in different equivalence sets, they can safely be put in different witness sets. The issue now is to partition \mathcal{A} in such a way that their respective unions contain an equal number of objects. This can be done by standard techniques from dynamic programming, see (Papadimitriou 1994, pg. 203).

Start out by constructing an $n \times m$ table, in which each cell initially is set to 0. If cell $\langle i, j \rangle$ has a 1, this intuitively denotes that there is a subset of $\{A_1, \dots, A_j\}$, whose union has cardinality i . The table is filled by iteratively considering a set $A_k \in \mathcal{A}$ and performing the routine step per cell:

- Write 1 on cell $\langle \|A_k\|, 1 \rangle$.
- If cell $\langle i, j \rangle$ has a 1, then write 1 on cell $\langle i + \|A_k\|, j + 1 \rangle$.

The very moment there exists a j such that cell $\langle n/2, j \rangle$ contains a 1 accept the input; otherwise reject it.

All of the above steps can be performed in polynomial time. □

The exact complexity of Disjoint halves is left as an open problem in this chapter.

5.8 Concluding remarks

In this chapter I studied branching quantifiers as defined in the literature on natural language semantics. In relation to branching quantifiers, two issues were addressed: game-theoretic semantics and computational analysis.

I developed a strategic framework in which branching quantifiers can be analyzed. In Theorem 5.3.4, I showed that this framework serves this end, to the point that one can define truth of a branching quantifier $\text{Br}(\mathbf{Q}, \mathbf{Q}')$ for any pair of non-universal, $\text{MON}\uparrow$ quantifiers \mathbf{Q} and \mathbf{Q}' . This shows that branching quantifiers can be seen to define strategic games, that involve a kind of imperfect information that is introduced by playing in parallel. Furthermore, I contemplated on directions for future research concerning the interface of strategic game theory and logic. I argued that the solution concept of Nash equilibrium may shed a fresh light on dependence in logic.

I used an observation of van Benthem's to claim that all natural language, type $\langle 1, 1 \rangle$ quantifiers (determiners) are \mathbf{L} -computable, hence "very tractable". By

contrast, Theorem 5.6.2 showed that branching most sentences are **NP**-complete, hence intractable (unless $\mathbf{P} = \mathbf{NP}$). The branching most quantifier is not the only natural language quantifier with a high complexity, as was shown in Section 5.7. The following table summarizes the computational results from this section, where Q, Q' stand for **FO**(+)-definable quantifiers:

Q	is	L -computable
$\neg Q, Q\neg$	are	L -computable
$It(Q, Q'), Cum(Q, Q'), Res^2(Q, Q')$	are	L -computable
$Br(Q, Q')$	is	NP -complete
Every . . . a different . . .	is	BIPARTITE MATCHING -hard
A few . . . All . . .	is	NP -complete
Disjoint halves	is	P -computable.

Let me close this chapter with some words about the impact of computational results for linguistic theorizing. In my view, the computational results put forward in this chapter can be used to back up a *density argument*. I observed that on the whole natural language determiners are **L**-computable, which is rather low. Furthermore, I observed that operations on **L**-computable quantifiers such as iteration and cumulation do not increase the complexity. In case it turns out that really the quantifiers in this chapter are the only ones with high complexity (provided their meaning is credible), one may be tempted to reconsider their status as natural language quantifiers on these very computational grounds. For such a density argument to have any impact, more results have to be established on the complexity of natural language quantifiers.

Discussing the applicability of complexity theory in linguistics, it needs notice that notions like hardness and completeness focus on the worst case behavior of quantifiers. What is the “average case” complexity of branching most expressions is not in any sense determined by Theorem 5.6.2. By this token, anyone who wishes to use **NP**-completeness results in a linguistic debate, is forced to argue why the worst case analysis is actually of relevance to the debate.⁷ Since completeness results are mostly due to farfetched instances, this may be an hairy affair.

In my analysis of branching most expressions, I departed from Barwise’s reading. Barwise defended this reading by an argument, that was justified by subjects’ intuitions on the semantics of branching most expressions. Yet, exploring subjects’ intuitions on quantifiers and discovering how people actually use them are two different matters. In fact, the observation that natural language users’ intuitions about natural language and actual usage thereof may differ substantially

⁷This argument criticizes also the train of reasoning set out in (Mostowski and Wojtyniak 2004), in which the provided **NP**-completeness result is taken to be the decisive assessment of the complexity of Hintikka sentences.

marks the movement in linguistics to analyze corpora of natural language expressions instead of intuitions. This movement also carries computational interest, for it may well turn out that branching most expressions, say, are used only under certain special conditions that influence their expression complexity.

As for the actual usage of branching most expressions, let me close this chapter with a small experiment of my own. The aim of the experiment is to find out whether branching most expressions are actually used in English. To this end I turned to the *British National Corpus* (BNC). Its content is from written and spoken language, and contains more than 100 million words, covering a broad variety of fields. The BNC can be accessed using on line available interfaces. I used the *Variation in English phrases and words* interface, developed at Brigham Young University, available from <http://view.byu.edu/>. Using this interface, the BNC can be searched for exact words, phrases, wild cards and combinations thereof. For instance, the expression “*most [n*]*” returns all occurrences of “most ...” in the corpus, where ... stands for a noun. In this manner the expression “*most [n*] and most [n*]*” returns the phrases in the corpus that partake in branching most sentences. None were returned.⁸

⁸I queried the database on March 27th, 2006. The actual settings of the query were as follows:

<i>Word/phrase</i>	most [n*] and most [n*]
<i>Sort by</i>	frequency
<i>Register 1 and 2</i>	–ignore–
<i>Min. freq. (both)</i>	0
<i>Limit (both)</i>	

Chapter 6

Scotland Yard

In this chapter I concern myself with the parlor game of Scotland Yard. Scotland Yard is a multi-player game, in which all but one player team up. The single player has the advantage of hiding his or her moves during most of the rounds. I will give a thorough analysis of Scotland Yard, showing that it can be conceived of as a perfect information game, by means of a power set argument. Surprisingly, it turns out that the naturally defined perfect information variant of Scotland Yard, in which the single player always has to show his whereabouts always, is equally hard as the imperfect information game. This shows that imperfect information is not bound to increase complexity, and that a more subtle analysis is needed of both general classes of games and specific parlor games.

6.1 Introduction

Background. The discipline of *combinatorial game theory* (CGT) deals almost exclusively with zero-sum games with perfect information. Although the existence of games with imperfect information is acknowledged in one of CGT's seminal publications (Berlekamp, Conway, and Guy 1982, pg. 16-7), only a marginal amount of literature appeared on games with imperfect information. The number of publications on games with perfect information is abundant and offers a robust picture of the computational behavior of games: One-person games or *puzzles* are usually solvable in **NP** and many of them turn out to be complete for this class.¹ Famous examples include the games of Minesweeper (Kaye 2000) and Clickomania (Biedl et al. 2002). Alternation increases complexity considerably: many two-player games have **PSPACE**-hard complexity, such as Go (Lichtenstein and Sipser 1980) and the semantic evaluation game of *quantified boolean formulae* (Stockmeyer and Meyer 1973; Schäfer 1978). Some even have **EXPTIME**-complete complexity. Typical examples in this respect are the games

¹To solve a game is to determine for an instance of the game whether a designated player has a winning strategy.

of Chess (Fraenkel and Lichtenstein 1981) and Checkers (Fraenkel et al. 1978). By and large, games with **EXPTIME**-complete complexity have a *loopy* nature, that is, the same configuration may occur over and over again. In real-life, loopy games may not be that much fun to play, as they allow for annoyingly long runs in which neither player makes any progress. Loopy runs are banned from Chess by postulating that, roughly speaking, no configuration of the game may occur more than three times.

Amusingly, putting an upper-bound on the duration of the game not only avoids loopy—and boring—sequences of play, but also has considerable computational impact. Papadimitriou (1994, pg. 460-2) argues that every game that meets the following requirements is solvable in **PSPACE**:

- the length of any legal sequence of moves is bounded by a polynomial in the size of the input; and
- given a “board position” of the game there is a polynomial-space algorithm which constructs all possible subsequent actions and board positions; or, if there aren’t any, decides whether the board position is a win for either player.

Note that Papadimitriou does not even mention the fact that this result concerns games of perfect information. The result stands due to the fact that the backwards induction algorithm can be run on the game’s game tree in **PSPACE**, given that it meets the above requirements.

As for games of imperfect information, some studies have been performed and their reports are basically a bad news show. Briefly, one can say that imperfect information increases the computational complexity of games. Convincing results are reported in (Koller and Megiddo 1992), where the authors show that it is possible to decide whether either player has a winning strategy in a finite, two-player game of perfect information using a polynomial time in the size of the game tree. On a positive note, they show that there is a **P**-algorithm that solves the same problem for games of imperfect information with perfect recall.² However, if one of the players suffers from imperfect recall the problem of deciding whether this player has a winning strategy is **NP**-hard in the size of the game tree.

In (Reif 1984; Peterson, Azhar, and Reif 2001) the authors regard computation trees as game trees. This view on computation trees is adopted from (Chandra, Kozen, and Stockmeyer 1981), in which so-called *alternating Turing machines* are considered which have existential and universal states. The aspect of alternation is reflected in the computation tree by regarding it as a game tree of a two-player game. The nodes corresponding to existential (universal) states belong to the existential (universal) player. From this viewpoint, non-deterministic Turing

²In Chapter 3, we saw that the perfect recall fragment of IF logic is considerably less complex, using expressive power as a measure of complexity.

machines have no universal states and thus give rise to one-player game trees. In (Reif 1984; Peterson, Azhar, and Reif 2001) this idea is extended to games of imperfect information. The authors define *private alternating Turing machines*, which give rise to computation trees that may be regarded as two-player game trees in which the existential player suffers from imperfect information, among other devices. It is shown that the space complexity of $f(n)$ of these machines is characterized in terms of the complexity of alternating Turing machines with space bound exponential in $f(n)$. Moreover, it is shown that private alternating Turing machines with three players—with two of the players teaming up—can recognize undecidable problems in constant space.

Dramatic as these results may be, being general studies they cannot tell us what the computational impact of the imperfect information found in actual games is. That is, games developed to be played rather than to be analyzed.³ It may well turn out that the imperfect information in these games has little computational impact and that the games themselves match the robust intuitions we have about the computational nature of perfect information games. As I pointed out before, there is but a small number of results concerning games with imperfect information, let alone computational studies of parlor games. For this reason, I will consider the game of Scotland Yard which gamers have enjoyed since 1983.⁴ Readers familiar with Scotland Yard will acknowledge that it is the imperfect information that makes the game an enjoyable waste of time and enthusiastic accounts of players' experiences with Scotland Yard are readily found on the Internet, for instance (Binz-Blanke 2006).

Game rules. I will now give a succinct description of the rules of Scotland Yard. Note however that this description serves merely to stress the kinship between the formalization used in this chapter and the actual game. A complete set of game rules of the formalization is supplied in the next section and should suffice to understand the formal details.

Scotland Yard is played on a game board which contains approximately 200 numbered intersections of colored lines denoting available means of transportation: yellow for taxis, green for buses, and pink for the Underground. A game is played by two to six people, one of them being *Mr. X*, the others teaming up and thusly forming Scotland Yard. They have a shared goal: capturing Mr. X. Initially, every player gets assigned a pawn and an intersection on the game board on which his or her pawn is positioned. Before the game starts every player gets a fixed number of tickets for every means of transportation. Mr. X and the cops

³Fraenkel (2002, pg. 476) makes the distinction between “*PlayGames*” and “*MathGames*”. The former being the games that “are challenging to the point that people will purchase them and play them”, whereas the latter games “are challenging to a mathematician [...] to play with or ponder about.”

⁴Scotland Yard is produced by *Ravensburger/Milton Bradley* and was prestigiously declared *Spiel des Jahres* in 1983.



Figure 6.1: The box of Scotland Yard and its items, amongst which the game board, Mr. X’s move board, and the players’ pawns. This picture is reproduced with permission of *Ravensburger*.

move alternatingly, Mr. X going first.

During every stage of the game, each player—be it Mr. X or his adversaries—takes an intersection in mind connected to his or her current intersection, subject to him or her owning at least one ticket of the appropriate kind. For instance, if a player would want to use the metro from Buckingham Palace, she would have to hand in her metro ticket. If either player is out of tickets for a certain means of transportation, he cannot travel along the related lines. Every player’s set of tickets is known to all players at every stage of the game.

If a cop has made up her mind to move to an intersection, this is indicated by her moving the pawn under her control to the intersection involved. However, when Mr. X has made up his mind he secretly writes the number of the intersection at stake at the designated entry of the *move board* and covers it with the ticket he has used. The cops know what means of transportation Mr. X has been using, but do not know his position. After round 3, 8, 13, 18, and 24, however, Mr. X is forced to show his whereabouts by putting his pawn on his current hideout.

The game lasts for 24 rounds during which Mr. X and the cops make their moves. If at any stage of the game any of the cops is at the same intersection as Mr. X the cops win.⁵ If Mr. X remains uncaught until after the last round, he

⁵Note that this is the description of the actual Scotland Yard game. My formalization—to be provided—has slightly different winning conditions.

wins the game. Cops who have a suspicious nature may want to check whether Mr. X's secret moves were consistent with the lines on the game board when the game is over. To this end, they would match the numbers on the move board with the returned tickets. If it turns out that Mr. X has cheated at any point, he loses no matter what the outcome of the game.

In view of these game descriptions, the generalization of the Scotland Yard game in Definition 6.1.1 may strike the reader as a natural abstraction. The reader will observe that the number of means of transportation is reduced to one and that the game board is modeled by a directed graph. However, all results in this chapter can be taken to hold for instances where several means of transportation and undirected graphs are involved.

6.1.1. DEFINITION. Let $G = \langle V, E \rangle$ be a finite, connected, directed graph with an out-degree of one or higher. That is, for every $v \in V$, there is a $u \in V$, such that $E(v, u)$. Let $u, v_1, \dots, v_n \in V$. Let $f : \{1, \dots, k\} \rightarrow \{show, hide\}$ be the *information function*, for some integer $2 < k < |V|$. Then, let $\langle G, \langle u, v_1, \dots, v_n \rangle, f \rangle$ be a (*Scotland Yard*) *instance*. Most of the time it will be convenient to abbreviate a string of vertices v_1, \dots, v_n by \vec{v} . Conversely, $\vec{v}(i)$ shall denote the i th element in \vec{v} . By $\{\vec{v}\}$ I refer to the set of all vertices in \vec{v} .

For $U \subseteq V$ write $E(U)$ to denote the set $\{u' \in V \mid E(u, u'), \text{ for some } u \in U\}$. If $\vec{v}, \vec{v}' \in V^n$, then write $E(\vec{v}, \vec{v}')$ to denote that for every $1 \leq i \leq n$, $E(\vec{v}(i), \vec{v}'(i))$.

The information function f controls the imperfect information throughout the game. If round i has property $f(i) = hide$, Mr. X hides himself. As will be seen the information function gives an intuitive meaning to “adding” and “removing” imperfect information from a Scotland Yard game. For instance, if one restricts oneself to information functions with range $\{show\}$, Mr. X shows his whereabouts after every move and one is effectively considering a game of perfect information. Under the latter restriction, one has arrived at so-called *graph games*, also called *Pursuit* or *Cops and robbers*. For an exposition of the literature on graph games, consult (Goldstein and Reingold 1995).

Aims and structure. The aims of this chapter are twofold. Firstly, to pinpoint the computational complexity of a real game of imperfect information. Secondly, I go through a reasonable amount of effort to spell out the relation between the game of Scotland Yard and a game of perfect information that is highly similar to the former. More precisely, I show that the games' game trees are isomorphic, and that a winning strategy in the one game constitutes a winning strategy in the other and vice versa. These similarity results may convince the reader that in some cases the wall between perfect and imperfect information is not as impenetrable as one might induce from the scarce literature on the complexity of imperfect information games.

As I pointed out before, the definition of Scotland Yard instance abstracts away from features of the game of Scotland Yard that are inessential to this chapter's aims. In fact, all that a Scotland Yard instance holds is a graph, a set of vertices on which the pawns are initially positioned, the duration of the game, and a means to control the imperfect information. In my view, the level of abstraction employed in Definition 6.1.1 justifies one's conceiving Scotland Yard instances as graph games. For this reason, I think that my analyses are not solely relevant to the specific game of Scotland Yard, but to the theory of graph games in general. Nevertheless, I will continue to refer to the games under consideration using the colloquial Scotland Yard terminology.

In Section 6.2, I will define the extensive game form of the Scotland Yard game to which a Scotland Yard instance gives rise.

In Section 6.3, I define a perfect information variant of Scotland Yard. In this perfect information game, Mr. X picks up sets of vertices, but he does so in public.

In Section 6.4, I show that the games that have been introduced admit for a bijection between the imperfect information game's information partitions and the histories in the perfect information Scotland Yard game. Furthermore, I show that both games, under the bijection analysis, are equivalent, i.e., the cops have a winning strategy in the imperfect information game iff they have one in the perfect information game.

In Section 6.5, the computational results are presented. In accordance with many polynomially bounded two-player games, Scotland Yard is complete for **PSPACE**, despite its imperfect information. That is, the computational complexity of Scotland Yard does not change when one only considers information functions with range $\{show\}$.

In fact, if one would add more imperfect information to the extent that the information flow function has range $\{hide\}$, the resulting decision problem is easier: **NP**-complete. This is shown in Section 6.6.

Section 6.7 concludes the chapter.

6.2 Scotland Yard formalized

In this section, I define the extensive games with imperfect information that are constituted by Scotland Yard instances. I abstract away from some of the actual game's properties, for which reason one may regard the formal Scotland Yard games as abstract graph games.

Let $sy = \langle G, \langle u_*, \vec{v}_* \rangle, f \rangle$ be a Scotland Yard instance as in Definition 6.1.1. Before I define the extensive game form of the Scotland Yard game to which sy gives rise, let me formulate the game rules of the game under consideration in

terms of sy .

The digraph G is the board on which the actual playing finds place. In the initial situation of the game $n + 1$ pawns, named $\forall, \exists_1, \dots, \exists_n$, are positioned on the respective vertices $u_*, \vec{v}_*(1), \dots, \vec{v}_*(n)$ on the digraph. The game is played by the two players \exists and \forall over k rounds, and with every round $1 \leq i \leq k$ in the game the property $f(i) \in \{show, hide\}$ is associated. Note that I converted the n -player game of Scotland Yard, where $2 \leq n \leq 6$ into a two-player game in which one player controls all pawns $\exists_1, \dots, \exists_n$. Furthermore, for reasons of succinctness I use the symbol \forall to refer to Mr. X (male) and \exists to refer to the player controlling Scotland Yard (female). Somewhat sloppily, sometimes I will not make a strict distinction between a player and (one of) his or her pawns.

First fix $i = 1$, $u = u_*$, and $\vec{v} = \vec{v}_*$; now, round i of Scotland Yard goes as follows:

1. If for some $1 \leq j \leq n$, the pawns \forall and \exists_j share the same vertex, i.e., $u = \vec{v}(j)$, \forall is said to be *captured* (by \exists_j). If \forall is captured the game stops and \exists wins. If \forall is not captured and $i > k$ the game also stops but \exists loses.
2. \forall chooses a vertex u' , such that $E(u, u')$. If $f(i) = show$, \forall physically puts his pawn on u' . If $f(i) = hide$, he secretly writes u' on his move board making sure that it cannot be seen by his opponent. Set $u = u'$.
3. Player \exists chooses a vector $\vec{v}' \in V^n$, such that $E(\vec{v}, \vec{v}')$, and for every $1 \leq j \leq n$, moves pawn \exists_j to $\vec{v}'(j)$. Set $\vec{v} = \vec{v}'$.
4. Set $i = i + 1$.

Note that these game rules do not consider the possibility of either player getting stuck, as in not being able to move a pawn under his or her control along an edge. This goes without loss of generality, as the digraphs at stake are supposed to have out-degree ≥ 1 .

Furthermore, it should be borne in mind, that for \forall it is not a guaranteed loss to move to a vertex occupied by one of \exists 's pawns. The game only terminates after \exists has moved *and* one of her pawns captures \forall , unlike the game rules for the board game of Scotland Yard.

Observe that \exists loses only after k rounds of the game have been played. If it so happens that the game terminates after the j th round, for $j < k$, then \exists has won.

Scotland Yard is modeled as an extensive game with imperfect information in Definition 6.2.1. The upcoming definition and Definition 6.3.1 are notationally akin to the definitions in Section 2.2 and (Osborne and Rubinstein 1994).

6.2.1. DEFINITION. Let $sy = \langle G, \langle u_*, \vec{v}_* \rangle, f \rangle$ be a Scotland Yard instance. Then, let the *extensive Scotland Yard game constituted by sy* be defined as the tuple $SY(sy) = \langle N, H, P, \sim, U \rangle$, where

- $N = \{\exists, \forall\}$ is the set of players.
- H is the set of *histories*, that is, the smallest set containing $\langle u_* \rangle, \langle u_*, \vec{v}_* \rangle$ and is closed under actions taken by \forall and \exists :
 - If $h \langle u, \vec{v} \rangle \in H$, $\ell(h \langle u, \vec{v} \rangle) < k$, $u \notin \{\vec{v}\}$, and $E(u, u')$, then $h \langle u, \vec{v} \rangle \langle u' \rangle \in H$.
 - If $h \langle u, \vec{v} \rangle \langle u' \rangle \in H$ and $E(\vec{v}, \vec{v}')$, then $h \langle u, \vec{v} \rangle \langle u', \vec{v}' \rangle \in H$.

For $h \in H$, let $\ell(h)$ denote the number of rounds in h , that is the number of tuples not equal $\langle u_*, \vec{v}_* \rangle$. Define $\ell(\langle u_*, \vec{v}_* \rangle) = 0$. Somewhat unlike custom usage in game theory, the length $\ell(h)$ of history h does not coincide with the number of plies in the game. This notation is chosen to reflect the game rule saying that a history only terminates after \exists has moved.

Let \succ be the *immediate successor* relation on H . That is, the smallest relation closed under the following conditions:

- If $h, h \langle u \rangle \in H$, then $h \succ h \langle u \rangle$.
- If $h \langle u \rangle, h \langle u, \vec{v} \rangle \in H$, then $h \langle u \rangle \succ h \langle u, \vec{v} \rangle$.

A history that has no immediate successor is called a *terminal history*. Let $Z \subseteq H$ be the set of terminal histories in H .

- $P : H - Z \rightarrow \{\exists, \forall\}$ is the *player function* that decides who is to move in a non-terminal history. Due to the notational convention, the value of P is easily determined from the history's form, in the sense that $P(h \langle u \rangle) = \exists$ and $P(h \langle u, \vec{v} \rangle) = \forall$, no matter h , u , and \vec{v} .
- \sim is the *indistinguishability relation* that formalizes the imperfect information in the game. It is defined such that for any pair of histories $h, h' \in H$ of equal length, where

$$h = \langle u_*, \vec{v}_* \rangle \langle u_1, \vec{v}_1 \rangle \dots \langle u_i \rangle \quad \text{and} \quad h' = \langle u_*, \vec{v}_* \rangle \langle u'_1, \vec{v}'_1 \rangle \dots \langle u'_i \rangle \quad (6.1)$$

it is the case that $h \sim h'$, if

- (a) $\vec{v}_j = \vec{v}'_j$, for every $1 \leq j \leq i - 1$; and
- (b) $u_j = u'_j$, for every $1 \leq j \leq i$ such that $f(j) = \text{show}$.

