The Logic in Computer Science Column

by

Yuri Gurevich

Microsoft Research
One Microsoft Way, Redmond WA 98052, USA
gurevich@microsoft.com

H :
, , [*]

Merlijn Sevenster
ILLC, Universiteit van Amsterdam
sevenstr@science.uva.nl

Abstract

In this paper, we study the game-theoretic and computational repercussions of Henkin's partially ordered quantifiers [19]. After defining a game-theoretic semantics for these objects, we observe that tuning the parameter of absentmindedness gives rise to quantifier prefixes studied in [28]. In the interest of computation, we characterize the complexity class $\mathrm{P}_{\parallel}^{\mathrm{NP}}$ in terms of partially ordered quantifiers, by means of a proof different from Gottlob's [17]. We conclude with some research questions at the interface of logic, game theory, and complexity theory.

[*] We gratefully acknowledge Johan van Benthem, Peter van Emde Boas, Yuri Gurevich, Marcin Mostowski, Eric Pacuit, Gabriel Sandu, Tero Tulenheimo, and the anonymous reviewer. This paper was finalized at Stanford University, whom we thank for hosting us as a visiting scholar.

1 Setting the stage

Henkin's partially ordered quantifier prefixes were initially introduced as a mathematical exercise [19], but have ever since been the subject of lively discussion in various disciplines. In linguistics, Hintikka [20] and Barwise [4] argued that partially ordered or *branching* quantifiers should be added to the linguist's toolbox to give certain natural language expressions their correct logical form. Sandu and Hintikka [21, 23, 33] imported the idea of a partial dependence relation between quantifiers in first-order logic, resulting in *Independence Friendly logic*. Independence Friendly logic has congenially been given a semantics in terms of *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.

In this paper we aim to show some of the repercussions of Henkin's exercise from a game-theoretic and (finite) model-theoretic angle. Game theory has penetrated logic successfully, providing an interactive and goal-oriented viewpoint on concepts in logic. The game-theoretic viewpoint allows us to compare logics in terms of the interaction, goals and knowledge.

In Section 2, we introduce logics with Henkin quantifiers and recall their model-theoretic behavior.

In Section 3, we show that Henkin quantifiers are played by agents with a limited number of memory cells, whereas first-order logic is played by Eloise enjoying an infallible faculty of memory.

In Section 4, we give a finite model-theoretic account of Henkin quantifiers. Finite model theory is the model-theoretic face of complexity theory, and provides a neat algorithmic view on Henkin quantifiers.

Section 5 concludes the paper.

2 Logic

Henkin's novelty in the theory of quantification is nowadays known under the header of *Henkin quantifier*. A Henkin quantifier is a two-dimensional object of the form

$$\begin{pmatrix} \forall x_{11} & \dots & \forall x_{1k} & \exists y_1 \\ \vdots & \ddots & \vdots & \vdots \\ \forall x_{n1} & \dots & \forall x_{nk} & \exists y_n \end{pmatrix}, \tag{1}$$

where $\mathbf{x}_i = x_{i1}, \dots, x_{ik}$. Henceforth, a string of variables is referred to by using the obvious symbol boldfaced.

With every Henkin quantifier (1) is associated its *dimensions* n and k. In the interest of space, we abbreviate the Henkin quantifier (1) as $\mathsf{H}^k_n\mathbf{xy}$. If no confusion threatens, we may also skip the variables and the integers indicating the dimensions, and simply write H^k_n and H. To identify a Henkin quantifier without referring to its dimensions, we write $\mathsf{H}_{(i)}$.

The two-dimensional way of representation aims to convey that the variable y_i depends on $\mathbf{x}_i$ and on $\mathbf{x}_i$ only. This is formalized by means of the notion of *Skolem function*, that underlies its semantics. Let $\models$ be the satisfaction relation properly defined for the formula $\phi(\mathbf{x}, \mathbf{y})$ on the structure $\mathfrak{A}$. Then, $\models$ is extended in the following way: $\mathfrak{A} \models \mathsf{H}^k_n\ \phi(\mathbf{x}, \mathbf{y})$ iff there exist k-ary functions $f_1, \dots, f_n$ on the universe of $\mathfrak{A}$ such that

$$\mathfrak{A} \models \forall \mathbf{x}_1 \dots \forall \mathbf{x}_n\ \phi(\mathbf{x}_1, \dots, \mathbf{x}_n, f_1(\mathbf{x}_1), \dots, f_n(\mathbf{x}_n)).$$

Note that the quantifier $\mathsf{H}^0_1 x$ has the same semantics as the quantifier $\exists x$, and that H^0_i is elementary definable for every $i \geq 1$.

In this paper, we are interested in two logics featuring Henkin quantifiers. The first one, denoted **H**, contains all strings of the form $\mathsf{H}^k_n\ \psi$, where ψ is first-order and n and k are arbitrary integers. The second logic's formulae are generated by the following grammar:

$$\phi \quad ::= \quad \psi \mid \neg\phi \mid \phi \vee \phi \mid \exists x\ \phi \mid \mathsf{H}^k_n\ \phi,$$

where ψ is first-order and n and k are arbitrary integers. Let us refer to the latter logic by $\mathbf{H}^*$.

For a thorough introduction to the logics **H** and $\mathbf{H}^*$ and their model-theoretic behavior, we refer the reader to [27].

The set of free variables $\mathit{Free}(\phi)$ in the $\mathbf{H}^*$-formula ϕ is inductively defined by the clauses that define the set of free variables $\mathit{Free}(\psi)$, for first-order ψ, plus the clause

$$\mathit{Free}\left(\mathsf{H}^k_n x_{11} \dots x_{nk} y_1 \dots y_n\ \phi\right) \quad = \quad \mathit{Free}(\phi) - \{x_{11}, \dots, x_{nk}, y_1, \dots, y_n\}.$$

An $\mathbf{H}^*$-formula without free variables is called a *sentence*. The satisfaction relation for formulae with free variables is defined in the standard way using assignments.

As an illustration of the expressive power of **H**, consider the following sentence:

$$\zeta \quad = \quad \begin{pmatrix} \forall x_1 & \exists y_1 \\ \forall x_2 & \exists y_2 \end{pmatrix} \exists z_1 \exists z_2 \exists z_3\ (\phi_1 \wedge \phi_2 \wedge \phi_3),$$

where

$$\begin{aligned} \phi_1 &= (x_1 = x_2) \rightarrow (y_1 = y_2) \\ \phi_2 &= R(x_1, x_2) \rightarrow (y_1 \neq y_2) \\ \phi_3 &= \bigwedge_{i \in \{1,2\}} \bigvee_{j \in \{1,2,3\}} (y_i = z_j). \end{aligned}$$

Let $\mathfrak{G}$ be a graph whose edges interpret R. Then by definition, $\mathfrak{G} \models \zeta$ iff for two unary functions f_1 and f_2 on the universe of $\mathfrak{G}$, (1) f_1 and f_2 are the same; (2) if x_1 and x_2 are joined by an edge, then $f_1(x_1) \neq f_2(x_2)$; and (3) the range of f_1 and f_2 is restricted to z_1, z_2, z_3. All in all, we see that f_1 (or f_2 for that matter) is a witness of $\mathfrak{G} \models \zeta$ iff it is a *3-coloring* of $\mathfrak{G}$.

The semantics for Henkin quantifiers overtly mentions functions, that reflect the (in)dependence relation between the universal and existential quantifiers carried by the Henkin quantifier. Bearing this in mind, it is quite straightforward to show that the truth condition of any sentence from **H** can be expressed in second-order, existential logic, symbolically Σ^1_1. It is a milestone result in the theory of partially ordered quantification that the converse holds as well: When it comes to expressive power, **H** and Σ^1_1 are equivalent, cf. [10, 44]. It was shown [10] that every **H***-sentence is equivalent to a sentence in Σ^1_2 and a sentence in Π^1_2. This finding renders **H*** translatable into Δ^1_2. Mostowski [29] showed that the converse does not hold: there is a sentence in Δ^1_2 that has no equivalent in **H***. These results invariably apply to structures of arbitrary cardinality. In case one restricts oneself to finite ordered structures, a very nice computational characterization of **H*** can be given, see [17] and also Section 4 of the current publication.