The previous condition, considering histories as in (6.1), defines \sim only as a relation between histories h belonging to \exists . This reflects the fact that it is \exists who experiences the imperfect information. Somewhat unusual, I extend \exists 's indistinguishability relation \sim to histories in which \forall has to move. The

reader is urged to take this extension as a technicality.⁶ I put as follows: for any pair of histories $h, h' \in H$, where

$$h = \langle u_*, \vec{v}_* \rangle \langle u_1, \vec{v}_1 \rangle \dots \langle u_i, \vec{v}_i \rangle \quad \text{and} \quad h' = \langle u_*, \vec{v}_* \rangle \langle u'_1, \vec{v}'_1 \rangle \dots \langle u'_i, \vec{v}'_i \rangle \quad (6.2)$$

it is the case that $h \sim h'$, if

- (a) $\vec{v}_j = \vec{v}'_j$, for every $1 \leq j \leq i$; and
 - (b) $u_j = u'_j$, for every $1 \leq j \leq i$ such that $f(j) = \text{show}$.
- $U : Z \rightarrow \{\text{win}, \text{lose}\}$ is the function that decides whether a terminal history $h \langle u, \vec{v} \rangle$ is won or lost for \exists . Formally,

$$U(h \langle u, \vec{v} \rangle) = \begin{cases} \text{win} & \text{if } u \in \{\vec{v}\} \\ \text{lose} & \text{if } u \notin \{\vec{v}\}. \end{cases}$$

Usually, we have one utility function per player, but as the game is win-loss and zero-sum, it suffices to consider only one function.

Since \sim is reflexive, symmetric, and transitive it defines an equivalence relation on H . I write $\mathcal{H} \subseteq \wp(H)$ for the set of equivalence classes, or *information cells*, in which H is partitioned by \sim . That is, $\mathcal{H} = \{C_1, \dots, C_m\}$, where $C_1 \cup \dots \cup C_m = H$ and for every $1 \leq i \leq m$, if $h \in C_i$ and $h \sim h'$, then $h' \in C_i$. A standard inductive argument suffices to see that for every $C_i \in \mathcal{H}$ and pair of histories $h, h' \in C_i$, the length of h and h' coincides and $P(h) = P(h')$.

I lift the relation \succ to \mathcal{H} , using the same symbol: For any pair $C, C' \in \mathcal{H}$, I write $C \succ C'$ if there exists histories $h \in C$ and $h' \in C'$ such that $h \succ h'$. It is easy to see that if h, h' are histories in a cell $C \in \mathcal{H}$, then $P(h) = P(h')$. Thus, the player function is meaningfully lifted as follows: if $C \in \mathcal{H}$ and h is a history in C , then $P(C) = P(h)$. Call a cell $C \in \mathcal{H}$ *terminal* if all its histories are terminal.

Since I study an extension of \sim , the set \mathcal{H} partitions all histories in H . As I pointed out in the definition of \sim , if histories h and h' stand in the \sim relation and belong to \forall , this should not be taken to reflect any conceptual consideration about \exists 's experiences, as it is merely a technicality. Yet, if h and h' belong to \exists , to write $h \sim h'$ reflects genuine indistinguishability for player \exists between the two histories h and h' . In this manner, observe that a *subset* of \mathcal{H} is a familiar game-theoretic object from game-theory. Consider the set $\mathcal{H}_\exists = \{C \in \mathcal{H} \mid P(C) = \exists\}$, that partitions the set of histories that belong to \exists . I claim that \mathcal{H}_\exists is an *information set* in the sense of Section 2.2 and (Osborne and Rubinstein 1994), that is, it meets the action consistency requirement. To prove this claim it suffices to show that for every information cell $C \in \mathcal{H}_\exists$ no two histories $h, h' \in C$ can be distinguished

⁶Note that van Benthem (2001) argued that it is nothing but natural to define the indistinguishability relation of one player also over the over player's histories. By doing so van Benthem axiomatizes perfect recall games in a dynamic-epistemic framework.

on the basis of the actions that \exists can take at h and h' . Formally, for every $C \in \mathcal{H}_\exists$ and for every pair of histories $h, h' \in C$ it is the case that $A(h) = A(h')$.

To this end, let

$$A(h\langle u, \vec{v} \rangle\langle u' \rangle) = \{\vec{v}' \in V^n \mid h\langle u, \vec{v} \rangle\langle u' \rangle \succ h\langle u, \vec{v} \rangle\langle u', \vec{v}' \rangle\} = \{\vec{v}' \in V^n \mid E(\vec{v}, \vec{v}')\}$$

define the actions available to \exists after $h\langle u, \vec{v} \rangle\langle u' \rangle$. Let h and h' be histories as in (6.1) sitting in the same cell $C \in \mathcal{H}$. Then, by (a) $\vec{v}_{i-1} = \vec{v}'_{i-1}$ and therefore $A(h) = A(h')$. Hence, \mathcal{H}_\exists is an information set. (Note that information cells in \mathcal{H}_\exists are usually called *information partitions*.)

For future reference, I lay down the following proposition.

6.2.2. PROPOSITION. *Let $SY(sy) = \langle N, H, P, \sim, U \rangle$ be the Scotland Yard game constituted by sy . Then, the following statements hold:*

- (1) *If $h_1\langle u_1 \rangle \sim h_2\langle u_2 \rangle$ and $f(\ell(h_1\langle u_1 \rangle)) = \text{hide}$, then $h_1 \sim h_2$.*
- (2) *If $h_1\langle u_1 \rangle \sim h_2\langle u_2 \rangle$ and $f(\ell(h_1\langle u_1 \rangle)) = \text{show}$, then $h_1 \sim h_2$ and $u_1 = u_2$.*
- (3) *If $h_1\langle u_1, \vec{v}_1 \rangle \sim h_2\langle u_2, \vec{v}_2 \rangle$, then $h_1\langle u_1 \rangle \sim h_2\langle u_2 \rangle$ and $\vec{v}_1 = \vec{v}_2$.*
- (4) *If $h_1 \not\sim h_2$ and $h_1\langle u_1 \rangle, h_2\langle u_2 \rangle \in H$, then $h_1\langle u_1 \rangle \not\sim h_2\langle u_2 \rangle$.*

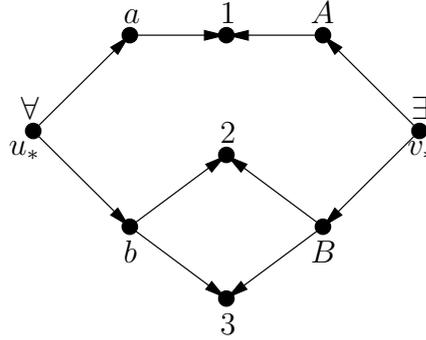
Proof. Readily observed from the definition of \sim in Definition 6.2.1. □

6.2.3. EXAMPLE. As an illustration of modeling a Scotland Yard instance⁷ as an extensive game with imperfect information, consider the digraph $G^\times = \langle V^\times, E^\times \rangle$, where

$$\begin{aligned} V^\times &= \{u_*, v_*, a, b, A, B, 1, 2, 3\} \\ E^\times &= \{\langle u_*, a \rangle, \langle u_*, b \rangle, \langle a, 1 \rangle, \langle b, 2 \rangle, \langle b, 3 \rangle, \langle v_*, A \rangle, \langle v_*, B \rangle, \langle A, 1 \rangle, \langle B, 2 \rangle, \langle B, 3 \rangle\}. \end{aligned}$$

For a depiction of G^\times , see Figure 6.2. Let f^\times be an information function such that $f^\times(1) = \text{hide}$ and $f^\times(2) = \text{show}$. Conclude the construction of the Scotland Yard instance sy^\times , by putting u_* and v_* as the initial vertices of \forall and \exists , respectively. In $SY(sy^\times)$, the set of histories H contains exactly the following histories:

⁷The digraph under consideration does not have out-degree ≥ 1 . The game terminates after two rounds, so adding reflexive edges on the vertices 1, 2, and 3 goes without affecting the winning conditions of the game.

Figure 6.2: The digraph G^\times , allowing for a two-round Scotland Yard game.

$\langle u_*, v_* \rangle$		
$\langle u_*, v_* \rangle \langle a \rangle$	$\langle u_*, v_* \rangle \langle a, B \rangle \langle 1 \rangle$	$\langle u_*, v_* \rangle \langle a, B \rangle \langle 1, 3 \rangle$
$\langle u_*, v_* \rangle \langle b \rangle$	$\langle u_*, v_* \rangle \langle b, A \rangle \langle 2 \rangle$	$\langle u_*, v_* \rangle \langle b, A \rangle \langle 2, 1 \rangle$
$\langle u_*, v_* \rangle \langle a, A \rangle$	$\langle u_*, v_* \rangle \langle b, A \rangle \langle 3 \rangle$	$\langle u_*, v_* \rangle \langle b, A \rangle \langle 3, 1 \rangle$
$\langle u_*, v_* \rangle \langle a, B \rangle$	$\langle u_*, v_* \rangle \langle b, B \rangle \langle 2 \rangle$	$\langle u_*, v_* \rangle \langle b, B \rangle \langle 2, 2 \rangle !$
$\langle u_*, v_* \rangle \langle b, A \rangle$	$\langle u_*, v_* \rangle \langle b, B \rangle \langle 3 \rangle$	$\langle u_*, v_* \rangle \langle b, B \rangle \langle 2, 3 \rangle$
$\langle u_*, v_* \rangle \langle b, B \rangle$	$\langle u_*, v_* \rangle \langle a, A \rangle \langle 1, 1 \rangle !$	$\langle u_*, v_* \rangle \langle b, B \rangle \langle 3, 2 \rangle$
$\langle u_*, v_* \rangle \langle a, A \rangle \langle 1 \rangle$	$\langle u_*, v_* \rangle \langle a, B \rangle \langle 1, 2 \rangle$	$\langle u_*, v_* \rangle \langle b, B \rangle \langle 3, 3 \rangle !$

The terminal histories marked with an exclamation mark are winning histories for \exists . Because $f^\times(1) = \text{hide}$, the game at hand is a genuine game of imperfect information. This fact is reflected in the set of information cells \mathcal{H} , containing the following three non-singletons:

$$\begin{aligned} & \{ \langle u_*, v_* \rangle \langle a \rangle, \langle u_*, v_* \rangle \langle b \rangle \}, \\ & \{ \langle u_*, v_* \rangle \langle a, A \rangle, \langle u_*, v_* \rangle \langle b, A \rangle \}, \\ & \{ \langle u_*, v_* \rangle \langle a, B \rangle, \langle u_*, v_* \rangle \langle b, B \rangle \}. \end{aligned}$$

(Note that under the customary definition of \sim , one would not have the latter two information cells, as they belong to \forall .) Game theorists often find it convenient to present extensive games as trees, as in Figure 6.3. \square

6.3 A perfect information Scotland Yard game

I observed that Scotland Yard is a game with imperfect information and in Definition 6.2.1 I modeled it as an extensive game with imperfect information. This model one may find Scotland Yard's canonical means of analysis, for admittedly, it gives a natural account of the imperfect information that makes Scotland Yard such a fun game to play. Canonical or not, this does not imply, of course, that

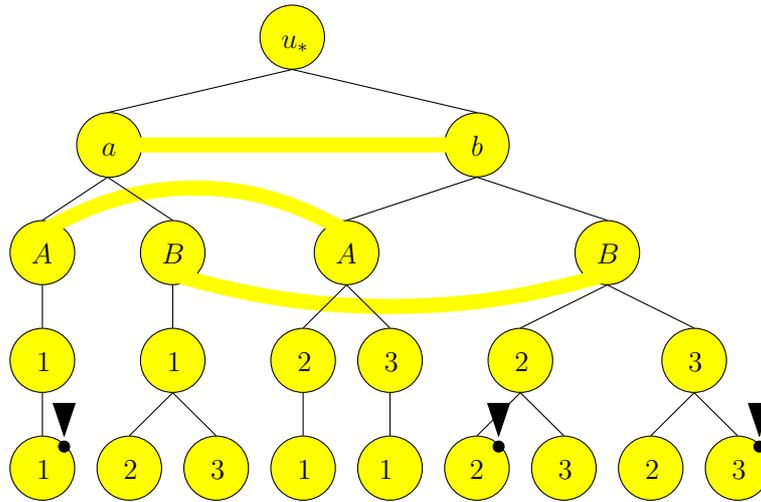


Figure 6.3: A graphical representation of the Scotland Yard game played on the game board constituted by the digraph G^\times from Figure 6.2. A path from the root to any of its nodes represents a history in the game. For instance, the path $u_*, b, A, 2$ corresponds with the history $\langle u_*, v_* \rangle \langle b, A \rangle \langle 2 \rangle$. The information cells are indicated by the shaded areas. The cells marked with an exclamation mark are won by \exists .

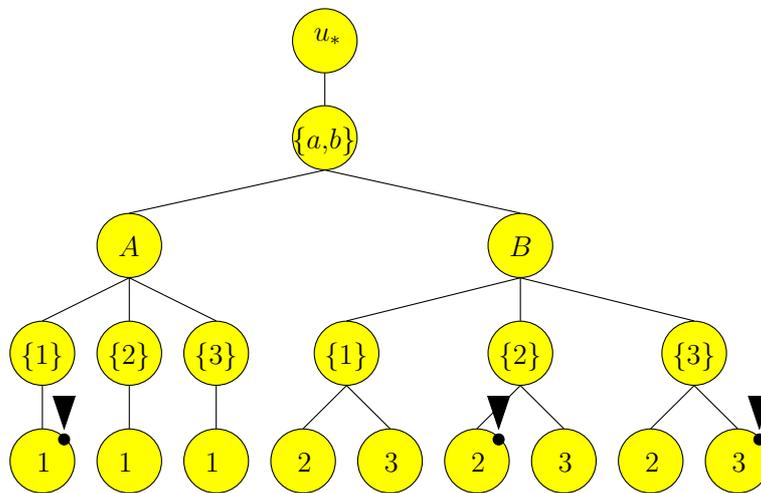


Figure 6.4: A graphical representation of the Scotland Yard-PI game played on the game board constituted by the digraph G^\times from Figure 6.2. A path from the root to any of its nodes represents a history in the game. For instance, the path $u_*, \{a, b\}, A, \{2\}$ corresponds with the history $\langle \{u_*\}, v_* \rangle \langle \{a, b\}, A \rangle \langle \{2\} \rangle$. The cells marked with an exclamation mark are won by \exists .

Scotland Yard can *only* be analyzed as an imperfect information game. In the remainder of this section I will show how a Scotland Yard instance may also give rise to a game of perfect information. The underlying idea has it that during rounds in which \forall moves and hides his whereabouts, he now picks up a *set of vertices* that contains all vertices where he can possibly be, from the viewpoint of \exists . In case \forall has to show himself, he selects one vertex from the current set of vertices and announces this vertex as his new location.

More abstractly, \forall 's powers are lifted from the level of picking up vertices to the level of picking up sets of vertices. \exists 's power remain unaltered, as compared to the game with imperfect information that was explicated above.

Modeling imperfect information by means of a power set construction—as we are about to do—is by no means new. For instance, the reader may find this idea occurring in the computational analyzes of games with imperfect information (Reif 1984; Peterson, Azhar, and Reif 2001). In logic, the idea of evaluating a Independence-friendly logic-sentence with respect to a set of assignments underlies Hodges' trump semantics (Hodges 1997). In automata theory, the move to power sets is made when converting a non-deterministic finite automaton into a deterministic one, see (Hopcroft and Ullman 1979).

As regards every single one of these disciplines, however, observe that the object that was analyzed through power sets is substantially more complex/powerful/bigger than the original object. For instance, in (Peterson, Azhar, and Reif 2001) it was shown that three-player games with imperfect information can be undecidable. In the realm of IF logic it was proven (Cameron and Hodges 2001) that no compositional semantics can be given based on single assignments only. It is well-known that in the worst case converting a non-deterministic finite automaton makes the number of states increase exponentially.

In view of these results it is striking that one can define a Scotland Yard game with perfect information using a power set argument, without experiencing a combinatorial explosion, cf. Theorem 6.5.3. What is meant by “highly similar” is made precise in Section 6.4. First let me postulate the game rules for the Scotland Yard game with perfect information and define its extensive game form in Definition 6.3.1.

Let $sy = \langle G, \langle u_*, \vec{v}_* \rangle, f \rangle$ be a Scotland Yard instance as in Definition 6.1.1. The initial position of the Scotland Yard-PI game constituted by sy is similar to the initial position of the Scotland Yard game that is constituted by sy . That is, a \forall pawn is positioned on u_* and for every $1 \leq j \leq n$, the \exists_j pawn is positioned on $\vec{v}(j)$. In Scotland Yard-PI, \forall does not have one pawn at his disposal but as many as there are vertices in G .

Fix $i = 1$, $U = \{u_*\}$, and $\vec{v} = \vec{v}_*$; round i of Scotland Yard-PI goes as follows:

- 1-PI. If $U - \{\vec{v}\} = \emptyset$, then the game stops and \exists wins. If $U - \{\vec{v}\} \neq \emptyset$ and $i > k$ the game also stops but \exists loses.

- 2-PI. Let $U' = E(U - \{\vec{v}\})$. If $f(i) = \textit{hide}$, then set $U = U'$ and \forall positions a \forall pawn on every vertex v in U . If $f(i) = \textit{show}$, then \forall picks a vertex $u' \in U'$, removes all his pawns from the board, and puts one pawn on u' . Set $U = \{u'\}$.
- 3-PI. Player \exists chooses a vector $\vec{v}' \in V^n$, such that $E(\vec{v}, \vec{v}')$, and for every $1 \leq j \leq n$, moves pawn \exists_j to $\vec{v}'(j)$. Set $\vec{v} = \vec{v}'$.
- 4-PI. Set $i = i + 1$.

Clearly, for arbitrary sy , the Scotland Yard-PI game constituted by sy is a game of perfect information. Thus extensive game theory provides natural means of analysis.

6.3.1. DEFINITION. Let $sy = \langle G, \langle u_*, \vec{v}_* \rangle, f \rangle$ be a Scotland Yard instance. Then, let the *extensive Scotland Yard-PI game constituted by sy* be defined as the tuple $SY\text{-PI}(sy) = \langle N_{\text{PI}}, H_{\text{PI}}, P_{\text{PI}}, U_{\text{PI}} \rangle$, where

- $N_{\text{PI}} = \{\exists, \forall\}$ is the set of players.
- H_{PI} is the set of *histories*, that is, the smallest set containing the strings $\langle \{u_*\} \rangle, \langle \{u_*\}, \vec{v}_* \rangle$, that is closed under taking actions for \exists and \forall :
 - If $h \langle U, \vec{v} \rangle \in H_{\text{PI}}$, $\ell(h \langle U, \vec{v} \rangle) < k$, $f(\ell(h \langle U, \vec{v} \rangle) + 1) = \textit{hide}$, and $U - \{\vec{v}\} \neq \emptyset$, then $h \langle U, \vec{v} \rangle \langle E(U - \{\vec{v}\}) \rangle \in H_{\text{PI}}$.
 - If $h \langle U, \vec{v} \rangle \in H_{\text{PI}}$, $\ell(h \langle U, \vec{v} \rangle) < k$, and $f(\ell(h \langle U, \vec{v} \rangle) + 1) = \textit{show}$, then $\{h \langle U, \vec{v} \rangle \langle \{u'\} \rangle \mid u' \in E(U - \{\vec{v}\})\} \subseteq H_{\text{PI}}$.
 - If $h \langle U, \vec{v} \rangle \langle U' \rangle \in H_{\text{PI}}$ and $E(\vec{v}, \vec{v}')$, then $h \langle U, \vec{v} \rangle \langle U', \vec{v}' \rangle \in H_{\text{PI}}$.

Let \succ_{PI} be the *immediate successor* relation on H_{PI} . That is, the smallest relation closed under the following conditions:

- If $h, h \langle U \rangle \in H_{\text{PI}}$, then $h \succ_{\text{PI}} h \langle U \rangle$.
- If $h \langle U \rangle, h \langle U, \vec{v} \rangle \in H_{\text{PI}}$, then $h \langle U \rangle \succ_{\text{PI}} h \langle U, \vec{v} \rangle$.

A history that has no immediate successor is called a *terminal history*. Let $Z_{\text{PI}} \subseteq H_{\text{PI}}$ be the set of terminal histories in H_{PI} .

- $P_{\text{PI}} : H_{\text{PI}} - Z_{\text{PI}} \rightarrow \{\exists, \forall\}$ is the player function that decides who is to move in a non-terminal history. Due to the notational convention, the value of P is determined by the history's form, in the sense that $P(h \langle U \rangle) = \exists$ and $P(h \langle U, \vec{v} \rangle) = \forall$, no matter h, U , and \vec{v} .

- $U_{\text{PI}} : Z_{\text{PI}} \rightarrow \{\text{win}, \text{lose}\}$ is the function that decides whether a terminal history $h\langle U, \vec{v} \rangle$ is won or lost for \exists . Formally,

$$U(h\langle U, \vec{v} \rangle) = \begin{cases} \text{win} & \text{if } U - \{\vec{v}\} = \emptyset \\ \text{lose} & \text{if } U - \{\vec{v}\} \neq \emptyset. \end{cases}$$

These definitions may be appreciated best by checking $SY\text{-PI}(sy^\times)$, where $sy^\times = \langle G^\times, \langle u_*, \vec{v}_* \rangle, f^\times \rangle$ and G^\times is the digraph depicted in Figure 6.2. I skip writing down all histories in this particular game, leaving the reader with a graphical representation of its game tree in Figure 6.4.

6.4 An effective equivalence

In this section, the similarity between Scotland Yard and its perfect information variant is established. Making use of this similarity, I prove in Theorem 6.4.12 that for any instance sy , \exists has a winning strategy in $SY(sy)$ iff she has one in $SY\text{-PI}(sy)$. In order to prove this result, I go about as follows: Firstly, it will be shown in Lemma 6.4.6 that the structures $\langle \mathcal{H}, \succ \rangle$ and $\langle H_{\text{PI}}, \succ_{\text{PI}} \rangle$ are isomorphic, in virtue of the bijection β . Secondly, I will formally introduce the notion of a winning strategy and the backwards induction algorithms for $SY(sy)$ and $SY\text{-PI}(sy)$. This algorithm typically labels every history with *win* or *lose*, starting from the terminal histories. Crucially, I show that the backwards induction algorithms correctly compute whether \exists has a winning strategy in the respective games. Finally, I will show that for every history h in $SY(sy)$, the label assigned to $C \ni h$ by the backwards induction algorithm for Scotland Yard equals the label assigned to $\beta^{-1}(h)$ by the backwards induction algorithm for Scotland Yard-PI. The claim then follows, as the initial histories $\langle u_*, \vec{v}_* \rangle$ and $\langle \{u_*\}, \vec{v}_* \rangle$ carry the same label.

6.4.1 Scotland Yard and Scotland Yard-PI are isomorphic

Main result of this subsection resides in Lemma 6.4.6, saying that the structures $\langle \mathcal{H}, \succ \rangle$ and $\langle H_{\text{PI}}, \succ_{\text{PI}} \rangle$ are isomorphic. The witness of this isomorphism is the bijection β , shortly defined in Definition 6.4.1. As some of the intermediate results that bring us to the bijection lemma are not very illuminating, I defer them to Appendix A.

The function β is a map from histories in the perfect information game $SY\text{-PI}(sy)$ to information cells in the game $SY(sy)$. An information cell is a set of histories that cannot be distinguished by \exists . The perfect information Scotland Yard game was defined in such a way that \exists 's imperfect information in $SY(sy)$ is propagated to perfect information about sets in $SY\text{-PI}(sy)$. It will be observed through the map β that there is a bijection between information cells—sets of histories—in the imperfect information game, and histories in the perfect

information game that hold sets of vertices owned by \forall . Thus, β is a map from the information cells in $SY(sy)$ to histories in $SY\text{-PI}(sy)$.

6.4.1. DEFINITION. Let $SY(sy)$ and $SY\text{-PI}(sy)$ be games constituted by sy . Define the function $\beta : H_{\text{PI}} \rightarrow \wp(H)$ inductively as follows:

$$\begin{aligned}\beta(\langle\{u_*\}\rangle) &= \{u_*\} \\ \beta(\langle\{u_*\}, \vec{v}_*\rangle) &= \{u_*, \vec{v}_*\} \\ \beta(h\langle U \rangle) &= \{g\langle u \rangle \in H \mid g \in \beta(h), u \in U\} \\ \beta(h\langle U, \vec{v} \rangle) &= \{g\langle u, \vec{v} \rangle \in H \mid g\langle u \rangle \in \beta(h\langle U \rangle)\}.\end{aligned}$$

The function β is (partially) depicted in Figure 6.5 mapping the histories from $SY\text{-PI}(sy^\times)$ to sets of histories from $SY(sy^\times)$.

Proposition 6.4.2 states that if in a history $h \in H_{\text{PI}}$ a pawn (owned by either player) is positioned on a vertex, then also in $\beta(h)$ there exists a history in which this vertex is occupied by a pawn.

6.4.2. PROPOSITION. *For every history $h' \in H_{\text{PI}}$, the following hold:*

- (1) *If $h' = h\langle U \rangle$ and $f(\ell(h\langle U \rangle)) = \text{hide}$, then it is the case that $U = \{u \mid g\langle u \rangle \in H, \text{ for some } g \in \beta(h)\}$.*
- (2) *If $P(h') = \forall$ and $f(\ell(h') + 1) = \text{show}$, then it is the case that $\{u \mid h' \succ h'\langle\{u\}\rangle, \text{ for some } h'\langle\{u\}\rangle \in H_{\text{PI}}\} = \{u \mid g\langle u \rangle \in H, \text{ for some } g \in \beta(h')\}$.*
- (3) *If $h' = h\langle U \rangle \in H_{\text{PI}}$ and $u \in U$, then there exists a history $g \in \beta(h)$ such that $g\langle u \rangle \in H$.*
- (4) *If $h' = h\langle U, \vec{v} \rangle \in H_{\text{PI}}$, then it is the case that $\beta(h\langle U, \vec{v} \rangle) = \{g\langle u, \vec{v} \rangle \mid g\langle u \rangle \in \beta(h\langle U \rangle)\}$.*

Proposition 6.4.3 is the converse of the previous proposition, as it links up histories in H with histories in H_{PI} .

6.4.3. PROPOSITION. *For every $g' \in H$, the following hold:*

- (1) *If $g' = g\langle u \rangle \in H$, then there exists a $h\langle U \rangle \in H_{\text{PI}}$ such that $g \in \beta(h)$ and $u \in U$.*
- (2) *If $g' = g\langle u, \vec{v} \rangle \in H$, then there exists a $h\langle U, \vec{v}' \rangle \in H_{\text{PI}}$ such that $g\langle u \rangle \in \beta(h\langle U \rangle)$ and $\vec{v} = \vec{v}'$.*

For β to be bijection between H_{PI} and \mathcal{H} , it ought to be the case that β has range \mathcal{H} rather than $\wp(H)$. I lay down the following result.

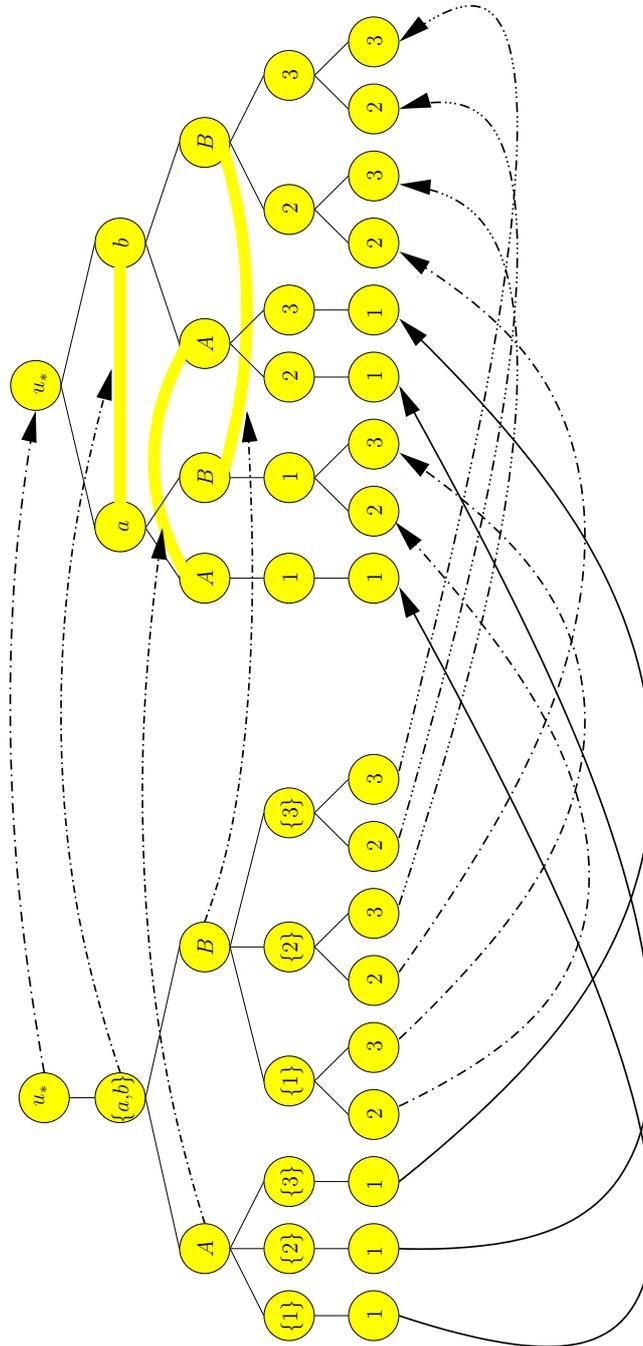


Figure 6.5: A partial depiction of the bijection β from histories in $SY\text{-PI}(sy^\times)$ to sets of histories from $SY(sy^\times)$. β is displayed using several kinds of arrows to enhance readability. Note that this visualization does not reflect any conceptual difference. Sets of histories in the range of β (found in the right-hand structure) turn out to be information cells, cf. Lemma 6.4.5.

6.4.4. LEMMA. β is a function of type $H_{\text{PI}} \rightarrow \mathcal{H}$.

The latter lemma is strengthened in the following lemma.

6.4.5. LEMMA. β is a bijection between H_{PI} and \mathcal{H} .

The isomorphism result follows from tying together the previous statements.

6.4.6. LEMMA. The structures $\langle H_{\text{PI}}, \succ_{\text{PI}} \rangle$ and $\langle \mathcal{H}, \succ \rangle$ are isomorphic.

Proof. Lemma 6.4.5 showed that β is a bijection between H_{PI} and \mathcal{H} . It remains to be shown that β preserves structure, that is, for every pair of histories $h, h' \in H_{\text{PI}}$, it is the case that $h \succ_{\text{PI}} h'$ iff $\beta(h) \succ \beta(h')$. Recall that for $C' \in \mathcal{H}$ to be the immediate successor of $C \in \mathcal{H}$, there must exist two histories g, g' from C, C' , respectively, such that $g \succ g'$. The claim is proved by a straightforward inductive argument on the length of the histories in H_{PI} . I shall omit spelling out the details of the proof, only mentioning the Propositions on which it relies:

From left to right. Follows from Proposition 6.4.2.3 and Proposition 6.4.2.4.

From right to left. Follows from Propositions 6.4.3.1 and 6.4.3.2. \square

Scotland Yard and its perfect information variant are highly similar in the sense that the game trees to which the games give rise are isomorphic.

6.4.2 Backwards induction algorithms

The structures $\langle H_{\text{PI}}, \succ_{\text{PI}} \rangle$ and $\langle \mathcal{H}, \succ \rangle$ are not only isomorphic, they also preserve the property of being winnable for the cops. Traditionally it is backwards induction algorithms that compute whether the cops win, but such algorithms are only defined on games with perfect information. As a consequence, the backwards induction algorithm from Section 2.2 readily applies to the game tree of any perfect information $SY\text{-PI}(sy)$. For future reference, let $B\text{-Ind}_{\text{PI}}(h) \in \{\text{win}, \text{lose}\}$ denote the label of h attached to it by the backwards induction algorithm applied to the game tree of $SY\text{-PI}(sy)$, and let $B\text{-Ind}_{\text{PI}}(sy)$ be the label of the initial history of sy .

Matters are not so straightforward in case of $SY(sy)$. But as I shall show, a backwards induction algorithm can be developed in this case as well. To this end, I rephrase the notion of a winning strategy for games with imperfect information from Section 2.2, and introduce a backwards induction algorithm that will be seen to compute whether a structure allows for a winning strategy, for the Scotland Yard game.

6.4.7. DEFINITION. Let $SY(sy) = \langle N, H, P, \sim, U \rangle$ be a Scotland Yard game constituted by sy . Then, call the structure $\langle \mathcal{S}, \succ' \rangle$ a *plan of action* for \exists in $SY(sy)$, if the following hold:

- $\{\langle u_*, \vec{v}_* \rangle\} \in \mathcal{S}$ and $\mathcal{S} \subseteq \mathcal{H}$.
- $\succ' = \succ \cap (\mathcal{S} \times \mathcal{S})$.
- For every $C \in \mathcal{S}$ such that $P(C) = \exists$, there exists exactly one $C' \in \mathcal{S}$ such that $C \succ' C'$.
- For every $C \in \mathcal{S}$ such that $P(C) = \forall$ and every $C' \in \mathcal{H}$ such that $C \succ C'$, it is the case that $C' \in \mathcal{S}$.

Call $\langle \mathcal{S}, \succ' \rangle$ a *winning plan of action for \exists in $SY(sy)$* , if $\langle \mathcal{S}, \succ' \rangle$ is a plan of action and every terminal cell $C \in \mathcal{S}$ only contains histories h such that $U(h) = \text{win}$.

Before we get to the backwards induction algorithm for Scotland Yard, let me first make sure that the unusual way of defining \exists 's indistinguishability relation does not affect \exists 's having a winning strategy. That is, \exists has a winning strategy in the customary model of the Scotland Yard game of sy iff she has one in $SY(sy) = \langle N, H, P, \sim, U \rangle$.

First let me rephrase Scotland Yard games modeled in the customary way in the current chapter's notation. A *strategy in $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$* is defined, cf. (Osborne and Rubinstein 1994), as a function S mapping every information partition $I \in \mathcal{I}_\exists$ to an action $A(h)$, for some $h \in C$.⁸ Let S be a strategy in $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$ and let $h = \langle u_*, \vec{v}_* \rangle \langle u_1, \vec{v}_1 \rangle \dots \langle u_i, \vec{v}_i \rangle \in H$ be a history, then call h *in accordance with S* , if for every $1 \leq j \leq i$, $S(I_j) = \vec{v}_j$, where I_j is the information partition in \mathcal{I}_\exists containing $\langle u_*, \vec{v}_* \rangle \langle u_1, \vec{v}_1 \rangle \dots \langle u_j \rangle$. A strategy S in $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$ is called *winning for \exists* if every terminal history h in H that is in accordance with S is won for \exists : $U(h) = \text{win}$.

6.4.8. PROPOSITION. *Let sy be a Scotland Yard instance and let $G(sy)$ be the extensive game with imperfect information modeling the Scotland Yard game constituted by sy in the customary way $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$. Then, \exists has a winning plan of action in $SY(sy)$ iff she has a winning strategy in $G(sy)$.*

Proof. We transform every winning strategy S in $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$ into a winning plan of action $\langle \mathcal{S}, \succ' \rangle$ in $SY(sy)$, and vice versa.

From left to right. Suppose S is a winning strategy in the extensive game $G(sy) = \langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$. Let H_S be all histories in H that are in accordance with S . Let \mathcal{H}_S be the partitioning of H_S , such that for any two histories $h, h' \in H_S$, if $h \sim h'$, then there is a cell $D \in \mathcal{H}_S$ containing both h and h' . I claim that $T = \langle \mathcal{H}_S, \succ \cap (\mathcal{H}_S \times \mathcal{H}_S) \rangle$ is a winning plan of action in $SY(sy)$. To this end, it needs proof that (i) $\mathcal{H}_S \subseteq \mathcal{H}$, and that (ii) T is a winning plan of action.

To prove (i), it suffices to show that \mathcal{H}_S is a set of information cells. To this end, it suffices to show that H_S is closed under \sim : if $h \in H_S$ and $h \sim h'$ for some

⁸Recall that for every pair of histories h and h' belonging to \exists , if h and h' sit in the same information partition I , then $A(h) = A(h')$.

$h' \in H$, then $h' \in H_S$. I do so by means of an inductive argument. The base case is trivial. Suppose that $h \in H_S$ and that h' is a history such that $h \sim h'$, where

$$h = h_0 \langle u_0, \vec{v}_0 \rangle \langle u_1, \vec{v}_1 \rangle \quad \text{and} \quad h' = h'_0 \langle u'_0, \vec{v}'_0 \rangle \langle u'_1, \vec{v}'_1 \rangle.$$

The other case is trivial and therefore omitted. It needs to be shown that h' is in accordance with S as well, that is, $h' \in H_S$. Since h is in accordance with S , $S(I) = \vec{v}_1$, where $I \in \mathcal{I}_\exists$ is the information partition holding h . Derive from Proposition 6.2.2.3 that $\vec{v}_1 = \vec{v}'_1$ and that $h_0 \langle u_0, \vec{v}_0 \rangle \langle u_1 \rangle \sim h'_0 \langle u'_0, \vec{v}'_0 \rangle \langle u'_1 \rangle$. Since h is in accordance with S , so is $h_0 \langle u_0, \vec{v}_0 \rangle \langle u_1 \rangle$. Applying the inductive hypothesis yields that $h'_0 \langle u'_0, \vec{v}'_0 \rangle \langle u'_1 \rangle$ is in accordance with S . Hence, $S(I) = \vec{v}_1 = \vec{v}'_1$, implying that h' is in accordance with S . Therefore, $h' \in H_S$.

As for (ii), it needs to be shown that T is closed under taking actions and preserves winning. This follows easily from S 's being a winning strategy.

From right to left. Suppose $\langle \mathcal{S}, \succ' \rangle$ is a winning plan of action in $SY(\text{sy})$. Then, for every $C \in \mathcal{S}$ belonging to \exists there is one C' such that $C \succ' C'$. Essentially similar to Proposition 6.4.2.4 one proves that all histories in C' extend the histories in C by the same vector of vertices \vec{v}_C : $C' = \{h \langle u, \vec{v}_C \rangle \mid h \langle u \rangle \in C\}$, for some $\vec{v}_C \in V^n$. Put $S(C) = \vec{v}_C$ and for every information partition C not present in \mathcal{S} , put $S(C) = \vec{v}$ for an arbitrary vector of vertices \vec{v} that properly extends every history in C . It is readily observed that S is a winning strategy in $\langle N, H, P, \langle \mathcal{I}_i \rangle_{i \in N}, U \rangle$. \square

6.4.9. DEFINITION. Let $SY(\text{sy}) = \langle N, H, P, \sim, U \rangle$ be a Scotland Yard game constituted by sy . The algorithm $B\text{-Ind}$ effectively labels every cell $C \in \mathcal{H}$ with $B\text{-Ind}(C) \in \{\text{win}, \text{lose}\}$ and proceeds as follows:

- Every $h \in Z$ is painted $\text{color}(h) \in \{\text{white}, \text{limegreen}\}$ in such a way that $\text{color}(h) = \text{white}$ iff $U(h) = \text{win}$.
- Every terminal information cell $C \in \mathcal{H}$ is given the label $B\text{-Ind}(C) = \text{win}$ iff $\text{color}(h) = \text{white}$, for every $h \in C$.
- Until every cell has been labelled, apply the following routine to every cell $C \in \mathcal{H}$ that has no label, but all of whose successors have:
 - If $P(C) = \exists$ and there exists a successor C' of C that has been labelled $B\text{-Ind}(C') = \text{win}$, then C gets the label $B\text{-Ind}(C) = \text{win}$; otherwise, C gets the label $B\text{-Ind}(C) = \text{lose}$.
 - If $P(C) = \forall$ and there exists a successor h' of C that has been labelled $B\text{-Ind}(h') = \text{lose}$, then C gets the label $B\text{-Ind}(C) = \text{lose}$; otherwise, C gets the label $B\text{-Ind}(C) = \text{win}$.

Let $C_* = \{tuple_{u_*}, \vec{v}_*\}$ be an information cell, then write $B-Ind(sy)$ to denote $B-Ind(C_*)$.

6.4.10. PROPOSITION. *Let $SY(sy)$ be a Scotland Yard game constituted by sy . Then, \exists has a winning strategy in $SY(sy)$ iff $B-Ind(sy) = win$.*

Proof. Straightforward. □

There is one interesting detail that gets lost in the proof of Proposition 6.4.10, namely the fact that the colorings of the histories with *white* and *limegreen* are only used during the first step of the algorithm in order to label the terminal information cells with *win* and *lose*. However, also non-terminal information cells may contain terminal histories. As the reader can easily verify, the terminal histories in non-terminal information cells are ignored in the previous backward induction algorithm. To see that this goes without affecting the soundness of the algorithm, recall that every history that terminates in round j , for $j < k$, is won by \exists . Therefore, every terminal history that sits in a non-terminal information cell, is won by \exists . The backwards induction algorithm performs a worst-case analysis on the game tree of $SY(sy)$, from \exists 's viewpoint. For this reason terminal histories in non-terminal information are dispensable when it comes to doing the backwards induction analysis.

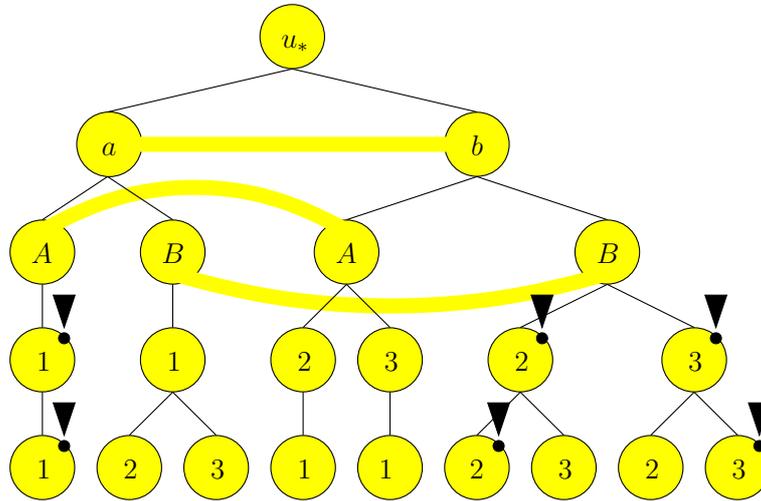
For an example of the two backwards induction algorithms at work, see Figure 6.6. The upcoming lemma states that β is a bijection respecting the labels of the respective histories.

6.4.11. LEMMA. *For every $h \in H_{PI}$, $B-Ind_{PI}(h) = B-Ind(\beta(h))$.*

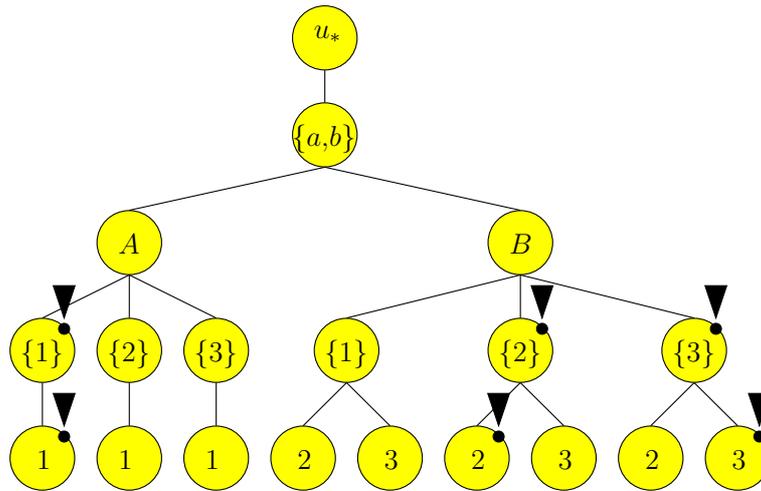
Proof. I prove by induction on the histories $h \in H_{PI}$. The most interesting case is the base step, concerning the terminal information cells.

Suppose $h\langle U, \vec{v} \rangle \in Z_{PI}$. Suppose $B-Ind_{PI}(h\langle U, \vec{v} \rangle) = win$. By definition of $B-Ind_{PI}$ applied to terminal histories it follows that $(U - \{\vec{v}\}) = \emptyset$. Put differently every $u \in U$ is an element of $\{\vec{v}\}$. Now consider an arbitrary history $g\langle u', \vec{v}' \rangle$ from $\beta(h\langle U, \vec{v} \rangle)$. By definition of β , it follows that $u' \in U$ and that $\vec{v}' = \vec{v}$. But then, $u' \in \{\vec{v}\}$ and consequently $U(g\langle u', \vec{v}' \rangle) = win$. Therefore the backwards induction for Scotland Yard paints $g\langle u', \vec{v}' \rangle$ with the color *white*. Since $g\langle u', \vec{v}' \rangle$ was chosen arbitrarily, conclude that every history in $\beta(h\langle U, \vec{v} \rangle)$ is painted *white*, whence $B-Ind(\beta(h\langle U, \vec{v} \rangle)) = win$.

Conversely, suppose $B-Ind_{PI}(h\langle U, \vec{v} \rangle) = lose$. By definition of $B-Ind_{PI}$ applied to terminal histories it follows that $(U - \{\vec{v}\})$ contains at least one object, call it u . By Proposition 6.4.2.3 derive that there exists a history $g\langle u \rangle \in \beta(h\langle U \rangle)$. From Proposition 6.4.2.4 it follows that $g\langle u, \vec{v} \rangle$ is a successor of $g\langle u \rangle$, since $h\langle U, \vec{v} \rangle$ is a successor of $h\langle U \rangle$. Furthermore, $g\langle u, \vec{v} \rangle$ is an element of $\beta(h\langle U, \vec{v} \rangle)$. Since u does not sit in $\{\vec{v}\}$ the history $g\langle u, \vec{v} \rangle$ is painted *limegreen*, by the backwards induction



(a)



(b)

Figure 6.6: The histories in $SY(sy^\times)$ and $SY\text{-PI}(sy^\times)$ labelled by the backwards induction algorithm $B\text{-Ind}$ and $B\text{-Ind}_{PI}$ depicted in (a) and (b), respectively. The information cells marked with an exclamation mark are labelled *win* by the respective backwards induction algorithm.

algorithm of Scotland Yard. Since one of its elements is painted *limegreen*, it is the case that $B\text{-Ind}(\beta(h\langle U, \vec{v} \rangle)) = \textit{lose}$.

Suppose $h\langle U \rangle$ is non-terminal. Suppose $B\text{-Ind}_{\text{PI}}(h\langle U \rangle) = \textit{win}$. Therefore, $h\langle U \rangle$ has a successor $h\langle U, \vec{v} \rangle$, such that $B\text{-Ind}_{\text{PI}}(h\langle U, \vec{v} \rangle) = \textit{win}$. Applying the inductive hypothesis to $h\langle U, \vec{v} \rangle$ yields that $B\text{-Ind}(\beta(h\langle U, \vec{v} \rangle)) = \textit{win}$. Lemma 6.4.6 established that β is a order preserving bijection. Hence, $\beta(h\langle U \rangle) \succ \beta(h\langle U, \vec{v} \rangle)$ and therefore $B\text{-Ind}(\beta(h\langle U \rangle)) = \textit{win}$. The converse case runs along the same line.