3 Games

There is a respectable tradition in logic to give game-theoretic accounts of concepts in logic. An early case in point are Lorenzen-style *dialogue games*. They are typically two-player games between Proponent and Opponent. Dialogue games aim to give a game-theoretic underpinning of the concept of proof. That is, a formula ϕ is provable in a logical system if, and only if, the Proponent has a way of playing the dialogue game of ϕ for the logical system at hand that wins against every way of playing by Opponent. In the game-theorist's parlance, we say that Proponent has a *winning strategy*.

So-called *game-theoretic semantics* was introduced by Hintikka giving a game-theoretic account of truth. For instance, consider a toy fragment of first-order logic, containing only strings of the form

$$\mathsf{Q}_1 x_1 \ldots \mathsf{Q}_n x_n\ R(\mathbf{x}),$$

where $\mathsf{Q}_i \in \{\exists, \forall\}$. Being a fragment of first-order logic, Tarski-semantics is properly defined for this toy language. But the Tarskian satisfaction relation can also be given a game-theoretic face, yielding games between Eloise and Abelard. To this end, let $\mathfrak{A}$ be a structure that interprets the predicate R and let the *semantic* *game* for a formula ϕ from the toy language on $\mathfrak{A}$ start from the position $\langle \phi, \mathfrak{A} \rangle$. The proceedings of this game are determined by the following game rules:

- If the position is $\langle \exists x_i \, \phi, \mathfrak{A} \rangle$, Eloise picks an object a_i from the universe of $\mathfrak{A}$, and the game continues as $\langle \phi, \mathfrak{A} \rangle$.

- If the position is $\langle \forall x_i \, \phi, \mathfrak{A} \rangle$, Abelard picks an object a_i from the universe of $\mathfrak{A}$, and the game continues as $\langle \phi, \mathfrak{A} \rangle$.

- If the position is $\langle R(x_1, \ldots, x_n), \mathfrak{A} \rangle$, the game ends. Eloise wins if the tuple $\langle a_1, \ldots, a_n \rangle$ that was built up during the game stands in the R-relation in $\mathfrak{A}$; otherwise Abelard wins.

The adequacy of the semantic games for the toy language, is typically cast as follows. For every formula ϕ and suitable structure $\mathfrak{A}$, $\mathfrak{A} \models \phi$ iff Eloise has a *winning strategy* in the semantic game of ϕ on $\mathfrak{A}$. As the reader acknowledges if we extend the toy language with connectives, negations, one also has to extend the set of game rules (and possibly tweak the current one) to maintain adequacy of the game-theoretic semantics with respect to the new logical system.

By this token it becomes clear that classes of semantic games should not be conceived of as objects floating in limbo. Just as one can compare the properties of two logical systems by means of model-theoretic means, one can compare the semantic games they give rise to. Again, Lorenzen's dialogue games are a case in point. A lively debate was held about the viability of the dialogue games for first-order logic in contradistinction to the dialogue games for Brouwer's intuitionistic logic. Some held the conviction that dialogue games for intuitionistic logic are 'more natural' than the ones for first-order logic, and took this as an argument in favor of Brouwer's system, cf. [41].

From the same viewpoint, the move from first-order logic to Independence Friendly logic can be appreciated. From a purely game-theoretic angle, Hintikka and Sandu [21, 23, 33] generalized the semantic games for first-order logic so as to incorporate imperfect information. In our view, this very argument may count as a motivation for Independence Friendly logic in itself. What exactly is the influx of the imperfect information in semantic games for Independence Friendly logic is a hard question, and definitely a topic for future research. As we pointed out, the idea of partial dependence relation over quantifiers in Independence Friendly logic has its precursor in Henkin's work. So from this angle alone it is worthwhile to develop at least some understanding of the game-theoretic face of Henkin quantifiers, involving imperfect information.