Suppose $h\langle U, \vec{v} \rangle$ is non-terminal. Analogous to the previous case. \square

Tying together these results brings us to the desired conclusion:

6.4.12. THEOREM. *Let sy be a Scotland Yard instance. Then, \exists has a winning strategy in $SY(sy)$ iff she has a winning strategy in $SY\text{-PI}(sy)$.*

Proof. Follows immediately from Propositions 6.4.10 and Lemma 6.4.11, since \exists has a winning strategy in $SY(sy)$ iff $B\text{-Ind}(sy) = \textit{win}$ iff $B\text{-Ind}_{\text{PI}}(sy) = \textit{win}$ iff \exists has a winning strategy in $SY\text{-PI}(sy)$. \square

6.5 Scotland Yard is **PSPACE**-complete

In this section, I define Scotland Yard as a decision problem and prove that it is **PSPACE**-complete, both the perfect and the imperfect information game. Hence, the imperfect information in Scotland Yard does not increase the game's complexity, under the current analysis.

Let **SCOTLAND YARD** be the set of all Scotland Yard instances sy such that \exists has a winning strategy in $SY(sy)$. As a special case let the set of Scotland Yard instances **SCOTLAND YARD \clubsuit** equal

$$\{\langle G, \langle u_*, \vec{v}_* \rangle, f \rangle \in \text{SCOTLAND YARD} \mid f \text{ has range } \{\clubsuit\}\},$$

where $\clubsuit \in \{\textit{show}, \textit{hide}\}$.

SCOTLAND YARD and **SCOTLAND YARD $_{\textit{show}}$** both have **PSPACE**-complete complexity, as I show in this section. From this one may conclude that the imperfect information in Scotland Yard does not have a computational impact. Surprisingly, if \exists cannot see the whereabouts of \forall at any stage of the game, the decidability problem ends up being **NP**-complete. That is, **SCOTLAND YARD $_{\textit{hide}}$** is complete for **NP**. The latter claim is substantiated in Section 6.6.

6.5.1. LEMMA. **SCOTLAND YARD** \in **PSPACE**.

Proof. Required is a **PSPACE** algorithm that for arbitrary Scotland Yard instances sy decides whether \exists has a winning strategy in $SY(sy)$. By Theorem 6.4.12, it is sufficient to provide a **PSPACE** algorithm that decides the same problem with respect to $SY\text{-PI}(sy)$. This equivalence comes in useful, because $SY\text{-PI}(sy)$ is a game of perfect information and can for this reason be dealt with by means of the traditional machinery. In fact, the very same machinery supplied by Papadimitriou cited in Section 6.1 will do. Recall that Papadimitriou, namely, observed that deciding the value of a game with perfect information can be done in **PSPACE** if the following requirements are met:

- the length of any legal sequence of moves is bounded by a polynomial in the size of the input; and
- given a “board position” of the game there is a polynomial-space algorithm which constructs all possible subsequent actions and board positions; or, if there aren’t any, decides whether the board position is a win for either player.

It is easy to see that $SY\text{-PI}(sy)$ meets those conditions. As to the first one, namely, the length of the description of any history is polynomially bounded by the number of rounds k of the game. By assumption $k \leq \|V\| \leq \|sy\|$, whence the description of every history is polynomial in the size of the input. As regards the second condition, if $h\langle U, \vec{v} \rangle$ is a non-terminal history, then its successors are either (depending on f) only $h\langle U, \vec{v} \rangle\langle \{w_1, \dots, w_m\} \rangle$ or all of $h\langle U, \vec{v} \rangle\langle \{w_1\} \rangle, \dots, h\langle U, \vec{v} \rangle\langle \{w_m\} \rangle$, where $E(U - \{\vec{v}\}) = \{w_1, \dots, w_m\}$. Those can clearly be constructed in **PSPACE**.

In the worst case, for an arbitrary history $h\langle U, \vec{v} \rangle\langle U' \rangle$ owned by \exists there are $\|V\|^n$ many vectors \vec{v}' such that $E(\vec{v}, \vec{v}')$, where n is the number of \exists 's pawns on the game board. This number is exponential in the size of the input. Nevertheless, every vector \vec{v}' in $V^n = \{v_1, \dots, v_{\|V\|}\}^n$ can be constructed in polynomial space, simply by writing down the vector $\langle v_1, \dots, v_1 \rangle \in V^n$ that comes first in the lexicographical ordering and successively constructing the remaining vectors that follow it up in the same ordering. \square

Hardness is shown by reduction from QBF. To introduce the problem properly, let me introduce some folklore terminology from propositional logic. A *literal* is a propositional variable or a negated propositional variable. A *clause* is a disjunction of literals. A boolean formula is in *conjunctive normal form* (CNF), if it is a conjunction of clauses. A boolean formula is said to be in *3-CNF*, if it is in CNF and all of its clauses contain exactly three literals. The decision problem QBF has *quantified boolean formulae* $\forall x_1 \exists y_1 \dots \forall x_n \exists y_n \phi$ as instances, in which ϕ is a boolean formula in 3-CNF. QBF questions whether for every truth value for x_1 , there is a truth value for y_1, \dots , such that the boolean formula $\phi(\vec{x}, \vec{y})$ is satisfied by the resulting truth assignment. Put formally, QBF is the set of

quantified boolean formulae ψ such that

$$\psi \in \text{QBF} \quad \text{iff} \quad \{true, false\} \models \psi.$$

6.5.2. LEMMA. SCOTLAND YARD_{show} is **PSPACE**-hard.

Proof. Given a QBF instance $\psi = \forall x_1 \exists y_1 \dots \forall x_n \exists y_n \phi$, where $\phi = C_1 \wedge \dots \wedge C_m$ is a boolean formula in 3-CNF, it suffices to construct in **P** a Scotland Yard instance sy_ψ , such that $\psi \in \text{QBF}$ if and only if $sy_\psi \in \text{SCOTLAND YARD}_{show}$. To this end, let me construct the initial position of the game constituted by sy_ψ . The formal specification of sy_ψ follows directly from these building blocks.

Set $i = 0$. For $i \leq n + 1$, do as follows:

- If $i = 0$, lay down the *opening-gadget*, that is schematically depicted in Figure 6.7.a. Moreover, distribute the pawns from

$$\{\exists_{x_1}, \dots, \exists_{x_n}, \exists_{y_1}, \dots, \exists_{y_n} \exists_d, \forall\}$$

over the vertices of the opening-gadget as indicated in its depiction.

- If $1 \leq i \leq n$, first put the x_i -gadget at the bottom of the already constructed game board. Next, put the y_i -gadget below the justly introduced x_i -gadget. Figures 6.7.b and 6.7.c give a schematic account of the x_i -gadget and y_i -gadget, respectively. (Note that as a result of these actions, every vertex in the board game is connected to at least one other vertex, except for the ones on the top row of the opening-gadget and the ones on the bottom row of the y_i -gadget.)
- If $i = n + 1$, put the *clause-gadget* (see Figure 6.7.d) at the bottom of the already constructed board game. This gadget requires a little tinkering before construction terminates, in order to encode the boolean formula ϕ by adding edges to the clause-gadgets (not present in the depiction), as follows:
 - For every variable $z \in \{\vec{x}, \vec{y}\}$ and clause C in ϕ : If z occurs as a literal in C , then join the vertices named “+ z ” and “ C ” by an edge. If $\neg z$ occurs as a literal in C , then join the vertices named “- z ” and “ C ” by an edge.
- Set $i = i + 1$.

Note that the board game can be considered to consists of *layers*, that are indicated by the horizontal, dotted lines. These layers are numbered $-2, -1, \dots, 4n + 5$, enabling us to refer to these layers when describing strategies. Note that the division in layers is not complete: in between the $4(i-1)+3$ rd and $4(i-1)+4$ th layer of the x_i -gadget are two floating vertices.

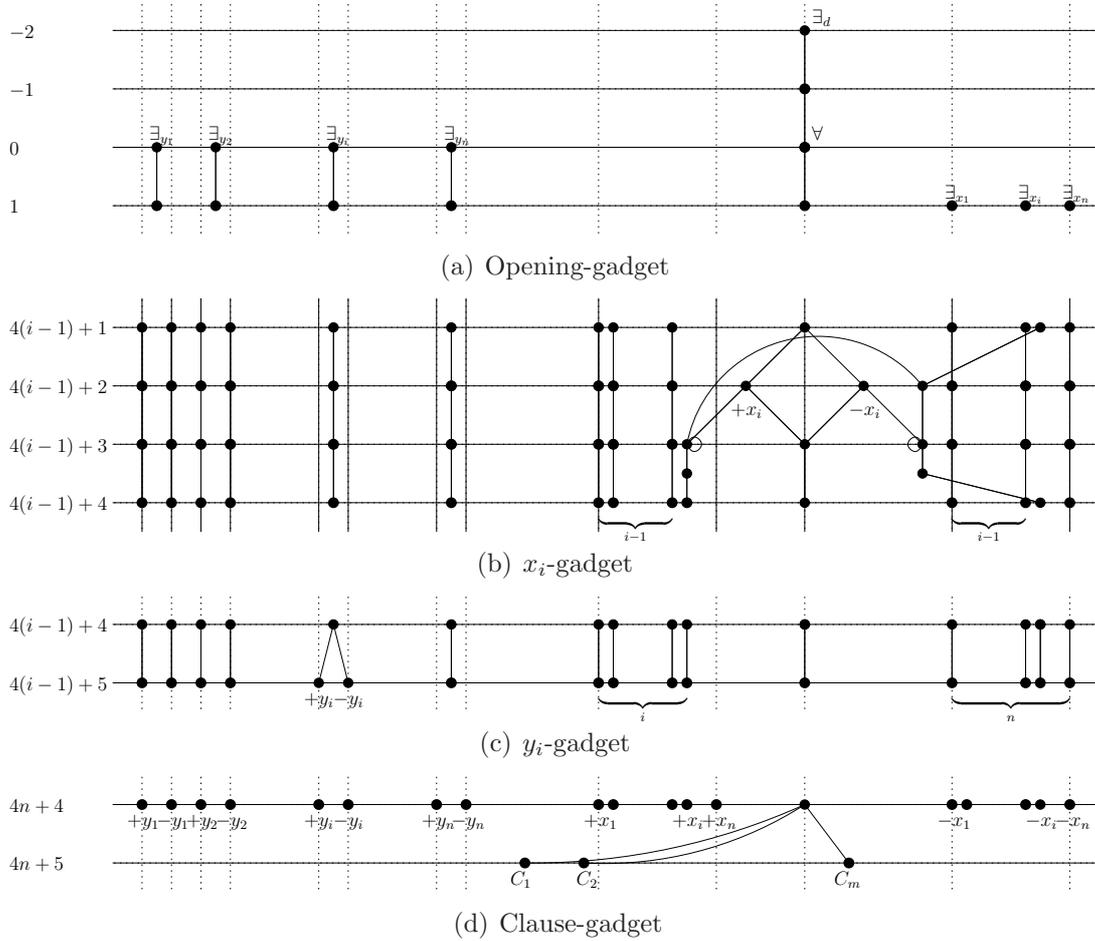


Figure 6.7: The gadgets that make up the initial position of the board game constituted by $SY(sy_\psi)$. The dotted lines are merely “decoration” of the game board, to enhance readability. The horizontal, dotted lines are referred to as “layers”.

The formal specifications of the graph and the initial positions of sy are easily derived from the previous descriptions. Therefore, sy_ψ is fully specified after putting $f : \{1, \dots, 4n + 5\} \rightarrow \{show\}$. Hence, sy_ψ is an instance of $SCOTLAND\ YARD_{show}$.

It remains to be shown that $\psi \in \text{QBF}$ iff $sy_\psi \in \text{SCOTLAND}\ YARD_{show}$.

From left to right. Suppose $\{true, false\} \models \psi$, then there is a way of *subsequently picking* truth values for the existentially quantified variables that renders ψ 's boolean part ϕ true, no matter what truth values are assigned to the universally quantified variables. \exists 's winning strategy in $SY(sy_\psi)$ (being witness of the fact that $sy_\psi \in \text{SCOTLAND}\ YARD_{show}$) shall be read off from the aforementioned

way of picking. I do so by interpreting moves in $SY(sy_\psi)$ as assigning truth values to variables and vice versa: Actions performed by \forall from layer $4(i-1)+1$ to layer $4(i-1)+2$ will be interpreted as assigning a truth value to the universally quantified variable x_i . In particular, a move by \forall to the vertex named “ $+x_i$ ” (“ $-x_i$ ”) will be interpreted as assigning to x_i the value *true* (*false*). Conversely, if \exists 's way of picking prescribes assigning *true* (*false*) to y_i this will be reflected in $SY(sy_\psi)$ by moving \exists_{y_i} to the vertex named “ $+y_i$ ” (“ $-y_i$ ”) on layer $4(i-1)+5$.

Roughly speaking, \exists goes about as follows: when she is to chose between moving \exists_{y_i} to the vertex named “ $+y_i$ ” or “ $-y_i$ ” she understands \forall 's previous moves as a truth assignment and observes which truth value is prescribed by the way of picking. Next, she interprets this truth value as a move in $SY(sy_\psi)$ as described above and moves \exists_{y_i} to the according vertex. This intuition underlies the full specification of \exists 's strategy, described below:

For $0 \leq i \leq n+1$ let \exists 's strategy be as follows:

- Above all: If any pawn can capture \forall , do so!
- For every pawn that stands on a vertex on layer j that is connected to exactly one vertex on layer $j+1$, move it to this vertex. If the pawn at stake is actually \exists_{x_i} standing on a vertex on layer $4(i-1)+3$, it cannot move to the vertex on layer $4(i-1)+4$, because there is a vertex v in between. In this case, move \exists_{x_i} to v and on the next round of the game move it downwards to layer $4(i-1)+4$.
- If \exists_{x_i} stands on a vertex on layer $4(i-1)+2$, then move it to the vertex on layer $4(i-1)+3$ that is connected to the vertex where \forall is positioned.
- If \exists_{y_i} stands on a vertex on layer $4(i-1)+4$, and the way of picking prescribes assigning *true* (*false*) to y_i , then move it to the vertex on layer $4(i-1)+5$ that is named “ $+y_i$ ” (“ $-y_i$ ”).
- If \exists_d stands on a vertex on layer j that has two successors on layer $j+1$, then move it along the left-hand (right-hand) edge, if j is even (odd).
- If \exists_z (for $z \in \{\vec{x}, \vec{y}\}$) stands on a vertex on layer $4n+4$ and this vertex is not connected to a vertex on which \forall is positioned, move it along an arbitrary edge (possibly upwards).

As to \forall 's behavior I claim without rigorous proof that after $4n+4$ rounds of the game (that is, without being captured at an earlier stage of the game) he has traversed a path leading through exactly one of the vertices named “ $+x_i$ ” and “ $-x_i$ ”, for every $x_i \in \{\vec{x}\}$, ending up in a vertex named “ C ”, for some clause C in ϕ . To see that this must be the case: moving \forall upwards at any stage of the game results in an immediate capture by \exists_d . (In fact, \exists_d 's sole purpose in life is

capturing \forall , if he moves upward.) If \forall is moved to one of the reflexive vertices on layer $4(i-1)+3$ he is captured by \exists_{x_i} who moves along the reflexive edge.

Upon arriving at layer $4n+4$, pawn \exists_z (for $z \in \{\vec{x}, \vec{y}\}$) stands on a vertex named “ $+z$ ” or “ $-z$ ”, reflecting that z was assigned *true* or *false*, respectively. By assumption on the successfulness of the way of picking, that guided \exists through $SY(sy_\psi)$, it is the case that the truth assignment that is associated with the positions of the pawns $\exists_{x_1}, \dots, \exists_{x_n}, \exists_{y_1}, \dots, \exists_{y_n}$ makes ϕ true. That is, under that truth assignment, for every clause C in ϕ there is a literal L that is made true. Now, if $L = z$, then \exists_z stands on the vertex named “ $+z$ ” and this vertex and the vertex named “ C ” are joined by an edge; and if $L = \neg z$, then \exists_z stands on the vertex named “ $-z$ ” and this vertex and the vertex named “ C ” are joined by an edge. So no matter to which vertex named “ C ” pawn \forall moves during his $4n+5$ th move, for at least one $z \in \{\vec{x}, \vec{y}\}$ it is the case that \exists_z can move to this vertex named “ C ” and capture him there.

From right to left. Suppose $\{true, false\} \not\models \psi$, then there is a way of picking truth values for the universally quantified variables that renders the boolean part false, no matter what truth values are subsequently assigned to the existentially quantified variables. I leave out the argumentation that this way of picking constitutes a winning strategy for \forall in $SY(sy_\psi)$, as it is similar to the argumentation in the converse direction. But note one crucial property of \forall 's winning strategy: it moves pawn \forall downwards, during every round in the game. Therefore, the only round in which it can be captured is the last one: on a vertex on layer $4n+5$.

Close attention is required, though, to \exists 's behavior. That is, it is to be observed that \exists cannot change her sad destiny (losing) by deviating from the behavior specified in the rules below. The gist of this behavior is that it results in pawn \exists_{x_i} remembering \forall 's moves on layer $4(i-1)+1$ and that after $4n+4$ rounds the pawns $\exists_{x_1}, \dots, \exists_{x_n}, \exists_{y_1}, \dots, \exists_{y_n}$ all stand on a vertex on layer $4n+4$. The point is that just as above, the positions of these pawns on vertices on layer $4n+4$ reflect a truth assignment. This time however, the truth assignment falsifies the boolean part ϕ .

The rules are as follows:

- (1) If \exists_{x_i} stands on a vertex on layer $4(i-1)+2$, then move it to the vertex on layer $4(i-1)+3$ that is connected to the vertex where \forall is positioned.
- (2) If \exists_d stands on a vertex on layer j that is connected to two vertices below, then move it along the left-hand (right-hand) edge, if j is even (odd); or along the right-hand (left-hand) edge, if j is even (odd).
- (3) For every pawn that stands on a vertex on layer j that is connected to exactly one vertex on layer $j+1$, move it to this vertex. (With the same exception as before with regard to \exists_{x_i} standing on a vertex on layer $4(i-1)+3$.)

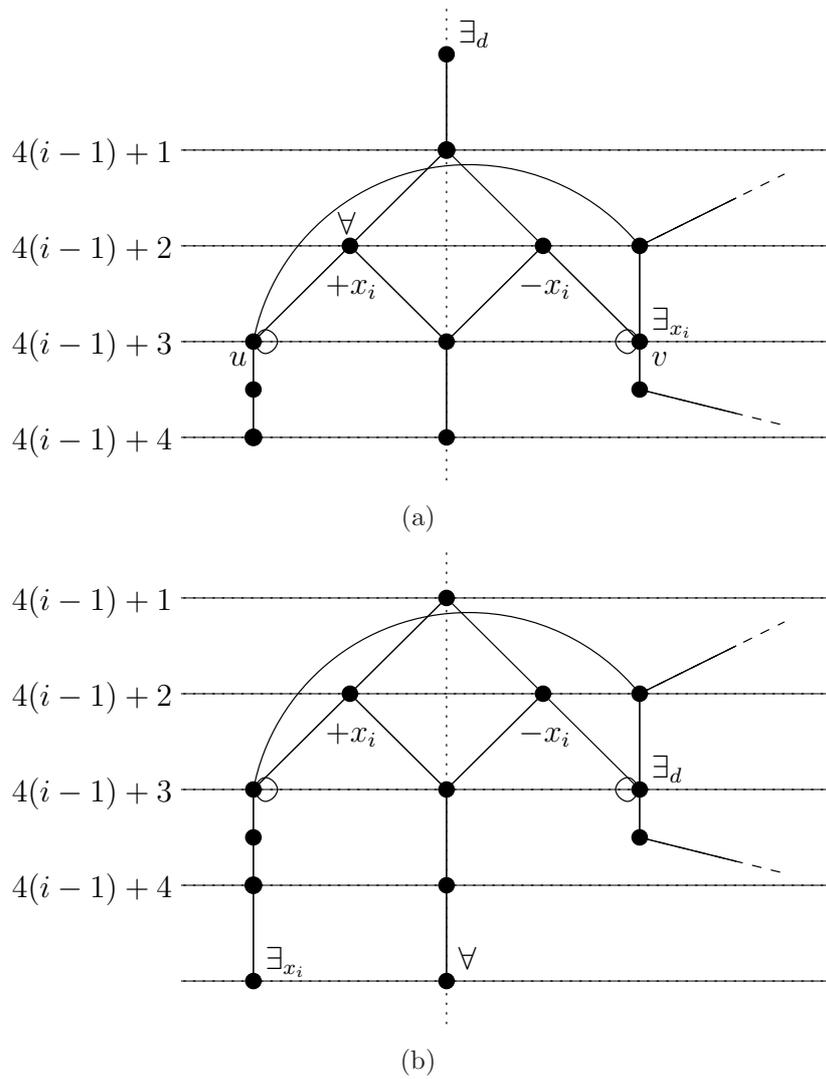


Figure 6.8: Positions on the game board that may occur if \exists does not play according to rule (1) and (2), depicted in (a) and (b), respectively.

I argue that not behaving in correspondence with (1)-(3) will also result in a loss for \exists :

- (1) Suppose \exists_{x_i} stands on the vertex on layer $4(i-1) + 2$, with two options: u and v . Let u be the vertex on layer $4(i-1) + 3$ that is connected to the vertex where \forall is positioned (see Figure 6.8.a). For the sake of the argument let us suppose that \exists_{x_i} is moved to v , violating rule (1). In that case, \forall may safely move to u . If \forall continues the game by moving its pawn downwards it wins automatically, since after the final round (round $4n + 5$) its pawn stands on a vertex on layer $4n + 4$, due to the extra vertex sitting in between layers $4(i-1) + 3$ and $4(i-1) + 4$, without there being any opportunity for \exists to capture him. As such, the pawn \exists_{x_i} is forced to *remember* what vertex \forall visited on layer $4(i-1) + 2$: the one named “ $+x_i$ ” or “ $-x_i$ ”.
- (2) Suppose \exists_d stands on a vertex on layer $4(i-1) + 1$ and from there moves along the right-hand edge twice (see Figure 6.8.b). \forall can exploit this move by moving as he would move normally, except for round $4n + 5$, during which he moves upwards. This behavior results in a guaranteed win for \forall , since none of \exists 's pawns is pursuing \forall closely enough to capture it, after moving upwards.
- (3) Suppose any pawn controlled by \exists moves upwards instead of downwards. This can never result in a win for \exists , because \forall (behaving as he does) can only be captured in the last round of the game, on a vertex on layer $4n + 5$. In particular, any pawn \exists_z , for $z \in \{\vec{x}, \vec{y}\}$, the shortest path to a vertex on layer $4n + 5$ is of length $4n + 5$. Now, if \exists_z is moved upwards, it cannot (during the last round of the game) capture \forall .

This concludes the proof. □

The previous two lemmata are sufficient arguments to settle completeness.

6.5.3. THEOREM. SCOTLAND YARD *and* SCOTLAND YARD_{show} *are* **PSPACE-complete**.

Proof. Lemma 6.5.1 holds that SCOTLAND YARD is solvable in **PSPACE**. To check whether an instance sy has a function f with range $\{show\}$ is trivial, therefore, also SCOTLAND YARD_{show} is solvable in **PSPACE**.

PSPACE-hardness was proven for SCOTLAND YARD_{show} in Lemma 6.5.2. Since the latter problem is a specialization of SCOTLAND YARD, it follows immediately that SCOTLAND YARD is **PSPACE**-hard as well.

Hence, both problems are complete for **PSPACE**. □

6.6 Ignorance is (computational) bliss

Intuitively, adding imperfect information makes a game harder. However, if one restricts oneself to Scotland Yard instances in which \forall 's whereabouts are only known at the beginning of the game, then *deciding* whether \exists has a winning strategy is **NP**-complete, cf. Theorem 6.6.3. After the proof of this theorem, I argue that from a quantitative point of view it is indeed harder for \exists to win an arbitrary Scotland Yard game, thus backing up our pre-computational intuitions.

6.6.1. LEMMA. SCOTLAND YARD_{hide} \in **NP**.

Proof. I make use of the equivalence between the Scotland Yard game and its perfect information counterpart Scotland Yard-PI. It suffices to give an **NP** algorithm that decides whether \exists has a winning strategy in an arbitrary SY -PI(sy), where sy 's information function has range $\{hide\}$. That is, for every integer i on which f is properly defined, we have that $f(i) = hide$. Let me now repeat the game rule from page 154 that regulates \forall 's behavior in the game of Scotland Yard-PI:

2-PI. Let $U' = E(U - \{\vec{v}\})$. If $f(i) = hide$, then set $U = U'$ and \forall positions a \forall pawn on every vertex v iff $v \in U$. If $f(i) = show$, then \forall picks a vertex $u' \in U'$, removes all his pawns from the board, and puts one pawn on u' . Set $U = \{u'\}$.

Since for no $1 \leq i \leq k$, $f(i)$ equals *show*, I can harmlessly replace it by the following rule:

2-PI'. Set $U' = E(U - \{\vec{v}\})$ and \forall positions a \forall pawn on every vertex v iff $v \in U$.

Doing so yields a game in which \forall plays no active role anymore, in the sense that the set U at any round of the game is completely determined by \exists 's past moves. Put differently, any game constituted by an instance of SCOTLAND YARD_{hide} is essentially a one-player game! Having obtained this insight, it is easy to see that the following algorithm decides in non-deterministic polynomial time whether \exists has a winning strategy in the k -round SY -PI(sy):

- Non-deterministically guess a k number of n -dimensional vectors of vertices $\vec{v}_1, \dots, \vec{v}_k \in V^n$.
- Set $U = \{u_*\}$, $\vec{v} = \vec{v}_*$, and $i = 1$; then for $i \leq k$ proceed as follows:
 - If $E(\vec{v}, \vec{v}_i)$, then set $\vec{v} = \vec{v}_i$; else, reject.
 - If $(U - \{\vec{v}\}) = \emptyset$, then accept; else, set $U = (U - \{\vec{v}\})$.
 - Set $i = i + 1$.

- If after k rounds there are still \forall pawns present on the game board, reject.

This algorithm is correct: \exists has a winning strategy in $SY\text{-PI}(sy)$ iff it accepts sy . Hence, $\text{SCOTLAND YARD}_{hide}$ is in **NP**. \square

To prove hardness, I reduce from 3-SAT, that takes boolean formulae ϕ as instance that are in 3-CNF. The boolean formula ϕ is in 3-SAT iff it is satisfiable, that is, there exists a truth assignment of its variables that makes ϕ true. Henceforth, I make the assumption that no clause in a 3-SAT instance contains two copies of one propositional variable. This goes without loss of generality.

6.6.2. LEMMA. $\text{SCOTLAND YARD}_{hide}$ is **NP-hard**.

Proof. To reduce from 3-SAT, let $\phi = C_1 \wedge \dots \wedge C_m$ be an instance of 3-SAT over the variables x_1, \dots, x_n . On the basis of ϕ I will construe a Scotland Yard instance sy_ϕ such that ϕ is satisfiable iff \exists has a winning strategy in $SY\text{-PI}(sy_\phi)$. In fact, sy_ϕ will be read off from the initial game board that is put together as follows.

Set $i = 0$; for $i \leq n$ proceed as follows:

- If $i = 0$, lay down the *clause-gadget* from Figure 6.9.a. The sub-graphs H_j are fully connected graphs with four elements, whose vertices are connected with the vertices w_j , for $1 \leq j \leq m$.
- If $1 \leq i \leq n$, put the *x_i -gadget* to the right of the already constructed game board, see Figure 6.9.b. It will be convenient to refer to the vertex q^i by means of $-^i_{m+1}$ and $+^i_{m+1}$.

For every $0 \leq j \leq m$, do as follows:

- If x_i occurs as a literal in C_j , add the edges $\langle +^i_j, w_j \rangle$ and $\langle w_j, +^i_{j+1} \rangle$.
- If $\neg x_i$ occurs as a literal in C_j , add the edges $\langle -^i_j, w_j \rangle$ and $\langle w_j, -^i_{j+1} \rangle$.
- Add the edges $\langle v_j, -^i_{j+1} \rangle$ and $\langle v_j, +^i_{j+1} \rangle$.

Note that C_0 refers to no clause, and that $-^i_{m+1} = +^i_{m+1} = q^i$.

- Set $i = i + 1$.

The Scotland Yard instance sy_ϕ is derived from the board game: the digraph is completely spelled out and the initial positions are as indicated in the gadgets. Note that the every vertex in the constructed digraph has at least one outgoing edge. In fact the reflexive edges in s^i and t^i serve merely to accomplish this fact. Therefore, sy_ϕ is fully specified after putting $f : \{1, \dots, 2m + 2\} \rightarrow \{hide\}$. Hence, sy_ϕ is an instance of $\text{SCOTLAND YARD}_{hide}$.

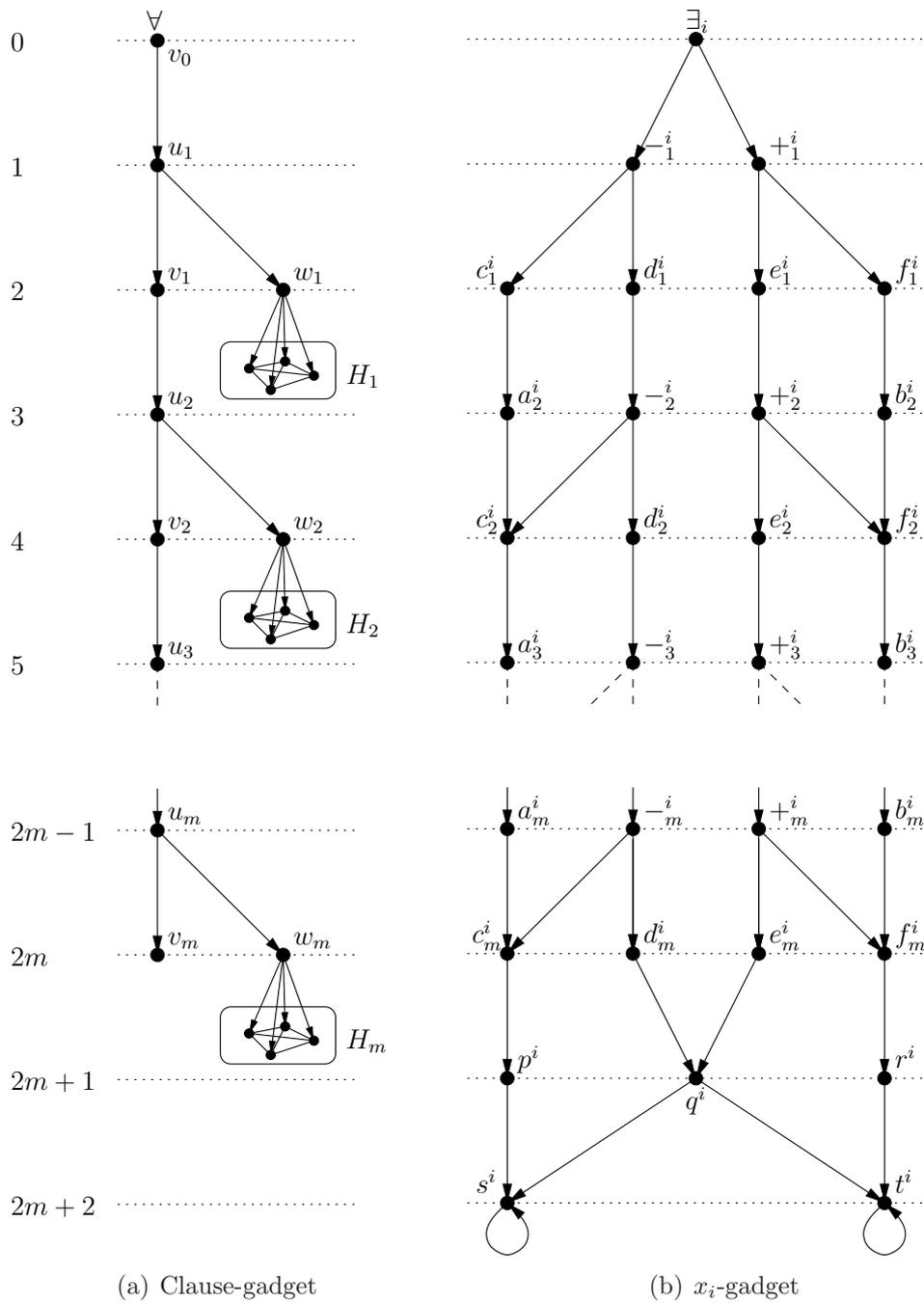


Figure 6.9: The gadgets that make up the initial position of $SY\text{-PI}(sy_\phi)$. The sub-graph H_j is a fully connected graph with 4 elements, all of whose vertices are connected with the vertex w_j .

It remains to be shown that $\phi \in 3\text{-SAT}$ iff $sy_\phi \in \text{SCOTLAND YARD}_{hide}$.

By Theorem 6.4.12, it is sufficient to show that $\phi \in 3\text{-SAT}$ iff \exists has a winning strategy in $SY\text{-PI}(sy_\phi)$.

From left to right. Suppose ϕ is satisfiable, then there exists a truth assignment $t : \{\vec{x}\} \rightarrow \{true, false\}$ such that for every clause C_j in ϕ , there exists at least one literal that is true under t . Let us describe a strategy for \exists that is based on t and argue that it is in fact a winning strategy for her in $SY\text{-PI}(sy_\phi)$:

- If \exists_i stands on the vertex on layer 0 and $t(x_i) = true$ (*false*), then move it to $+_1^i$ ($-_1^i$) on layer 1.
- If \exists_i stands on $-_j^i$ ($+_j^i$), and $-_j^i$ ($+_j^i$) happens to be connected to w_j , then move it to w_j . If \exists_i stands on $-_j^i$ ($+_j^i$) and $-_j^i$ ($+_j^i$) is not connected to w_j , then move it to d_j^i (e_j^i).
- If \exists_i stands on w_j , move it to \pm_{j+1}^i , for $\pm \in \{+, -\}$. Note that this move is deterministic, since there is an edge from w_j to \pm_{j+1}^i , say, only if x_i occurs as a literal in C_j . By assumption of ϕ being an instance of 3-SAT, it cannot be the case that also $\neg x_i$ occurs as a literal in C_j . Hence, there is no edge from w_j to \mp_{j+1}^i .
- If \exists_i stands on d_j^i (e_j^i) then move it to $-_{j+1}^i$ ($+_{j+1}^i$). If \exists_i stands on d_m^i or e_m^i then move it to q^i .
- If \exists_i stands on q^i , then move it to s^i if $t(x_i) = true$ and to t^i if $t(x_i) = false$.

Observe that if \exists plays according to the above strategy, every pawn \exists_i will eventually traverse either all vertices $-_1^i, \dots, -_m^i$ or all vertices $+_1^i, \dots, +_m^i$, given that $t(x_i) = false$ or $t(x_i) = true$, respectively.

To show that this strategy is indeed winning against any of \forall 's strategies, consider the sets of vertices U_j^i that \forall occupies on the clause-gadget and the x_i -gadget, after round $0 \leq j \leq 2m + 2$ of the game in which \exists moved as described above. Initially, \forall has one pawn on v_0 ; thus, $U_0^i = \{v_0\}$. Let us suppose without loss of generality that $t(x_i) = true$. Then, $U_1^i = \{u_1, -_1^i\}$ as the \forall pawn put on $+_1^i$ is captured by \exists_i . I leave it to the reader to check that for $1 \leq j \leq m - 1$, it is the case that

$$\begin{aligned} U_{2j}^i &= \{v_j, c_j^i, d_j^i\} \\ U_{2j+1}^i &= \{u_{j+1}, a_{j+1}^i, -_{j+1}^i\}. \end{aligned}$$

The crucial insight being that the \forall pawn put on w_j can be captured iff there exists at least one literal in C_j that is made true by t . Since t was assumed to be a satisfying assignment, there must be at least one \exists pawn that captures the universal pawn on w_j . It is prescribed by the above strategy that \exists_i is moved to

any w_j -vertex, if possible. Furthermore, it is required to return to the x_i -gadget on the next round of the game, capturing the \forall pawn that was positioned on $+_{j+1}^i$, from v_j .

After round $2m - 1$, \forall cannot continue walking on the *safe path* v_0, u_1, \dots, v_m ; indeed, he has one pawn on v_m and two pawns per x_i -gadget: $U_{2m}^i = \{c_m^i, d_m^i\}$. The pawn put on q^i from v_m is captured by \exists_i coming from e_m^i , so we get that $U_{2m+1}^i = \{p^i\}$. Following the strategy above, \exists moves \exists_i from q^i to s^i , whence $U_{2m+2}^i = \emptyset$. Since i was chosen arbitrarily, it is the case that \forall has no pawns left on any x_i -gadget and therefore has lost after exactly $2m + 2$ rounds of playing.

From right to left. Suppose ϕ is not satisfiable, then for every truth assignment t to the variables in ϕ , there exists a clause C_j in ϕ , that is made false. In the converse direction of this proof, I concluded that every \exists_i traverses one of the paths $-_1^i, \dots, -_m^i, q^i$ and $+_1^i, \dots, +_m^i, q^i$, depending on $t(x_i)$. This behavior I call *in accordance with the truth value $t(x_i)$ assigned to x_i* ; if this behavior is displayed with respect to every $1 \leq i \leq n$, then I say that it is *in accordance with the truth assignment t* .

For now, assume that \exists plays in accordance with some truth assignment t . Since ϕ is not satisfiable, it is not satisfied by t either. Therefore, there is a clause C_j that is not satisfied by t . This is reflected during the playing of the game by the fact that after round $2j$ there is a \forall pawn positioned on w_j that cannot be captured by any \exists_i . This state of affairs will result in a win for \forall , as he positions pawns on every vertex in H_j during round $2j + 1$. By construction, H_j is a connected graph on which he can keep on putting pawns indefinitely.

Remains to be shown that \exists cannot avoid losing by deviating from playing in accordance with some truth assignment. I make the following claims: (A) If after round $1 \leq 2j - 1 \leq 2m + 1$ there is an i such that no \exists pawn is positioned on $-_j^i$ or $+_j^i$, then \exists loses. (B) If after round $2 \leq 2j \leq 2m$ there is an \exists pawn positioned on c_j^i or f_j^i , then \exists loses. I prove by induction. While proving these claims, I take the easily derived fact for granted that during round $2j - 1$ of the game \forall has a pawn on u_j and that during round $2j$ of the game \forall has a pawn on v_j .

Base step. (A) Suppose after round $2m + 1$ no \exists pawn is on q^i (recall that $-_{m+1}^i = +_{m+1}^i = q^i$). Then, there is a \forall pawn on q^i , since by construction of the game board, v_m is connected to q^i . During the next round, \forall has pawns on both s^i and t^i , none of which is captured by \exists , as she has no pawns on the x_i -gadget.

(B) Suppose after round $2m$ there is an \exists pawn positioned on c_m^i , say. We make a case distinction regarding the state of affairs after round $2m + 1$: (i) there is an \exists pawn on q^i . Obviously, this pawn cannot be the one on c_m^i after round $2m$, since there is no edge from c_m^i to q^i . Therefore there are two of \exists 's pawns on the x_i -gadget. As there are exactly n pawns at \exists 's disposal, during round $2m + 1$ there is an x_h -gadget avoid of \exists pawns. In particular, there is no \exists pawn on q^h . Applying clause (A), yields that \exists cannot win from this position. (ii) There is no

\exists pawn on q^i . Then, there is a \forall pawn on q^i after round $2m + 1$, coming from v_m . Therefore, after round $2m + 2$ there is a \forall pawn on t^i ; since \exists can only capture \forall 's pawn at s^i .

Induction step. (A) Suppose after round $1 \leq 2j - 1 < 2m - 1$ there is an i such that no \exists pawn is positioned on $-_j^i$ or $+_j^i$. Since \forall has a pawn on v_{j-1} after round $2j - 2$, he has pawns on both $-_j^i$ and $+_j^i$ after round $2j - 1$. If after the next round \forall occupies the vertices c_j^i, d_j^i, e_j^i or d_j^i, e_j^i, f_j^i this implies that one of \exists 's pawns is on c_j^i or f_j^i , respectively. But then she loses in virtue of the inductive hypothesis of (B). So, suppose that after round $2j$ \forall occupies all the vertices $c_j^i, d_j^i, e_j^i, f_j^i$. Then, for the x_i -gadget to be cleansed of \forall pawns it is prescribed that on some later round of the game there are at least two \exists pawns on this gadget. But then on this round the inductive hypothesis of (A) applies, yielding that \exists loses.

(B) Suppose after round $2 \leq 2j < 2m - 2$ there is an \exists pawn positioned on c_j^i , say. Then, after round $2j + 2$ the same pawn is positioned on c_{j+1}^i . Applying the inductive hypothesis of (B) teaches that \exists loses.

I leave it to the reader to check that if \exists plays in such a way that if during any of the rounds of the game the premises of (A) and (B) do not apply, then she plays in accordance with some truth assignment. However, also playing according to any truth assignment is bound to be a losing way of playing, as I argued earlier. This concludes the proof. \square

Tying together the latter two theorems yields **NP**-completeness for the specialization of Scotland Yard in which \forall does not give any information.

6.6.3. THEOREM. SCOTLAND YARD_{hide} is **NP**-complete.

Proof. Immediate from Lemmata 6.6.1 and 6.6.2. \square

From a computational point of view it is easier to solve the decision problem SCOTLAND YARD, when \forall does not reveal himself during the game. Yet, in a quantitative sense it becomes harder for \exists to play this game, in that there are games in which \exists has no winning strategy if \forall does not reveal himself at all, but she would have had a winning strategy if \forall was to reveal himself at least once. To make this claim precise fix two functions g and h , where

$$g : \{1, \dots, k\} \rightarrow \{hide\} \quad \text{and} \quad h : \{1, \dots, k\} \rightarrow \{hide, show\}$$

such that $h(j) = show$, for some j . I leave it to the reader to check that \exists has a winning strategy in $SY(F, \langle u_*, \vec{v}_* \rangle, h)$ but none in $SY(G, \langle u_*, \vec{v}_* \rangle, g)$. In the latter games, F is the graph depicted in Figure 6.10.

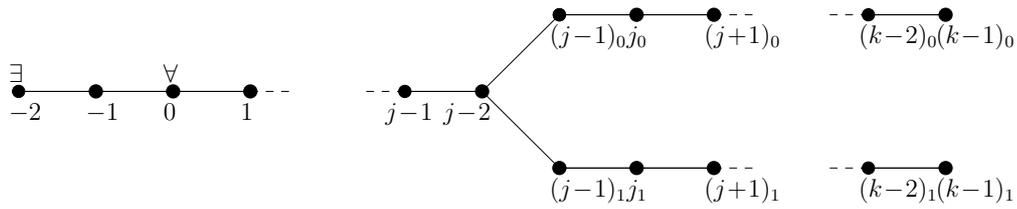


Figure 6.10: The forked graph F . \exists has a winning strategy iff she knows \forall 's position during round j .

6.7 Concluding remarks

By means of a power set construction I observed that imperfect information of vertices can be propagated to perfect information of sets of vertices, without affecting the property of the cops having a winning strategy, cf. Theorem 6.4.12. Lemma 6.5.1 shows that the decision problem $\text{SCOTLAND YARD}_{\text{show}}$ is **PSPACE**-hard; Theorem 6.5.3 shows that that it is **PSPACE**-complete. This finding is in line with the literature on combinatorial game theory, since the former decision problem concerns two-player graph games with perfect information. More surprisingly, it was shown in Lemma 6.5.1 that the power set analysis does not come at a computational cost: also SCOTLAND YARD is solvable in **PSPACE**.

The question why, on an abstract level, the imperfect information in Scotland Yard does not increase the computational complexity is not addressed in this chapter. Thus, a direction for future research is to explore what are Scotland Yard's properties that cause it to behave like most two-player games with perfect information.

I made the point that under the current analysis, Scotland Yard games enjoy the same level of abstraction as graph games. Still the games under consideration form a coherent lot. To name some of their shared properties: the duration is bound by the graph's size; the imperfect information satisfies perfect recall; the information function cannot account for very subtle patterns of ignorance; and, the graphs were only supposed to have out-degree ≥ 1 . Thus a more general theory is desired that charts the computational landscape of imperfect information graph games. In particular the question is worthwhile under what conditions the complexity of graph games does not increase when imperfect information is inserted.

On the whole, the contents of this chapter show that games with imperfect information can be subject to computational analysis just as games with perfect information.

Background. The contents of this dissertation can be understood as a case-based exploration of the logic-games-computation triangle, with a focus on imperfect information structures. Since every topic gives rise to different questions which are of interest to its original discipline, the emphasis in the various sections switches between logic, games, and computation.

In Chapters 3, 4, and 5, I adopted a game-theoretic view on various logical systems, including IF logic, logics with partially ordered quantifier prefixes, and branching quantifiers. In Chapter 3, I studied the expressive power of a game-theoretically motivated fragment of IF logic. This study took place on the nexus of logic and games.

Parts of Chapter 3, 4, and 5 relate to the interface of logic and computation. The modal fragment of IF logic is put forward in the hope that the modal spirit has an ameliorating effect on IF logic's high complexity; the finite model theory for logics with partially ordered connectives aims to find interesting fragments of NP; and the complexity analysis of generalized quantifier expressions is supposed to give the initial impetus to a theory of computational semantics.

The interface of games and computation was addressed directly in the analysis of Scotland Yard in Chapter 6, which can be perceived as the study of a class of abstract graph games with imperfect information. The computational results concerning the logical systems under investigation pertain to the games they were seen to define indirectly, and yet the connection is by no means farfetched.

The questions revisited. Let me now address the two questions that were pursued systematically throughout this dissertation.

Question 1: Which games with imperfect information can be defined by logical means, and which reasonable sources can be seen to cause the imperfect information?

I have sought to find answers to this question along two lines, in Chapters 3, 4, and 5. Firstly, I discovered and reinforced game theory for logical languages. Game-theoretic semantics for IF logic and partially ordered connectives were seen to define extensive games whose imperfect information reflect the partial ordering of the quantifiers. Strategic games were defined in the context of branching quantifiers, and promising directions for future research on their impact for IF logic were outlined.

Secondly, I assessed the question as to what causes the imperfect information in those semantic games. Or, put differently, what kind of information flows can be seen to underly the logical concepts at hand. In Chapter 1, I gave three sources of imperfect information in real-life game situations: attributes, cognitive bounds, and rules. Each one of these was at work in the logical systems studied.

The natural games and the intuitive sources that explain their imperfect information give birth to the hope that more results along these lines can be obtained. Discovering more and more similar results may result in a critical mass of insights, that may lead to revealing deeper and more systematic structures underlying logic and games.

Question 2: What are the computational costs of the imperfect information in logic and combinatorial game theory?

To answer this question various measures of complexity were employed.

I used the measures from descriptive complexity to study languages with partially ordered connectives and generalized quantifier expressions. Thus my results determine the expression complexity of much larger categories of game defining languages than were previously known. I observed that, according to descriptive complexity, the logic **D** floats between first-order logic and **NP**, although from a computational viewpoint **D** is equally hard as **NP**, namely **NP**-complete. In the theory of branching quantifiers I observed that the expression complexity of most natural language determiners as well as composed quantifiers is contained in **L**. By contrast, branching quantifiers can be **NP**-complete, which is a considerable jump in expression complexity.

The notion of satisfiability complexity was used to study the complexity of the newly defined IF modal language. The IF modal language, extended with restricted quantifiers and the equality symbol, was proved undecidable.