For the sake of simplicity let us restrict ourselves to **H**-formulae in which the first-order part is atomic. On this assumption, the game rules for the semantic game of the **H**-sentence

$$\mathsf{H}_n^k x_{11} \ldots x_{nk} y_1 \ldots y_n \; R(\mathbf{x}, \mathbf{y})$$

are simply the ones for the semantic game for

$$\forall x_{11} \ldots \forall x_{1k} \exists y_1 \ldots \forall x_{n1} \ldots \forall x_{nk} \exists y_n \; R(\mathbf{x}, \mathbf{y}).$$

But as the latter sentence is a sentence from our toy language, the game seems to have become a game with perfect information as before.

How can this be?

On second thought, it turns out that we have been a bit careless when introducing the semantic games for the toy language. Surely we gave the players eloquent names, but omitted to specify the players are such that the semantic games in which they participate would actually be modeled as games with *perfect information*. It would have made little sense, for instance, to declare that we think of Eloise as a cauliflower. It's not that cauliflowers cannot be regarded as game-theoretic agents, witness the literature on evolutionary game theory. Rather, had we done so modeling the semantic game of a formula from the toy language as a game with perfect information would be counterintuitive, to say the least.

To be on the safe side, we'd better postulate that Eloise has an infallible faculty of memory.

The imperfect information in semantic games for **H**-sentences $\mathsf{H}_n^k \; R(\mathbf{x})$ can be seen to be brought into being by assuming that each agent has exactly k memory cells. Henceforth, we shall assume that agents govern these memory cells in a *first in first out* manner. In unison, these assumption imply that when the agent is deciding on an object for y_i, it knows only the objects picked up over the k previous rounds, that is, the objects assigned to $x_{i1}, \ldots, x_{ik}$. Furthermore, I postulate that this agent is not *absentminded*, that is, it knows in which round of the game it is. This postulate implies that when choosing an object to assign to y_i the agent knows that the object selected will be assigned to y_i and not to, say, y_{i+1} or y_{i-1}.

In this manner, every **H**-sentence $\phi = \mathsf{H}_n^k \; R(\mathbf{x})$ and structure $\mathfrak{A}$ give rise to a semantic game that would be modeled as an *extensive game with imperfect information*, call it *Sem-game*$^{\mathbf{H}}(\phi, \mathfrak{A})$. An extensive game with imperfect information is a rigorous mathematical object $\langle N, H, P, \langle \mathfrak{I}_i \rangle_{i \in N}, W \rangle$, well-known from game theory [30]. N is the set of *players*. H is the set of *histories*—all permissible sequences of actions in the game. P is the *player function* deciding which player $P(h) \in N$ is to move at history h. $\mathfrak{I}_i$ is a partition of the histories in which player i is to move, modeling the imperfect information. W is the *win function*, that decides who has won when the game has come to an end.

In the context of ϕ and $\mathfrak{A}$, the set H equals

$$\bigcup_{0 \leq i \leq ((n \cdot k)+n)} A^i,$$

where A is the universe of $\mathfrak{A}$. With every history $h \in H$ of length $((n{\cdot}k) + n)$—i.e., *terminal history*—we straightforwardly associate an assignment function $\mathbf{a}_h$ to the variables $x_{11}, \ldots, x_{nk}, y_1, \ldots, y_n$.

Sem-game$^{\mathbf{H}}(\phi, \mathfrak{A})$ can be regarded as a tree structure—a *game tree*—defined by the prefix relation on H. The game tree is decorated by P.

The set $\mathfrak{I}_i$ contains all sets of histories that are indistinguishable for our k-cell, non-absentminded agent (first in first out, remember). The particulars of the agent at hand uniquely determine $\mathfrak{I}_i$. That is, $h, h' \in I \in \mathfrak{I}_i$ if, and only if, h and h' are equally long (non-absentmindedness) and the last k elements of h and h' coincide (k-cell and first in first out). Clearly, $W(h) =$ Eloise iff $\langle \mathbf{a}_h(x_{11}), \ldots, \mathbf{a}_h(x_{nk}), \mathbf{a}_h(y_1), \ldots, \mathbf{a}_h(y_n) \rangle$ is an R-tuple in $\mathfrak{A}$.