These measures of complexity echo different computational tasks in game theory. For instance, the expression complexity corresponds to the complexity of deciding what outcome is predicted by a certain solution concept. I focused on winning strategies, but a wide variety of different solution concepts is offered in the literature on game theory. In a similar vein the satisfiability complexity corresponds to the complexity of mechanism design, that evolves around the question whether there is a pattern of interaction which leads rational agents to end up in a desired state.

Whether imperfect information increases complexity is not answered unequivocally by my results. I am tempted to say that the increase of complexity depends on the richness of the space of information flows. For instance, although the set of actions was restricted in semantic games for \mathbf{D} as compared to the semantic games for \mathbf{H} , the flows of information were essentially left untouched. But then, from a descriptive complexity point of view the logics \mathbf{D} and \mathbf{H} are very similar. It seems that the same phenomenon occurs in the context of the independence-friendly modal logic. The language $\mathbf{IFV}^{\mathbf{ML}}$ restricts the syntax of \mathbf{IF} , but does not in principle affect the variety of independence schemes found in full \mathbf{IF} . Although modal languages are usually computationally well-behaved, the (extended) modal fragment of IF logic is undecidable. In the context of IF logic, the restriction to perfect recall games, which *does* affect the flows of imperfect information, was seen to have dramatic consequences on the expressive power of the logic.

An interesting case of imperfect information not increasing complexity is found in Scotland Yard. With or without imperfect information, the game is equally complex as many other two-player, perfect information games. Indeed, my analysis showed that Scotland Yard can be modeled as a perfect information game. Further research has to point out in what way the space of information flows in Scotland Yard games is delimited, causing the imperfect information not to have any impact on the game's complexity.

Conclusion. This dissertation contributes to the literature that studies the interplay between logic, games, and computation. Especially, it improved our understanding of the role structures with imperfect information play in logic and combinatorial game theory, and what their computational behavior looks like. One of the points of departure of this writing was the observation that the complexity of imperfect information has been studied, but only within very general frameworks. The publications at stake provide some understanding of the impact of imperfect information on complexity, but does not necessarily paint in full detail the picture for imperfect information games that are encountered in practice.

This dissertation investigated particular classes of games within the frameworks of the disciplines in which they are generally studied. For this reason the reported results do reveal what kinds of imperfect information are actually “out there”, and what their complexity is. As such the present dissertation puts forward handles to pursue a systematic theory of the complexity of imperfect information games, which aims to provide a more fine-grained analysis of the complexity of real games with imperfect information.

In my research I benefited from the fact that logic, games, and computation are highly intertwined. Because this allowed me to apply notions from computation to, let's say, logic. But what is more, at various points in this dissertation the

three disciplines appeared to be of a very similar nature. Now, Fagin's Theorem formalizes this sentiment for logic and computation, and enables one to equate computational tasks and logical concepts. In this spirit, an important consequence of this study is provide methods with which one can quantify, qualify, and define the following statement:

computational task = logical concept = strategic interaction.

Appendix A

The boring bits of Chapter 6

6.4.2. PROPOSITION. *For every history $h' \in H_{\text{PI}}$, the following hold:*

- (1) *If $h' = h\langle U \rangle$ and $f(\ell(h\langle U \rangle)) = \text{hide}$, then it is the case that $U = \{u \mid g\langle u \rangle \in H, \text{ for some } g \in \beta(h)\}$.*
- (2) *If $P(h') = \forall$ and $f(\ell(h') + 1) = \text{show}$, then it is the case that $\{u \mid h' \succ h'\langle \{u\} \rangle, \text{ for some } h'\langle \{u\} \rangle \in H_{\text{PI}}\} = \{u \mid g\langle u \rangle \in H, \text{ for some } g \in \beta(h')\}$.*
- (3) *If $h' = h\langle U \rangle \in H_{\text{PI}}$ and $u \in U$, then there exists a history $g \in \beta(h)$ such that $g\langle u \rangle \in H$.*
- (4) *If $h' = h\langle U, \vec{v} \rangle \in H_{\text{PI}}$, then it is the case that $\beta(h\langle U, \vec{v} \rangle) = \{g\langle u, \vec{v} \rangle \mid g\langle u \rangle \in \beta(h\langle U \rangle)\}$.*

Proof. The proof hinges on one big inductive argument on the length of the histories. I warn the reader that the proof of the one item may use the inductive hypothesis of the other item.

The base case in which $\ell(h) = 0$ is trivial and omitted.

- (1) Fix $h\langle U, \vec{v} \rangle\langle U' \rangle \in H_{\text{PI}}$ such that $f(\ell(h\langle U, \vec{v} \rangle\langle U' \rangle)) = \text{hide}$.

From left to right. Suppose $u' \in U'$, then it suffices to show that there exists a history $g\langle u, \vec{v} \rangle\langle u' \rangle \in H$, such that $g\langle u, \vec{v} \rangle \in \beta(h\langle U, \vec{v} \rangle)$. To this end, first observe that $U - \{\vec{v}\} \neq \emptyset$ and that for some $u \in (U - \{\vec{v}\})$ it is the case that $E(u, u')$. Apply the inductive hypothesis of item 3 of this proposition to $h\langle U \rangle$, yielding that there exists a history $g\langle u \rangle \in \beta(h\langle U \rangle)$, since $u \in U$. By the inductive hypothesis of item 4 of this proposition we get that $g\langle u, \vec{v} \rangle \in \beta(h\langle U, \vec{v} \rangle)$. Since $u \in (U - \{\vec{v}\})$, it certainly does not sit in $\{\vec{v}\}$. Hence, $g\langle u, \vec{v} \rangle\langle u' \rangle$ is a history in H , as $E(u, u')$ and $U - \{\vec{v}\} \neq \emptyset$.

From right to left. Suppose $g\langle u, \vec{v} \rangle\langle u' \rangle \in H$, where $g\langle u, \vec{v} \rangle \in \beta(h\langle U, \vec{v} \rangle)$, then it suffices to show that $u' \in U'$. By definition of β it is the case that $g\langle u \rangle \in \beta(h\langle U \rangle)$ and that $u \in U$. Since $g\langle u, \vec{v} \rangle\langle u' \rangle$ is a history, $g\langle u, \vec{v} \rangle$ cannot

be a terminal history, whence $u \notin \{\vec{v}\}$ and $E(u, u')$. Consequently, $u' \in U'$, as required.

- (2) Fix $h\langle U, \vec{v} \rangle \in H_{\text{PI}}$ such that $P(h) = \forall$ and $f(\ell(h\langle U, \vec{v} \rangle) + 1) = \text{show}$.

From left to right. Suppose $h\langle U, \vec{v} \rangle \succ h\langle U, \vec{v} \rangle\langle \{u'\} \rangle$, then it suffices to show that there exists a history $g\langle u, \vec{v} \rangle\langle u' \rangle \in H$, such that $g\langle u, \vec{v} \rangle \in \beta(h\langle U, \vec{v} \rangle)$. To this end, firstly observe that for some $u \in (U - \{\vec{v}\})$ it must be the case that $E(u, u')$. Applying the inductive hypothesis of item 3 of this proposition to $h\langle U \rangle$, yields that there exists a history $g\langle u \rangle \in \beta(h\langle U \rangle)$, since $u \in U$. By the inductive hypothesis of item 4 of this proposition derive that $g\langle u, \vec{v} \rangle \in \beta(h\langle U, \vec{v} \rangle)$. Since $u \in (U - \{\vec{v}\})$, it certainly does not sit in $\{\vec{v}\}$. Hence, $g\langle u, \vec{v} \rangle\langle u' \rangle$ is a history in H , as $E(u, u')$.

From right to left. Suppose $g\langle u, \vec{v} \rangle\langle u' \rangle \in H$, where $g\langle u, \vec{v} \rangle \in \beta(h\langle U, \vec{v} \rangle)$. It suffices to show that $h\langle U, \vec{v} \rangle \succ h\langle U, \vec{v} \rangle\langle \{u'\} \rangle$. By definition of β it is the case that $g\langle u \rangle \in \beta(h\langle U \rangle)$ and that $u \in U$. Since $g\langle u, \vec{v} \rangle\langle u' \rangle$ is a history, $g\langle u, \vec{v} \rangle$ cannot be a terminal history, whence $u \notin \{\vec{v}\}$ and $E(u, u')$. Hence, $h\langle U, \vec{v} \rangle\langle \{u'\} \rangle$ is a history in H_{PI} .

- (3) Follows immediately from items 1 and 2 of this proposition.

- (4) *From left to right.* Follows from the definition.

From right to left. Fix $h\langle U, \vec{v} \rangle\langle U', \vec{v}' \rangle \in H_{\text{PI}}$. It suffices to show that if $g\langle u, \vec{v} \rangle\langle u' \rangle \in \beta(h\langle U, \vec{v} \rangle\langle U' \rangle)$, then $g\langle u, \vec{v} \rangle\langle u', \vec{v}' \rangle$ is a history. By the inductive hypothesis of this proposition, $\beta(h\langle U, \vec{v} \rangle) = \{g\langle u, \vec{v} \rangle \mid g\langle u \rangle \in \beta(h\langle U \rangle)\}$. It is readily observed from the definition of β that for every history $g\langle u, \vec{w} \rangle\langle u' \rangle \in \beta(h\langle U, \vec{v} \rangle\langle U' \rangle)$ it is the case that $\vec{v} = \vec{w}$. By definition of H , it follows that every history $g\langle u, \vec{w} \rangle\langle u' \rangle \in \beta(h\langle U, \vec{v} \rangle\langle U' \rangle)$ has $g\langle u, \vec{w} \rangle\langle u', \vec{v}' \rangle$ as a successor history, since $E(\vec{v}, \vec{v}')$. Hence, the claim follows.

This concludes the proof. □

Proposition 6.4.3 is the converse of Proposition 6.4.2, as it links up histories in H with histories in H_{PI} .

6.4.3. PROPOSITION. *For every $g' \in H$, the following hold:*

- (1) *If $g' = g\langle u \rangle \in H$, then there exists a $h\langle U \rangle \in H_{\text{PI}}$ such that $g \in \beta(h)$ and $u \in U$.*
- (2) *If $g' = g\langle u, \vec{v} \rangle \in H$, then there exists a $h\langle U, \vec{v}' \rangle \in H_{\text{PI}}$ such that $g\langle u \rangle \in \beta(h\langle U \rangle)$ and $\vec{v} = \vec{v}'$.*

Proof. Again, the proof is one big inductive argument on the length of the histories.

The base case in which $\ell(h) = 0$ is trivial and omitted.

- (1) Fix $g\langle u, \vec{v} \rangle \langle u' \rangle \in H$. Clearly, $g\langle u, \vec{v} \rangle$ is no terminal history and therefore $u \notin \{\vec{v}\}$ and $E(u, u')$. By the inductive hypothesis of item 2 of this proposition, it follows that there exists a $h\langle U \rangle \in H_{\text{PI}}$, such that $g\langle u \rangle \in \beta(h\langle U \rangle)$. By definition of β , derive that $u \in U$. Consequently, $U - \{\vec{v}\}$ contains at least one object, namely u . This implies that $h\langle U, \vec{v} \rangle$ is not a terminal history. Since $E(u, u')$, there must exist a history $h\langle U, \vec{v} \rangle \langle U' \rangle$ such that $u' \in U'$.
- (2) Fix $g\langle u, \vec{v} \rangle \langle u', \vec{v}' \rangle \in H$. Clearly, $g\langle u, \vec{v} \rangle$ is no terminal history, whence $u \notin \{\vec{v}\}$ and furthermore $E(\vec{v}, \vec{v}')$. By the inductive hypothesis of item 1 of this proposition, it follows that $g\langle u, \vec{v} \rangle \in \beta(h\langle U, \vec{v} \rangle)$, for some $h\langle U, \vec{v} \rangle \in H_{\text{PI}}$ such that $u \in U$. Since $u \notin \{\vec{v}\}$, $U - \{\vec{v}\}$ is not empty. Consequently, the history $h\langle U, \vec{v} \rangle \langle E(U - \{\vec{v}\}) \rangle$ exists and by definition of β , $g\langle u, \vec{v} \rangle \langle u' \rangle \in \beta(h\langle U, \vec{v} \rangle \langle E(U - \{\vec{v}\}) \rangle)$. Since $E(\vec{v}, \vec{v}')$, it follows that $h\langle U, \vec{v} \rangle \langle E(U - \{\vec{v}\}) \rangle, \vec{v}'$ is a history in H_{PI} as well.

This concludes the proof. \square

6.4.4. LEMMA. β is a function of type $H_{\text{PI}} \rightarrow \mathcal{H}$.

Proof. I prove by induction on the structure of histories $h' \in H_{\text{PI}}$. I omit the base step.

Suppose $h' = h\langle U \rangle$. By definition $\beta(h\langle U \rangle) = \{g\langle u \rangle \in H \mid g \in \beta(h) \text{ and } u \in U\}$. It is easily derived from Proposition 6.4.2.3 that $\beta(h\langle U \rangle)$ is non-empty. By the inductive hypothesis, derive that $\beta(h) = \{g_1, \dots, g_m\} \in \mathcal{H}$, whence $g_1 \sim \dots \sim g_m$. I show that for every $g\langle u \rangle, g'\langle u' \rangle \in \beta(h\langle U \rangle)$, $g\langle u \rangle \sim g'\langle u' \rangle$. To this end I make a case distinction:

Suppose $f(\ell(h\langle U \rangle)) = \text{hide}$. This case follows directly from the definition of \sim , since $g \succ g\langle u \rangle$ and $g' \succ g'\langle u' \rangle$.

Suppose $f(\ell(h\langle U \rangle)) = \text{show}$. Since \forall has to reveal his position, $U = \{v\}$ is a singleton. But then, if $g\langle u \rangle, g'\langle u' \rangle$ are both histories in $\beta(h\langle \{v\} \rangle)$, then $u = u' = v$. Consequently, it follows from the definition of \sim that $g\langle u \rangle \sim g'\langle u' \rangle$.

Remains to be shown that there exists no superset of $\beta(h\langle U \rangle)$ that is closed under \sim as well. For the sake of contradiction, let $g^+\langle u^+ \rangle$ be such that $g^+\langle u^+ \rangle \sim g\langle u \rangle$, for every $g\langle u \rangle \in \beta(h\langle U \rangle)$, but $g^+\langle u^+ \rangle \notin \beta(h\langle U \rangle)$. From the latter I derive that either (A) $g^+ \notin \beta(h)$ or (B) $u^+ \notin U$. For the sake of contradiction assume (A), that is, $g^+ \notin \beta(h)$. Therefore, $g^+ \not\prec g$, for any $g \in \beta(h)$. From Proposition 6.2.2.4 it follows immediately that $g^+\langle u^+ \rangle \not\prec g_1\langle u_1 \rangle$, since $g_1\langle u_1 \rangle \in \beta(h\langle U \rangle)$. This contradicts the assumption and therefore $g^+ \in \beta(h)$. To derive that (B) cannot hold as well, observe that it follows from Proposition 6.2.2.2 that $u^+ = u_1$, since $g^+\langle u^+ \rangle \sim g_1\langle u_1 \rangle$ and $f(\ell(g\langle u \rangle)) = f(\ell(h\langle U \rangle)) = \text{show}$. Since $g_1\langle u_1 \rangle \in \beta(h\langle U \rangle)$, $u^+ = u_1 \in U$. Hence, (B) is not true.

Therefore, $\beta(h\langle U \rangle)$ is a greatest subset of H closed under \sim and as such sits in \mathcal{H} .

Suppose $h' = h\langle U, \vec{v} \rangle$. From the inductive hypothesis it follows that $\beta(h\langle U \rangle) \in \mathcal{H}$. Put $\beta(h\langle U \rangle) = \{g_1\langle u_1 \rangle, \dots, g_m\langle u_m \rangle\}$. It is easily derived from Proposition 6.4.2.3 that $\beta(h\langle U \rangle)$ is non-empty ($m > 0$) and that any two histories from $\beta(h\langle U \rangle)$ are \sim -related. It follows from Proposition 6.4.2.4 that $\beta(h\langle U, \vec{v} \rangle) = \{g_1\langle u_1, \vec{v} \rangle, \dots, g_m\langle u_m, \vec{v} \rangle\}$. Furthermore, it follows directly from the definition of \sim , that any two histories from $\beta(h\langle U, \vec{v} \rangle)$ are \sim -related.

Remains to be shown that there exists no superset of $\beta(h\langle U, \vec{v} \rangle)$ that is also closed under \sim . For the sake of contradiction, let us suppose there exists a history $g^+\langle u^+, \vec{v}^+ \rangle$, such that $g^+\langle u^+, \vec{v}^+ \rangle \sim g\langle u, \vec{v} \rangle$, for every $g\langle u, \vec{v} \rangle \in \beta(h\langle U, \vec{v} \rangle)$, but $g^+\langle u^+, \vec{v}^+ \rangle \notin \beta(h\langle U, \vec{v} \rangle)$. From the latter and the definition of β we derive that either (A) $\vec{v}^+ \neq \vec{v}$ or (B) $g^+\langle u^+ \rangle \notin \beta(h\langle U \rangle)$. But actually both (A) and (B) lead to a contradiction: For both (A) and (B) contradict the assumption that $g^+\langle u^+, \vec{v}^+ \rangle \sim g\langle u, \vec{v} \rangle$, for every $g\langle u, \vec{v} \rangle \in \beta(h\langle U, \vec{v} \rangle)$, in virtue of Propositions 6.2.2.3 and 6.2.2.4, respectively.

Therefore, $\beta(h\langle U, \vec{v} \rangle)$ is a greatest subset of H closed under \sim and as such sits in \mathcal{H} . \square

6.4.5. LEMMA. β is a bijection between H_{PI} and \mathcal{H} .

Proof. It suffices to show that β is surjective and injective.

Surjection. It suffices to prove that for every $C' \in \mathcal{H}$, there exists a history $h \in H_{\text{PI}}$, such that $C' = \beta(h)$. I do so by induction on the structure of the histories in $C' \in \mathcal{H}$.

Suppose $C' = \{g_1\langle u_1, \vec{v}_1 \rangle, \dots, g_m\langle u_m, \vec{v}_m \rangle\}$. Since $C' \in \mathcal{H}$, it is closed under \sim , that is, $g_1\langle u_1, \vec{v}_1 \rangle \sim \dots \sim g_m\langle u_m, \vec{v}_m \rangle$. I derive from Proposition 6.2.2.3 that $\vec{v}_1 = \dots = \vec{v}_m = \vec{v}$ and also that $g_1\langle u_1 \rangle \sim \dots \sim g_m\langle u_m \rangle$. Therefore, there must be one cell $C \in \mathcal{H}$ that contains $g_1\langle u_1 \rangle \sim \dots \sim g_m\langle u_m \rangle$. By the inductive hypothesis, derive that there exists a history $h\langle U \rangle \in H_{\text{PI}}$, such that $\beta(h\langle U \rangle) = C$. By Proposition 6.4.3.2 it is the case that $h\langle U, \vec{v} \rangle \in H_{\text{PI}}$ and by Proposition 6.4.2.4 it is the case that $\beta(h\langle U, \vec{v} \rangle) = \{g\langle u, \vec{v} \rangle \mid g\langle u \rangle \in \beta(h\langle U \rangle) \text{ and } u \in U\} = C'$.

Suppose $C' = \{g_1\langle u_1 \rangle, \dots, g_m\langle u_m \rangle\}$ and $f(\ell(g_1\langle u_1 \rangle)) = \text{show}$. Since $C' \in \mathcal{H}$, it is closed under \sim , that is, $g_1\langle u_1 \rangle \sim \dots \sim g_m\langle u_m \rangle$. I derive from Proposition 6.2.2.2 that $u_1 = \dots = u_m = u$ and also that $g_1 \sim \dots \sim g_m$. Therefore, there must be one cell $C \in \mathcal{H}$ that contains g_1, \dots, g_m . By the inductive hypothesis, derive that there exists a history $h \in H_{\text{PI}}$, such that $\beta(h) = C$.

By Proposition 6.4.3.1 we derive that there is a set U containing u such that $h\langle U \rangle$ is a successor of h , since $g_1\langle u \rangle$ is a successor of g_1 and $g_1 \in \beta(h)$. Since $f(\ell(g_1\langle u_1 \rangle)) = \text{show}$, U must in fact be a singleton, whence $U = \{u\}$. By definition of β it is the case that $\beta(h\langle \{u\} \rangle) = \{g_1\langle u \rangle, \dots, g_m\langle u \rangle\}$, which is simply C' .

Suppose $C' = \{g_1\langle u_1 \rangle, \dots, g_m\langle u_m \rangle\}$ and $f(\ell(g_1\langle u_1 \rangle)) = \text{hide}$. Since $C' \in \mathcal{H}$, it is closed under \sim , that is, $g_1\langle u_1 \rangle \sim \dots \sim g_m\langle u_m \rangle$. I derive from Proposition

6.2.2.1 that $g_1 \sim \dots \sim g_m$. Therefore, there must be one cell $C \in \mathcal{H}$ that contains g_1, \dots, g_m . By the inductive hypothesis, derive that there exists a history $h \in H_{\text{PI}}$, such that $\beta(h) = C$.

From Proposition 6.4.2.1 derive that $h\langle U \rangle$ is a successor of h , where $U = \{u \mid g\langle u \rangle \in H, \text{ for some } g \in \beta(h)\}$. Since $g_1 \in C = \beta(h)$, it follows that $u_1 \in U$. By definition of β it is immediate that $g_1\langle u_1 \rangle \in \beta(h\langle U \rangle)$. In consequence, all of $g_1\langle u_1 \rangle, \dots, g_m\langle u_m \rangle$ sit in $\beta(h\langle U \rangle)$, since they are all \sim -related. Hence, $C' = \beta(h\langle U \rangle)$.

Injection. It needs proof that for any pair of histories $h, h' \in H_{\text{PI}}$, if $h \neq h'$ then $\beta(h) \neq \beta(h')$. I do so by induction on the structure of the histories in H .

Suppose $h\langle U \rangle \neq h'\langle U' \rangle$. I distinguish two cases. (i) $h \neq h'$. h and h' give rise to $\beta(h)$ and $\beta(h')$ which are present in \mathcal{H} , by Lemma 6.4.4. By the inductive hypothesis $\beta(h) \neq \beta(h')$. Since $\beta(h), \beta(h') \in \mathcal{H}$, I conclude that $\beta(h) \cap \beta(h') = \emptyset$. From Proposition 6.4.2.3 it follows that for every $u \in U$ there exists a $g \in \beta(h)$ such that $g\langle u \rangle \in \beta(h\langle U \rangle)$ and that for every $u' \in U'$ there exists a $g' \in \beta(h')$ such that $g'\langle u' \rangle \in \beta(h'\langle U' \rangle)$. Since the intersection of $\beta(h)$ and $\beta(h')$ is empty, it is the case that $g \neq g'$. Hence, $g\langle u \rangle \neq g'\langle u' \rangle$ and therefore $\beta(h\langle u \rangle) \neq \beta(h'\langle u' \rangle)$. (ii) $h = h'$ and $U \neq U'$. Obviously (without loss of generality), there exists a $u \in U$ that does not sit in U' . From Proposition 6.4.2.3 it follows that there exists a $g \in \beta(h)$ such that $g\langle u \rangle \in \beta(h\langle U \rangle)$. By definition of β and the fact that $u \notin U'$, $g\langle u \rangle$ is not an element of $\beta(h'\langle U' \rangle)$ and therefore $\beta(h\langle U \rangle) \neq \beta(h'\langle U' \rangle)$.

Suppose $h\langle U, \vec{v} \rangle \neq h'\langle U', \vec{v}' \rangle$. I distinguish two cases. (i) $h\langle U \rangle \neq h'\langle U' \rangle$. By the inductive hypothesis, it follows that $\beta(h\langle U \rangle) \neq \beta(h'\langle U' \rangle)$. Proposition 6.4.2.4 has it that $\beta(h\langle U, \vec{v} \rangle) = \{g\langle u, \vec{v} \rangle \mid g\langle u \rangle \in \beta(h\langle U \rangle)\}$ and $\beta(h'\langle U', \vec{v}' \rangle) = \{g'\langle u', \vec{v}' \rangle \mid g'\langle u' \rangle \in \beta(h'\langle U' \rangle)\}$. Hence, $\beta(h\langle U, \vec{v} \rangle) \neq \beta(h'\langle U', \vec{v}' \rangle)$. (ii) $h\langle U \rangle = h'\langle U' \rangle$ and $\vec{v} \neq \vec{v}'$. This case follows trivially from Proposition 6.4.2.4. \square

Bibliography

- Ajtai, M. (1983). Σ_1^1 -formulae on finite structures. *Annals of Pure and Applied Logic* 24, 1–48.
- Ajtai, M. (March 17th 2005). Personal communication.
- Ajtai, M. and R. Fagin (1990). Reachability is harder for directed than for undirected graphs. *Journal of Symbolic Logic* 55(1), 113–150.
- Ajtai, M., R. Fagin, and L. Stockmeyer (2000). The closure of monadic NP. *Journal of Computer and System Sciences* 60(3), 660–716.
- Andréka, H., J. F. A. K. van Benthem, and I. Németi (1998). Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic* 27, 217–274.
- Arora, S. and R. Fagin (1997). On winning strategies in Ehrenfeucht–Fraïssé games. *Theoretical Computer Science* 174(1–2), 97–121.
- Baltag, A., L. S. Moss, and S. Solecki (1998). The logic of public announcements, common knowledge and private suspicions. In I. Gilboa (Ed.), *Proceedings of TARK 7*, pp. 43–56. Morgan Kaufmann Publishers.
- Barrington, D. A. M., N. Immerman, and H. Straubing (1988). On uniformity within NC^1 . In *Structure in Complexity Theory: 3rd Annual Conference*, pp. 47–59. Washington: IEEE Computer Society Press.
- Barwise, J. (1979). On branching quantifiers in English. *Journal of Philosophical Logic* 8, 47–80.
- Berlekamp, E. R., J. H. Conway, and R. K. Guy (1982). *Winning ways*. London: Academic Press.
- Biedl, T. C., E. D. Demaine, M. L. Demaine, R. Fleischer, L. Jacobsen, and J. I. Munro (2002). The complexity of Clickomania. In R. J. Nowakowski (Ed.), *More games of no chance*, Volume 42 of *MSRI Publications*, pp. 389–404. Cambridge University Press.

- Billings, D., A. Davidson, J. Schaeffer, and D. Szafron (2002). The challenge of Poker. *Artificial Intelligence Journal* 134(1–2), 201–240.
- Binmore, K. (1996). A note on imperfect recall. In W. Albers, W. Güth, P. Hammerstein, B. Moldovanu, and E. van Damme (Eds.), *Understanding Strategic Interaction—Essays in Honor of Reinhard Selten*, pp. 51–62. New York: Springer-Verlag.
- Binz-Blanke, E. (2006). Game review – Scotland Yard. <http://www.io.com/~beckerdo/games/reviews/ScotlandYardReview.html>.
- Blackburn, P., M. de Rijke, and Y. Venema (2001). *Modal Logic*. Cambridge: Cambridge University Press.
- Blackburn, P. and J. Seligman (1995). Hybrid languages. *Journal of Logic, Language and Information* 4(3), 251–272.
- Blass, A. and Y. Gurevich (1986). Henkin quantifiers and complete problems. *Annals of Pure and Applied Logic* 32(1), 1–16.
- Bonanno, G. (2004). Memory and perfect recall in extensive games. *Games and Economic Behavior* 47(2), 237–256.
- Boolos, G. (1981). For all A there is a B. *Linguistic Inquiry* 12, 465–467.
- Boolos, G. (1993). *The Logic of Provability*. Cambridge: Cambridge University Press.
- Bradfield, J. C. (2000). Independence: logics and concurrency. In P. G. Clote and H. Schwichtenberg (Eds.), *Proceedings of the 14th Workshop on Computer Science Logic (CSL 2000)*, Volume 1862 of *LNCS*. Springer.
- Bradfield, J. C. and S. B. Fröschle (2002). Independence-friendly modal logic and true concurrency. *Nordic Journal of Computing* 9(2), 102–117.
- Buss, S. R. and L. Hay (1991). On truth-table reducibility to SAT. *Information and Computation* 91(1), 86–102.
- Caicedo, X., F. Dechesne, and T. M. V. Janssen (in preparation). Equivalence and quantifier rules for a logic with imperfect information.
- Caicedo, X. and M. Krynicki (1999). Quantifiers for reasoning with imperfect information and Σ_1^1 -logic. In W. A. Carnielli and I. M. Ottaviano (Eds.), *Contemporary Mathematics*, Volume 235, pp. 17–31. American Mathematical Society.
- Cameron, P. J. and W. Hodges (2001). Some combinatorics of imperfect information. *Journal of Symbolic Logic* 66(2), 673–684.
- Chandra, A. K., D. C. Kozen, and L. J. Stockmeyer (1981). Alternation. *Journal of the Association for Computing Machinery* 28, 114–133.

- Cook, S. A. (1971). The complexity of theorem-proving procedures. *Proceedings of the 3rd IEEE Symposium on the Foundations of Computer Science*, 151–158.
- Dawar, A., G. Gottlob, and L. Hella (1998). Capturing relativized complexity classes without order. *Mathematical Logic Quarterly* 44(1), 109–122.
- de Bruin, B. (2004). *Explaining Games, On the Logic of Game Theoretic Explanations*. Ph. D. thesis, ILLC, Universiteit van Amsterdam.
- Dechesne, F. (2005). *Game, Set, Maths: Formal investigations into logic with imperfect information*. Ph. D. thesis, SOBU, Tilburg university and Technische Universiteit Eindhoven.
- Dechesne, F. (2006). Thompson transformations for IF-logic. *Synthese* 149(2), 285–309. Published in the Knowledge, Rationality and Action section.
- Durand, A., C. Lautemann, and T. Schwentick (1998). Subclasses of Binary-NP. *Journal of Logic and Computation* 8(2), 189–207.
- Ebbinghaus, H.-D. and J. Flum (1999). *Finite Model Theory*. Berlin: Springer-Verlag.
- Ehrenfeucht, A. (1961). An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae* 49, 129–141.
- Eiter, T., G. Gottlob, and Y. Gurevich (2000). Existential second-order logic over strings. *Journal of the Association for Computing Machinery* 47(1), 77–131.
- Enderton, H. B. (1970). Finite partially ordered quantifiers. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 16, 393–397.
- Enderton, H. B. (1972). *A mathematical introduction to logic*. Academic press.
- Fagin, R. (1974). Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp (Ed.), *SIAM-AMS Proceedings, Complexity of Computation*, Volume 7, pp. 43–73.
- Fagin, R. (1975). Monadic generalized spectra. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 21, 89–96.
- Fagin, R., J. Y. Halpern, Y. Moses, and M. Y. Vardi (1995). *Reasoning about knowledge*. MIT Press.
- Feferman, S. (2006). What kind of logic is “Independence Friendly” logic. In R. E. Auxier and L. E. Hahn (Eds.), *The philosophy of Jaakko Hintikka*, Library of Living Philosophers, pp. 453–469. Carus Publishing Company.
- Fraenkel, A. S. (2002). Combinatorial games: selected bibliography with a succinct gourmet introduction. In R. J. Nowakowski (Ed.), *More games of no chance*, Volume 42 of *MSRI Publications*, pp. 475–535. Cambridge University Press.

- Fraenkel, A. S., M. R. Garey, D. S. Johnson, T. Schäfer, and Y. Yesha (1978). The complexity of checkers on an $N \times N$ board – preliminary report. In *Proceedings of the 19th Annual Symposium on the foundations of Computer Science*, pp. 55–64. IEEE Computer Society.
- Fraenkel, A. S. and D. Lichtenstein (1981). Computing a perfect strategy for $n \times n$ chess requires time exponential in n . *Journal of Combinatorial Theory Series A* 31, 199–214.
- Fraïssé, R. (1954). Sur quelques classifications des systèmes de relations. *Publications Scientifiques, Série A*, 35–182 1, Université d’Alger.
- Gale, D. and F. Stewart (1953). Infinite games with perfect information. In H. W. Kuhn and A. W. Tucker (Eds.), *Contributions to the Theory of Games II*, Volume 28 of *Annals of Mathematics Studies*, pp. 245–266. Princeton: Princeton University Press.
- Garey, M. R. and D. S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. San Francisco: W. H. Freeman and Company.
- Gil, D. (1982). Quantifier scope, linguistic variation, and natural language semantics. *Linguistics and Philosophy* 5, 421–472.
- Goldstein, A. S. and E. M. Reingold (1995). The complexity of pursuit on a graph. *Theoretical Computer Science* 143, 93–112.
- Goldwasser, S., S. Micali, and C. Rackoff (1989). The knowledge complexity of interactive proof systems. *SIAM Journal on Computing* 18(1), 186–208.
- Gottlob, G. (1997). Relativized logspace and generalized quantifiers over finite ordered structures. *Journal of Symbolic Logic* 62(2), 545–574.
- Gottlob, G. (2004). Second-order logic over finite structures - report on a research programme. In D. Basin and M. Rusinowitch (Eds.), *Proceedings of 2nd International Joint Conference on Automated Reasoning (IJCAR)*, Volume 3097 of *LNAI*, pp. 229–243. Springer.
- Gottlob, G., N. Leone, and H. Veith (1995). Second order logic and the weak exponential hierarchy. In J. Wiedermann and P. Hájek (Eds.), *Proceedings of the 20th International Symposium on Mathematical Foundations of Computer Science*, Volume 969 of *LNCS*, pp. 66–81.
- Harel, D. (1985). Recurring dominoes: making the highly undecidable highly understandable. *Annals of Discrete Mathematics* 24, 51–72.
- Hella, L. and G. Sandu (1995). Partially ordered connectives and finite graphs. In M. Krynicki, M. Mostowski, and L. W. Szczurba (Eds.), *Quantifiers: Logics, Models and Computation*, Volume II of *Synthese library: studies in epistemology, logic, methodology, and philosophy of science*, pp. 79–88. Dordrecht: Kluwer Academic Publishers.

- Hella, L., J. Väänänen, and D. Westerståhl (1997). Definability of polyadic lifts of generalized quantifiers. *Journal of Logic, Language and Information* 6(3), 305–335.
- Henkin, L. (1961). Some remarks on infinitely long formulas. In P. Bernays (Ed.), *Infinitistic Methods. Proceedings of the Symposium on Foundations of Mathematics*, Oxford and Warsaw, pp. 167–183. Pergamon Press and PWN.
- Hintikka, J. (1974). Quantifiers vs. quantification theory. *Linguistic Inquiry* 5, 153–177.
- Hintikka, J. (1993). Independence-friendly logic as a medium of knowledge representation and reasoning about knowledge. In S. O. et al. (Ed.), *Information Modelling and Knowledge Bases III: Foundations, Theory and Applications*, Amsterdam, pp. 258–265. IOS Press.
- Hintikka, J. (1996). *Principles of mathematics revisited*. Cambridge University Press.
- Hintikka, J. (2002a). Hyperclassical logic (a.k.a. IF logic) and its implications for logical theory. *Bulletin of Symbolic Logic* 8(3), 404–423.
- Hintikka, J. (2002b). Quantum logic as a fragment of Independence-friendly logic. *Journal of Philosophical Logic* 31, 197–209.
- Hintikka, J. and G. Sandu (1997). Game-theoretical semantics. In J. F. A. K. van Benthem and A. ter Meulen (Eds.), *Handbook of Logic and Language*, pp. 361–481. Amsterdam: North Holland.
- Hodges, W. (1997). Compositional semantics for a language of imperfect information. *Logic Journal of the IGPL* 5(4), 539–563.
- Hodges, W. (2001). Formal aspects of compositionality. *Journal of Logic, Language and Information* 10(1), 7–28.
- Hodges, W. (March 14th 2006). Games, aims, claims, frames and maybe some names. Presentation delivered at the opening event of the GLoRiClass project.
- Hopcroft, J. E. and J. D. Ullman (1979). *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley.
- Hyttinen, T. and G. Sandu (2000). Henkin quantifiers and the definability of truth. *Journal of Philosophical Logic* 29(5), 507–527.
- Hyttinen, T. and T. Tulenheimo (2005). Decidability of IF modal logic of perfect recall. In R. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing (Eds.), *Advances in Modal Logic*, Volume 5, pp. 111–131. King’s College Publications.

- Immerman, N. (1988). Nondeterministic space is closed under complementation. *SIAM Journal on Computing* 17, 935–938.
- Immerman, N. (1999). *Descriptive Complexity*. Graduate texts in computer science. New York: Springer.
- Janssen, T. M. V. (1997). Compositionality. In J. F. A. K. van Benthem and A. ter Meulen (Eds.), *Handbook of Logic and Language*, pp. 417–474. Amsterdam: North Holland.
- Janssen, T. M. V. (2002). Independent choices and the interpretation of IF logic. *Journal of Logic, Language and Information* 11, 367–387.
- Janssen, T. M. V. and F. Dechesne (to appear). Signalling: a tricky business. In J. F. A. K. van Benthem, G. Heinzmann, M. Rebuschi, and H. Visser (Eds.), *The Age of Alternative Logics: Assessing the Philosophy of Logic and Mathematics Today*, pp. 223–242. Kluwer Academic Publishers.
- Jones, N. D. (1978). Blindfold games are harder than games with perfect information. *Bulletin EATCS* 6, 4–7.
- Joosten, J. J. (2004). *Interpretability formalized*. Ph. D. thesis, Universiteit Utrecht, Utrecht.
- Kaye, R. (2000). Minesweeper is NP-complete. *Mathematical Intelligencer* 22(2), 9–15.
- Keenan, E. L. (1992). Beyond the Frege boundary. *Linguistics and Philosophy* 15(2), 199–221.
- Koller, D. and N. Megiddo (1992). The complexity of two-person zero-sum games in extensive form. *Games and economic behavior* 4, 528–552.
- Kooi, B. P. (2003). *Knowledge, Chance, and Change*. Ph. D. thesis, Rijksuniversiteit Groningen, ILLC Dissertation Series 2003-01, Groningen.
- Krynicky, M. (1993). Hierarchies of finite partially ordered connectives and quantifiers. *Mathematical Logic Quarterly* 39, 287–294.
- Krynicky, M. and M. Mostowski (1995). Henkin quantifiers. In M. Krynicky, M. Mostowski, and L. Szczurba (Eds.), *Quantifiers: Logics, Models and Computation*, Volume I of *Synthese library: studies in epistemology, logic, methodology, and philosophy of science*, pp. 193–262. The Netherlands: Kluwer Academic Publishers.
- Krynicky, M. and J. Väänänen (1989). Henkin and function quantifiers. *Annals of Pure and Applied Logic* 43(3), 273–292.
- Kuhn, H. W. (1950). Extensive games. *Proceedings of the National Academy of Sciences of the United States of America* 36, 570–576.
- Kuhn, H. W. (1953). Extensive games and the problem of information. In H. W. Kuhn and A. W. Tucker (Eds.), *Contributions to the Theory of Games*

- II*, Volume 28 of *Annals of Mathematics Studies*, pp. 193–216. Princeton: Princeton University Press.
- Lichtenstein, D. and M. Sipser (1980). GO is polynomial-space hard. *Journal of the Association for Computing Machinery* 27, 393–401.
- McMillan, C. T., R. Clark, P. Moore, C. Devita, and M. Grossman (2005). Neural basis for generalized quantifier comprehension. *Neuropsychologia* 43, 1729–1737.
- Mostowski, M. and D. Wojtyniak (2004). Computational complexity of some natural language construction. *Annals of Pure and Applied Logic* 127, 219–227.
- Osborne, M. J. and A. Rubinstein (1994). *A Course in Game Theory*. MIT Press.
- Papadimitriou, C. H. (1994). *Computational complexity*. Reading, Massachusetts: Addison-Wesley.
- Parikh, R. and J. Väänänen (2005). Finite information logic. *Annals of Pure and Applied Logic* 134(1), 83–93.
- Pauly, M. (2001). *Logic for Social Software*. Ph. D. thesis, Universiteit van Amsterdam, ILLC Dissertation Series 2001-10, Amsterdam.
- Peterson, G., S. Azhar, and J. H. Reif (2001). Lower bounds for multiplayer noncooperative games of incomplete information. *Computers and Mathematics with Applications* 41, 957–992.
- Piccione, M. and A. Rubinstein (1997). On the interpretation of decision problems with imperfect recall. *Games and Economic Behavior* 20(1), 3–24.
- Pietarinen, A. (1998). Imperfect information in epistemic logic. In I. Kruijff-Korbayová (Ed.), *Proceedings of the 3rd ESSLLI Student Session*, pp. 223–234.
- Pietarinen, A. (2001). Varieties of IFing. In G. Sandu and M. Pauly (Eds.), *Proceedings of the ESSLLI'01 Workshop on Logic and Games*.
- Pratt-Hartmann, I. (2004). Fragments of language. *Journal of Logic, Language and Information* 13(2), 207–223.
- Reif, J. H. (1984). The complexity of two-player games of incomplete information. *Journal of Computer and System Science* 29, 274–301.
- Sandu, G. (1993). On the logic of informational independence and its applications. *Journal of Philosophical Logic* 22(1), 29–60.
- Sandu, G. (1997). The logic of informational independence and finite models. *Logic Journal of the IGPL* 5(1), 79–95.
- Sandu, G. and J. Hintikka (2001). Aspects of compositionality. *Journal of Logic, Language and Information* 10, 49–61.

- Sandu, G. and J. Väänänen (1992). Partially ordered connectives. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 38(4), 361–372.
- Savitch, W. J. (1970). Relationships between nondeterministic and deterministic space complexities. *Journal of Computer and System Sciences* 4(2), 177–192.
- Scha, R. J. H. (1981). Distributive, collective, and cumulative quantification. In J. A. G. Groenendijk, T. M. V. Janssen, and M. B. J. Stokhof (Eds.), *Formal methods in the study of language*, pp. 483–512. CWI: Mathematisch Centrum Amsterdam.
- Schäfer, T. J. (1978). Complexity of some two-person perfect-information games. *Journal of Computer and System Sciences* 16, 185–225.
- Sevenster, M. (2006a). The complexity of Scotland Yard. Technical Report PP-2006-18, ILLC.
- Sevenster, M. (2006b). Henkin quantifiers: logic, games, and computation. *Bulletin of the EATCS* 89(July), 136–155.
- Sevenster, M. and T. Tulenheimo (2006). Partially ordered connectives and Σ_1^1 on finite models. In A. Beckmann, U. Berger, B. Löwe, and J. V. Tucker (Eds.), *Proceedings of the 2nd Computability in Europe Conference (CiE 2006), Logical Approaches to Computational Barriers*, Volume 3988 of LNCS, pp. 515–524. Springer.
- Stewart, I. A. (1993a). Logical characterization of bounded query class I: Logspace oracle machines. *Fundamenta Informaticae* 18, 65–92.
- Stewart, I. A. (1993b). Logical characterization of bounded query class II: Polynomial-time oracle machines. *Fundamenta Informaticae* 18, 93–105.
- Stockmeyer, L. J. and A. R. Meyer (1973). Word problems requiring exponential time. In *Proceedings of the 5th ACM Symposium on the Theory of Computing (STOC 73)*, pp. 1–9.
- Szelepcsényi, R. (1987). The method of forcing for nondeterministic automata. *Bulletin of the EATCS* 33, 96–100.
- ten Cate, B. D. (2005). *Model theory for extended modal languages*. Ph. D. thesis, ILLC, Universiteit van Amsterdam.
- Thompson, F. B. (1952). Equivalence of games in extensive form. Technical Report Research memorandum RM-759, U.S. Air Force Project Rand.
- Tulenheimo, T. (2003). On IF modal logic and its expressive power. In P. Balbiani, N.-Y. Suzuki, F. Wolter, and M. Zakharyashev (Eds.), *Advances in modal logic*, Volume 4, pp. 474–498. King’s College Publications.
- Tulenheimo, T. (2004). *Independence-Friendly Modal Logic*. Ph. D. thesis, University of Helsinki, Finland.

- Tulenheimo, T. and M. Sevenster (2006). On modal logic, IF logic and IF modal logic. In I. Hodkinson and Y. Venema (Eds.), *Advances in Modal Logic*, Volume 6, pp. 481–501. College Publications.
- Turán, G. (1984). On the definability of properties of finite graphs. *Discrete Mathematics* 49, 291–302.
- Turing, A. M. (1936). On computable numbers, with an application to the *Entscheidungsproblem*. *Proceedings of the London Mathematical Society* 42, 230–265.
- Väänänen, J. (2002). On the semantics of informational independence. *Logic Journal of the IGPL* 10(3), 337–350.
- Väänänen, J. (unpublished). Independence friendly logic. Unpublished manuscript. I am referring to the document that was on line available from <http://mat-238.math.helsinki.fi/ift/material.htm> in March 2006.
- van Benthem, J. F. A. K. (1976). *Modal correspondence theory*. Ph. D. thesis, Mathematisch Instituut & Instituut voor Grondlagenonderzoek, Universiteit van Amsterdam.
- van Benthem, J. F. A. K. (1983). Five easy pieces. In A. ter Meulen (Ed.), *Studies in Model-Theoretical Semantics*, pp. 1–17. Dordrecht: Foris.
- van Benthem, J. F. A. K. (1986). *Essays in logical semantics*. The Netherlands: Reidel.
- van Benthem, J. F. A. K. (2001). Games in dynamic-epistemic logic. *Bulletin of Economic Research* 53(4), 219–248.
- van Benthem, J. F. A. K. (2004). Probabilistic features in logic games. In D. Kolak and D. Symons (Eds.), *Quantifiers, Questions and Quantum Physics*, pp. 189–194. Springer.
- van Benthem, J. F. A. K. (2006). The epistemic logic of IF games. In R. E. Auxier and L. E. Hahn (Eds.), *The philosophy of Jaakko Hintikka*, Library of Living Philosophers, pp. 481–513. Carus Publishing Company.
- van Benthem, J. F. A. K. (lecture notes). Logic and games. Unpublished classroom notes. I am referring to the 2001 print.
- van Ditmarsch, H. P. (2000). *Knowledge games*. Ph. D. thesis, Rijksuniversiteit Groningen.
- van Emde Boas, P. (1990). Machine models and simulations. In J. van Leeuwen (Ed.), *Handbook of Theoretical Computer science*, Volume A, pp. 3–66. North Holland.
- van Emde Boas, P. (1996). The convenience of tiling. Ct-96-01, ILLC, University of Amsterdam.

- van Emde Boas, P. (2003). Games, complexity and interaction: The role of games in computer science. In H. Kilov and K. Baclawski (Eds.), *Practical Foundations of Business System Specifications*, pp. 313–327. Kluwer Academic Publishers.
- Vardi, M. Y. (1982). The complexity of relational query languages. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, New York, pp. 137–146. ACM Press.
- von Neumann, J. and O. Morgenstern (1944). *Theory of games and economic behavior*. New York: John Wiley and Sons.
- Wagner, K. W. (1990). Bounded query classes. *SIAM Journal on Computing* 19(5), 833–846.
- Walkoe, W. (1970). Finite partially-ordered quantification. *Journal of Symbolic Logic* 35(4), 535–555.
- Westerståhl, D. (1987). Branching generalized quantifiers and natural language. In P. Gärdenfors (Ed.), *Generalized Quantifiers: Linguistics and Logical Approaches*, pp. 269–298. The Netherlands: D. Reidel.
- Westerståhl, D. (1989). Quantifiers in formal and natural languages. In D. Gabbay and F. Guenther (Eds.), *Handbook of Philosophical Logic: Volume IV: Topics in the Philosophy of Language*, pp. 1–131. Dordrecht: Reidel.
- Westerståhl, D. (1995). Quantifiers in natural language: A survey of some recent work. In M. Krynicki, M. Mostowski, and L. Szczerba (Eds.), *Quantifiers: Logics, Models and Computation*, Volume I of *Synthese Library: Studies in Epistemology, Logic, Methodology, and Philosophy of Science*, pp. 359–408. The Netherlands: Kluwer Academic Publishers.
- Zermelo, E. (1913). Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiel. In E. Hobson and A. Love (Eds.), *Proceedings of the 5th Congress of Mathematicians*, Volume II, pp. 501–504. Cambridge University Press.