Any function $S : \mathfrak{I}_i \to A$ is a strategy for player i in *Sem-game*$^{\mathbf{H}}(\phi, \mathfrak{A})$. Say that a strategy for player i is *winning*, if i following the strategy at each of i's moves only results in terminal histories h such that $W(h) = i$,

Let $\phi = \mathsf{H}^k_n\, R(\mathbf{x})$ and let $\mathfrak{A}$ be a structure interpreting R. Let *Sem-game*$^{\mathbf{H}}(\phi, \mathfrak{A})$ be the extensive game with imperfect information modeling the semantic game of ϕ on $\mathfrak{A}$ played by a k-memory cell agent.

Proposition 1. *For every* **H**-*sentence* $\phi = \mathsf{H}^k_n\, R(\mathbf{x})$ *and structure* $\mathfrak{A}$ *interpreting* R*, a non-absentminded agent with* k *memory cells has a winning strategy in the semantic game Sem-game*$^{\mathbf{H}}(\phi, \mathfrak{A})$ *iff* $\mathfrak{A} \models \phi$.

Proof. The proof is straightforward once one notices that a series of Skolem functions $f_1, \ldots, f_n$ witnessing $\mathfrak{A} \models \phi$ encodes a winning strategy in *Sem-game*$^{\mathbf{H}}(\phi, \mathfrak{A})$, and vice versa. □

It was observed in [27, pg. 223] that many **H**-sentences appearing in the literature express the existence of *one* single function on the universe. The sentence ζ that expresses 3-colorability of graphs we discussed earlier is a case in point. In the same vein many other interesting **H**-sentences sit in a certain fragment of **H**, that was studied in [28]. This particular fragment is defined by the *function quantifier* F^k_n, that binds the variables $x_{11}, \ldots, x_{nk}, y_1, \ldots, y_n$, just like the Henkin quantifier with dimensions n and k. (We will adhere to the same notational conventions as with Henkin quantifiers.) The logic **F** is defined to be the language containing all strings (sentences) of the form

$$\mathsf{F}^k_n\, \mathbf{x}_1 \ldots \mathbf{x}_n y_1 \ldots y_n\, R(\mathbf{x}_1, \ldots, \mathbf{x}_n, y_1, \ldots, y_n), \tag{2}$$

where $\mathbf{x}_i = x_{i1}, \ldots, x_{ik}$ as before and R is an atom. As regards its semantics, any formula (2) is true on a structure $\mathfrak{A}$ interpreting R iff there exists *one single* k-ary function f on the universe of $\mathfrak{A}$ such that

$$\mathfrak{A} \models \forall \mathbf{x}\, R(\mathbf{x}_1, \ldots, \mathbf{x}_n, f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)).$$

Henkin quantifiers differ from function quantifiers in that the former allow for multiple functions $f_1, \ldots, f_n$, whereas function quantifiers allow for only one. For a model-theoretic comparison of logics with Henkin quantifiers and function quantifiers see [18, 28].

From a game-theoretic point of view, we show that the move from Henkin quantifiers to function quantifiers resembles to imposing absentmindedness on our k-cell agent playing according to the game rules of the semantic game of $\forall \mathbf{x}_1 \exists y_1 \ldots \forall \mathbf{x}_n \exists y_n\, R(\mathbf{x}, \mathbf{y})$ on $\mathfrak{A}$. So in particular the game rules for the sentence

$$\psi \quad = \quad \begin{pmatrix} \forall x_1 & \exists y_1 \\ \forall x_2 & \exists y_2 \end{pmatrix} R(\mathbf{x}, \mathbf{y}),$$

where

$$R(\mathbf{x}, \mathbf{y}) \quad = \quad (x_1 = x_2 \rightarrow y_1 = y_2) \wedge (y_1 = x_2 \rightarrow y_2 = x_1) \wedge (x_1 \neq y_1).$$

on the structure $\mathfrak{B}$ would be equal to the ones for $\forall x_1 \exists y_1 \forall x_2 \exists y_2\, R(\mathbf{x}, \mathbf{y})$. (The sentence ψ characterizes the finite structures whose universes have even cardinality, see [35].) Considering an absentminded 1-cell agent, we see that during neither of his rounds it knows whether the object it choses will be assigned to y_1 or y_2; it is aware of the last action though. So in particular if a, b, c are three different objects from the universe of $\mathfrak{B}$, it cannot tell apart the histories $\langle a, b, c\rangle$ and $\langle c\rangle$. On the other hand it can distinguish $\langle c\rangle$ from $\langle a\rangle$ and $\langle c, b, a\rangle$.