- absentmindedness, 12, 79, 82
- action consistency requirement, 12, 36, 149
- action recall, 40, 43
- Ajtai, 8, 75, 97, 110, 115, 121
- assignment (α), 15
- atom, 13
- Azhar, 4, 6, 142, 153

- backward induction, 11, 158
- Barwise, 6, 40, 115–117
- Berlekamp, 1, 7, 141
- Blass, 7, 8, 81, 84, 112, 115, 121, 123
- Boolos, 116
- Bradfield, 53
- branching (Br), 121
- Buss, 110

- Caicedo, 32, 38, 48
- Cameron, 38, 52, 73, 153
- capture, 22
- coalitional semantic game, 41
- complete, 20
- complexity class, 18
- complexity theory, 5
- Conway, 1, 7, 141
- cumulation (Cum), 131

- Dawar, 112
- Dechesne, 31, 33, 34, 36, 43, 48, 49
- Dedekind infinity, 32

- determined, 11, 12
- digraph, 20
- dimension, 76, 82

- Ebbinghaus, 107
- efficiently solvable, 19
- Ehrenfeucht-Fraïssé game
 - for \mathbf{D} , 104
 - for \mathbf{FO} , 102
 - for $\mathbf{M}\Sigma_1^1$, Σ_1^1 , 103
- Eiter, 113
- Enderton, 77
- equivalence
 - for \mathbf{IF} -formulae, 31
 - for \mathbf{SO} -formulae, 17
- expressing a property, 21
- expression complexity
 - of a \mathbf{Q} -expression, 119
 - of a sentence, 22
- extensive game
 - of Scotland Yard, 147
 - of Scotland Yard-PI, 154
 - with imperfect information, 12
 - with perfect information, 10

- Fagin, 1, 75, 103, 108, 110
- Fagin’s Theorem, 22, 37, 75, 132
- first-order correspondence language,
 - 59
- Flum, 107

- free variable
 - in IF logic, 29
 - in second-order logic, 14
- Fröschle, 53
- function quantifier, 79
- Gale-Stewart Theorem, 11, 103
- game-theoretic semantics
 - for first-order logic, 18
 - for IF logic, 36
- games
 - with imperfect information, 2
 - with incomplete information, 2
- Gil, 116
- Gottlob, 6, 108, 112, 113
- graph, 20
- Guarded Fragment, 27
- Gurevich, 7, 8, 81, 84, 112, 113, 121, 123
- Guy, 1, 7, 141
- Hay, 110
- Hella, 112, 132
- Henkin, 6, 75
- Henkin depth (*hd*), 111
- Henkin quantifier, 76
- Hintikka, 1, 3, 6, 26, 28, 33, 37, 40, 53, 115, 116
- Hintikka sentence, 26, 116, 129
- Hintikka's Thesis, 116
- history, 10
 - terminal, 10
- Hodges, 33, 37, 38, 52, 56, 73, 153
- Hyttinen, 53, 54, 56, 60, 73, 112
- IF procedure, 29, 60
- independence scheme, 30, 121
- Independence-friendly logic, 3, 6, 26, 28, 126
- information
 - cell, 149
 - partition, 12
 - set, 12, 149
- interactive proof systems, 4
- interpretation, 15
- iteration (*It*), 130
- Janssen, 33, 37, 38, 48, 50
- Jones, 4, 6
- knowledge memory, 40, 43
- Koller, 4, 6, 142
- Krynicky, 32, 79, 86
- linear order, 21
- literal, 87
- Lorenzen, 1
- matrix formula
 - explicit, 90
 - implicit, 82
- maximally consistent subset, 90
- Megiddo, 4, 6, 142
- model, 57
- model checking complexity, 23
- monotone
 - decreasing, 120
 - increasing, 120
- Mostowski, M., 108, 129
- Nash equilibrium, 122
- negation normal form, 15
- operator, 28
- Papadimitriou, 142, 164
- Pareto optimal, 123
- Parikh, 38
- Partial Information logic, 38
- partial isomorphism, 102
- partially ordered connective, 82
- perfect recall, 40, 43, 56, 177
- Peterson, 4, 6, 142, 153
- Pietarinen, 53
- plan of action, 11, 158
- player function, 10
- pointed model, 58
- pre-strategic evaluation game, 121
- probability distribution, 122

- uniform, 122
- proto-literal, 87
- quantifier
 - branching, 6
 - disjunctive and conjunctive, 14
 - first-order, 14
 - partially ordered, 6
 - restricted, 14
 - second-order, 14
 - universal, 120
- quantifier rank (qr), 101
- reduction, 20
- Reif, 4, 6, 142, 153
- relation variable rank (rvr), 93
- Rescher quantifier (R), 119
- resumption (Res), 131
- Sandu, 3, 6, 7, 26, 28, 33, 37, 40, 53, 81, 82, 85, 86, 112, 116
- satisfaction relation (\models)
 - for H , 77
 - for IF , $IF\forall$, 36
 - for $\mathcal{L}(D)$, D , 83
 - for $ML\forall$, 58
 - for Q -expressions, 119
 - for SO , FO , $FO\forall$, 15
 - for V , 86
- scope
 - binding, 26, 28
 - priority, 26, 28
- Scotland Yard, 8, 141, 143
 - instance, 145
- semantic evaluation game
 - for first-order logic, 17
 - for IF logic, 34
- sentence
 - in first-order logic, 14
 - in IF logic, 30
 - in $\mathcal{L}(D)$, 83
- signaling in IF logic, 33
- Skolem semantics, 31
- Skolemization (Sk), 31
- slash-set, 29
- sober formula, 89
- spy point, 63
- standard translation
 - for D , 87
 - for $ML\forall$, 58
- Stewart normal form, 112
- Stockmeyer, 75, 110, 141
- strategic evaluation game, 122
- strategy
 - in imperfect information game, 12
 - in perfect information game, 11
- structure, 15
- subformula
 - in IF logic, 30
 - in second-order logic, 14
- Subgame semantics, 39
- subordinate, 28
- superordinate, 28
- Team logic, 38
- term, 13
- Thompson transformation, 49
- tractable, 19
- trump semantics, 37, 153
- Tulenheimo, 53–56, 60, 73
- Turing machine, 18
- universe, 15
- Väänänen, 7, 38, 79, 81, 82, 85, 132
- van Benthem, 8, 39, 40, 43, 49, 57, 115, 116, 120, 121, 127–129, 135
- van Emde Boas, 4, 18
- vocabulary (τ), 12
- von Neumann-Morgenstern property, 12, 36
- Walkoe, 77
- Westerståhl, 117, 120, 132
- Wojtyniak, 129

List of symbols

τ	vocabulary, a set of relation symbols including “=”	12
$Token(\tau)$	set of tokens associated with τ	13
\vee, \wedge	restricted quantifier symbols (disjunctive, conjunctive)	13
SO	second-order logic	13
FO, FO\vee	first-order logic (extended with restricted quantifiers)	13
Σ_1^1	existential, second-order logic	14
$Free(\Phi)$	set of free variables in second-order Φ	14
$Sub(\Phi)$	set of subformulae in second-order Φ	14
$\mathfrak{A}, \mathfrak{B}, \mathfrak{S}$	structure	15
$R^{\mathfrak{A}}$	interpretation of R on structure \mathfrak{A}	15
α	assignment	15
$[\alpha.x/a]$	assignment that agrees with α , but assigns a to x	15
$[x^{\mathfrak{A}}]$	assignment that assigns arbitrary object $x^{\mathfrak{A}}$ from \mathfrak{A} to x	15
L < L'	L has weaker expressive power than L' (similar for \leq and $=$)	17
$Sem\text{-}game^{\mathbf{L}}$	extensive semantic game for logic L	17
\bar{L}	the complement of the language L with respect to Σ^*	19
$n\text{-COLORABILITY}$	set of n -colorable graphs	21
FO', FO\vee'	negation n. form and unique quantification fragment of FO, FO\vee	27
C_{Φ}	set of uniquely identified operators in Φ	28
$>_{\Phi}$	superordinateness relation over operators in Φ	28
\succ_{Φ}	binding scope relation over operators in Φ	28
IF, IF\vee	IF logic (extended with restricted quantifiers)	28
IF (ϕ)	set of IF-formulae generated from first-order ϕ	29
$Sk(\Phi)$	Skolemization of IF-formula Φ	31
$AR_i(h)$	action experiences of player i in history h	43
$KM_i(h)$	knowledge experiences of player i in history h	43
IF^{PR}	perfect recall fragment of IF	44
IFML^{PR}	perfect recall modal logic from (Hyttinen and Tulenheimo 2005)	56
$Token(\mu)$	set of tokens associated with set of modalities μ	57
ML\vee	basic modal logic with restricted quantifiers	57

$\langle \mathfrak{M}, w \rangle$	pointed model	58
$ST(\phi)$	standard translation of \mathbf{MLV} -sentence ϕ	58
$\mathfrak{A}_{\mathfrak{M}}$	structure constituted by model \mathfrak{M}	59
$\mathbf{IFV}^{\mathbf{ML}}$	result of applying the IF procedure to $ST(\mathbf{MLV})$	60
TILING	set of finite sets of tiles that can tile $\mathbb{N} \times \mathbb{N}$	62
Φ_T	$\mathbf{IFV}^{\mathbf{ML}}$ -formula that is satisfiable iff T tiles the plane	63
$H_k^n \vec{x}$	Henkin quantifier with height n and width k	76
\mathbf{H}	first-order logic closed under single application of H_{\dots}	77
$\mathcal{L}(\mathbf{D})$	first-order logic closed under application of D_{\dots}	82
$D_k^n \vec{x} \vec{i}$	partially ordered connective with height n and width k	82
\mathbf{D}	first-order logic closed under single application of D_{\dots}	83
$\Sigma_{1,k}^1$	fragment of Σ_1^1 with only k -ary relation variables	83
\mathbf{V}	first-order logic closed under single application of V_{\dots}	86
$L(\Phi)$	set of proto-literals of \mathbf{SO} -formula Φ	87
$L(D_k^n)$	set of proto-literals of partially ordered connective D_k^n	87
$T_L(\gamma)$	L -explication of implicit matrix formula γ	87
$T(\Gamma)$	standard translation of \mathbf{D} -formula Γ	87
$\Sigma_{1,k}^1 \heartsuit$	fragment of $\Sigma_{1,k}^1$ with only universally quantified sober formulae	87
S_L	set of maximally consistent subsets of literals based on L	90
$rvr(\Phi)$	relation variable rank of Φ , defined $\ L(\Phi)\ $	93
$qr(\Gamma)$	quantifier rank of \mathbf{D} -formula Γ	101
$\mathfrak{A} \equiv_m^{\mathbf{L}} \mathfrak{B}$	No \mathbf{L} -formula with quantifier rank $\leq m$ distinguishes \mathfrak{A} and \mathfrak{B}	102
$EF_{\dots}^{\mathbf{L}}(\mathfrak{A}, \mathfrak{B})$	Ehrenfeucht-Fraïssé game for logic \mathbf{L} on \mathfrak{A} and \mathfrak{B}	102
CONNECTED	set of connected graphs	103
$\mathbf{P}_{\parallel}^{\mathbf{NP}}$	class of problems, \mathbf{P} -solvable with \mathbf{NP} oracle in parallel	109
$\mathbf{H}^+, \mathbf{D}^+$	first-order closure of \mathbf{H}, \mathbf{D}	110
Most	type $\langle 1, 1 \rangle$ generalized quantifier	119
$Supp(\delta)$	support of probability distribution δ , defined $\{a \mid \delta(a) > 0\}$	122
<i>Str-game</i>	strategic evaluation game for branching quantifier expressions	122
$\mathbf{FO}(+)$	first-order additive logic	128
$SY(sy)$	extensive Scotland Yard game constituted by sy	147
\succ	immediate successor relation over histories in Scotland yard game	148
\sim	indistinguishability relation over histories in Scotland yard game	148
\mathcal{H}	set of information cells in Scotland Yard game	149
β	isomorphism between $SY\text{-PI}(sy)$ and $SY(sy)$	156
$SY\text{-PI}(sy)$	extensive PI-Scotland Yard game constituted by sy	154
SCOTLAND YARD	decision problem of Scotland Yard	163
QBF	set of quantified boolean formulae true on $\langle \{true, false\} \rangle$	164
3-SAT	set of satisfiable boolean formulae in 3-CNF	172

Samenvatting

Speltheorie bestudeert situaties, waarin verscheidene spelers voorkomen, die elk een eigen agenda (*utility function*) hebben. De speltheorie heeft een grote reikwijdte wat betreft toepassingen. Zo wordt speltheorie niet alleen gebruikt om radiofrequenties te verwerven, maar ook voor het analyseren van voortplantingsgedrag van natuurlijke organismen. Zodoende wordt het woord “speler” bijzonder veelomvattend gebruikt binnen de speltheorie: commerciële bedrijven zijn spelers, evenals de Nederlandse overheid en bloemetjes en bijtjes. Evenzo heeft ook het woord “spel” een ruimere betekenis dan het doorgaans toegekend wordt in het Nederlands.

In veel spelen zijn spelers onvolledig geïnformeerd over de feitelijke stand van zaken. Een duidelijk voorbeeld hiervan vormt het bordspel Scotland Yard, waarin de politie een boef moet inrekenen, die zijn locatie slechts op gezette tijden prijsgeeft. Speltheoretische spelen met *onvolledige informatie* (*imperfect information*) staan centraal in dit proefschrift. In het bijzonder probeer ik in dit proefschrift een gevoel te ontwikkelen voor de manier, waarop onvolledige informatie de spelen *moeilijker* maakt. Iedereen, die bekend is met Scotland Yard, zal bijvoorbeeld beamen, dat het gemakkelijker wordt voor de politie om de boef te vangen, als de laatste zich vaker moet laten zien.

De “moeilijkheid van een spel” wordt in dit proefschrift voornamelijk gemeten met behulp van de maten, die ontwikkeld zijn in een tak van de theoretische informatica: complexiteitstheorie. In deze discipline wordt de moeilijkheid, of *complexiteit*, van een probleem (voornamelijk) gemeten aan de hand van de hoeveelheid rekentijd of harde schijfruimte, die een computer nodig heeft om het probleem op te lossen. De moeilijkheid van een spel is dan gedefinieerd als de moeilijkheid van het probleem om te berekenen of een bepaalde speler een manier van spelen heeft die winst garandeert (*winnende strategie*). Er zijn enkele zeer algemene studies gedaan naar dit onderwerp en hun conclusie luidt, dat spelen met onvolledige informatie moeilijker zijn dan spelen met volledige informatie. Uit deze studies kan echter niet worden afgeleid welke specifieke spelen binnen een

bepaalde groep van spelen de complexiteit van die groep als geheel opdrijven. Het is mogelijk, dat er spelen met onvolledige informatie zijn die weliswaar bijzonder complex zijn, maar voor geen enkel vakgebied van belang zijn.

In mijn proefschrift bestudeer ik vier groepen spelen met onvolledige informatie: één per hoofdstuk. Elke groep spelen richt zich op een andere toepassing; drie van de vier zijn echter duidelijk georiënteerd op de logica. Dientengevolge speelt logica een belangrijke rol binnen dit proefschrift. Dit betekent dat dit proefschrift zich op het snijvlak van speltheorie, informatica en logica bevindt, met onvolledige informatie als rode draad. De resultaten, die in dit proefschrift behaald zijn, zijn echter niet alleen bijdragen aan de kennis van spelen met onvolledige informatie. Per hoofdstuk hangt de onderzoeksrichting ook af van de voor de onderhavige toepassing relevante vraagstukken.

Ook logica ligt binnen het bereik van speltheorie. Eén rol, die speltheorie vervult binnen de logica, is het leveren van zogenoemde *speltheoretische karakterisering*en van haar concepten. Dergelijke karakterisering en herdefiniëren concepten uit de logica gebruikmakend van speltheoretische noties, zoals spelers, agenda's en informatie. Om één of andere reden, die niet goed begrepen wordt, plegen speltheoretische karakterisering en een intuïtiever beeld op te leveren van het onderhavige logische concept. Deze karakterisering en stellen de logicus dan ook in staat om dieper tot het wezen van de logische concepten door te dringen, en meer over hun eigenschappen te weten te komen. In Hoofdstuk 3, 4 en 5 worden speltheoretische karakterisering en bekeken, die spelen met onvolledige informatie opleveren.

Hoofdstuk 1 is het inleidende hoofdstuk van dit proefschrift. De lezer treft hier een korte beschrijving van de theoretische achtergrond aan en motiverende vragen.

Hoofdstuk 2 is een zeer beknopte uiteenzetting van de definities van de termen, die in dit proefschrift gebezigd worden.

In Hoofdstuk 3 wordt *Independence-friendly logic* (IF-logica) bestudeerd. IF-logica breidt eerste-orde logica uit door middel van *geslashte* kwantoren: $(\exists x/Y)$, die een begrip van kwantonafhankelijkheid formaliseren. De speltheoretische karakterisering van IF-logica modelleert deze onafhankelijkheid door middel van onvolledige informatie. Het is bekend dat de complexiteit van IF-logica hoger is dan die van eerste-orde logica. In Hoofdstuk 3 bestudeer ik twee fragmenten van IF-logica met als doel de oorzaken van deze hogere complexiteit te begrijpen. Deze twee fragmenten zijn respectievelijk gemotiveerd vanuit de speltheorie en de theoretische informatica (computationele logica).

Hoofdstuk 4 geeft een speltheoretische karakterisering van zogenoemde *partially ordered connectives*. *Partially ordered connectives* zijn een variatie op de bekende Henkin kwantoren, ook wel bekend als *partially ordered quantifiers*. De resultaten in Hoofdstuk 4 suggereren, dat variaties op een logisch concept gekarakteriseerd kunnen worden als variaties op de speltheoretische karakterisering van dit logische concept. Dit bevestigt het gevoel dat logica en speltheorie nauw

verbonden zijn. In de rest van Hoofdstuk 4 staat de analyse van logica's met *partially ordered connectives* centraal, voornamelijk vanuit het oogpunt van de *descriptive complexity*. Laatstgenoemde discipline biedt een perspectief op complexiteitstheorie vanuit de logica en hanteert als zodanig een fijner begrip van complexiteit.

Hoofdstuk 5 richt zich op de partieel geordende kwantoren zoals die gebruikt worden in de formele semantiek. Een motiverende zin uit het Engels voor het gebruik van partieel geordende kwantoren luidt *Most boys and most girls dated each other*. Ik benader partieel geordende kwantoren vanuit de speltheorie en de complexiteitstheorie. In het eerste deel wordt een nieuw speltheoretisch raamwerk opgebouwd, waarin partieel geordende kwantoren bestudeerd kunnen worden. Gebruikmakend van de complexiteitstheoretische noties “meet” ik de complexiteit van kwantoren, die voorkomen in natuurlijke taal. Het blijkt, dat de partieel geordende kwantor, die gebruikt wordt in de formalisatie van bovenstaande zin (*branching most*) een relatief hoge complexiteit heeft (**NP**-volledig).

In Hoofdstuk 6 wordt het bordspel Scotland Yard onder de loep genomen, dat wil zeggen, een wiskundige abstractie van Scotland Yard. Deze abstractie stelt mij in staat om ook voor de volledige informatievariant van Scotland Yard de moeilijkheid te bepalen. Zoals ik in het begin van deze samenvatting stelde is Scotland Yard met volledige informatie gemakkelijker te spelen voor de politie. Daarom is het des te opvallender, dat volgens de complexiteitstheorie Scotland Yard met volledige informatie even moeilijk is als Scotland Yard met onvolledige informatie.

Hoofdstuk 7 besluit het proefschrift met enkele conclusies.

Titles in the ILLC Dissertation Series:

ILLC DS-2001-01: **Maria Aloni**

Quantification under Conceptual Covers

ILLC DS-2001-02: **Alexander van den Bosch**

Rationality in Discovery - a study of Logic, Cognition, Computation and Neuropharmacology

ILLC DS-2001-03: **Erik de Haas**

Logics For OO Information Systems: a Semantic Study of Object Orientation from a Categorical Substructural Perspective

ILLC DS-2001-04: **Rosalie Iemhoff**

Provability Logic and Admissible Rules

ILLC DS-2001-05: **Eva Hoogland**

Definability and Interpolation: Model-theoretic investigations

ILLC DS-2001-06: **Ronald de Wolf**

Quantum Computing and Communication Complexity

ILLC DS-2001-07: **Katsumi Sasaki**

Logics and Provability

ILLC DS-2001-08: **Allard Tamminga**

Belief Dynamics. (Epistemo)logical Investigations

ILLC DS-2001-09: **Gwen Kerdiles**

Saying It with Pictures: a Logical Landscape of Conceptual Graphs

ILLC DS-2001-10: **Marc Pauly**

Logic for Social Software

ILLC DS-2002-01: **Nikos Massios**

Decision-Theoretic Robotic Surveillance

ILLC DS-2002-02: **Marco Aiello**

Spatial Reasoning: Theory and Practice

ILLC DS-2002-03: **Yuri Engelhardt**

The Language of Graphics

ILLC DS-2002-04: **Willem Klaas van Dam**

On Quantum Computation Theory

ILLC DS-2002-05: **Rosella Gennari**

Mapping Inferences: Constraint Propagation and Diamond Satisfaction

- ILLC DS-2002-06: **Ivar Vermeulen**
A Logical Approach to Competition in Industries
- ILLC DS-2003-01: **Barteld Kooi**
Knowledge, chance, and change
- ILLC DS-2003-02: **Elisabeth Catherine Brouwer**
Imagining Metaphors: Cognitive Representation in Interpretation and Understanding
- ILLC DS-2003-03: **Juan Heguiabehere**
Building Logic Toolboxes
- ILLC DS-2003-04: **Christof Monz**
From Document Retrieval to Question Answering
- ILLC DS-2004-01: **Hein Philipp Röhrig**
Quantum Query Complexity and Distributed Computing
- ILLC DS-2004-02: **Sebastian Brand**
Rule-based Constraint Propagation: Theory and Applications
- ILLC DS-2004-03: **Boudewijn de Bruin**
Explaining Games. On the Logic of Game Theoretic Explanations
- ILLC DS-2005-01: **Balder David ten Cate**
Model theory for extended modal languages
- ILLC DS-2005-02: **Willem-Jan van Hoeve**
Operations Research Techniques in Constraint Programming
- ILLC DS-2005-03: **Rosja Mastop**
What can you do? Imperative mood in Semantic Theory
- ILLC DS-2005-04: **Anna Pilatova**
A User's Guide to Proper names: Their Pragmatics and Semantics
- ILLC DS-2005-05: **Sieuwert van Otterloo**
A Strategic Analysis of Multi-agent Protocols
- ILLC DS-2006-01: **Troy Lee**
Kolmogorov complexity and formula size lower bounds
- ILLC DS-2006-02: **Nick Bezhanishvili**
Lattices of intermediate and cylindric modal logics
- ILLC DS-2006-03: **Clemens Kupke**
Finitary coalgebraic logics

ILLC DS-2006-04: **Robert Špalek**

Quantum Algorithms, Lower Bounds, and Time-Space Tradeoffs

ILLC DS-2006-05: **Aline Honingh**

The Origin and Well-Formedness of Tonal Pitch Structures

ILLC DS-2006-06: **Merlijn Sevenster**

Branches of imperfect information: logic, games, and computation