Just as we had with $\mathbf{H}$, if ϕ is an $\mathbf{F}$-sentence let *Sem-game*$^{\mathbf{F}}(\phi, \mathfrak{A})$ be the extensive game with imperfect information that models an absentminded agent with k memory cells in the latter game. In particular in *Sem-game*$^{\mathbf{F}}(\psi, \mathfrak{B})$ there is an information partition containing both $\langle a, b, c\rangle$ and $\langle c\rangle$, but not $\langle a\rangle$ and $\langle c, b, a\rangle$. Generally speaking, in these extensive games with imperfect information for $\mathbf{F}$, two histories h and h' sit in the same information partition, if the last k elements in h and h' coincide. However as we saw before, h and h' need not be of equal length.

Proposition 2. *For every* $\mathbf{F}$*-sentence* $\phi = \mathsf{F}^k_n\, R(\mathbf{x})$ *and structure* $\mathfrak{A}$ *interpreting* R*, an absentminded agent with k memory cells has a winning strategy in the semantic game Sem-game*$^{\mathbf{F}}(\phi, \mathfrak{A})$ *iff* $\mathfrak{A} \models \phi$.

The reader may wonder, what's next? Well, in the same vein one may restrict the agent's powers to an even greater extent and supply it with a fixed array of actions. Recall that in the semantic games for $\mathbf{H}$ and $\mathbf{F}$ the agents pick up their actions from the universe of the structure at hand, that has unbounded cardinality. If we consider the agent to be non-absentminded and in possession of a fixed and finite number of actions, it is capable of 'playing Henkin quantifiers with restricted quantifiers', see [5, 34, 35]. To the best of my knowledge the logic that is played by absentminded agents with a limited number of memory cells and a fixed number of actions has not been studied.

In semantic games for **H**, the k-cell agent is supposed to recall only the last k variables. This undoubtedly is an assumption without theoretical backing. Hintikka and Sandu [21, 23, 33] overcome this needless restriction by introducing the / item in first-order logic, to indicate knowledge of a variable or absence thereof. The resulting system is the Independence Friendly logic we spoke of earlier. In this logic, the sentence

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

gives rise to games in which Eloise does not know x_1 when deciding for y_1; but she recollects it when she is to decide for y_2. Given the syntactic formation rules of Independence Friendly logic, one infers rather straightforwardly that every *pattern of ignorance* concerning objects previously played can be accounted for. That is, if we have a first-order formula ϕ in whose semantic games the occurrence of $\exists x$ triggers a move for Eloise informed about $x_1,\ldots,x_n$, then the / item allows one to limit the knowledge of Eloise to any subset of $\{x_1,\ldots,x_n\}$. From this game-theoretic perspective Independence Friendly logic truly is the imperfect information generalization of first-order logic. But note that some sentences from Independence Friendly logic give rise to games that are hard to actually play, as they violate *perfect recall*, cf. [8, 40, 41]. A perfect information approach to Independence Friendly logic was pursued in [38].

Even more delicate flows of information were studied in the *Partial Information logic* by Parikh and Väänänen [32] whose formulae give rise to imperfect information games in which Eloise may be partially informed about the previous actions. In semantic games for the first-order formula $\forall x\exists y\ R(x,y)$, for instance, Eloise knows the object assigned to x. 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 maps every object on x itself Eloise is aware of x after all. But f may just as well return 1 if x is even and 0 otherwise. In this manner, if P is a predicate, the formula $\chi = \forall x(\exists y/\!\!/P(x))\ (x \neq y)$ gives rise to games in which Eloise does not know x, but she knows whether Abelard chose a P-object. The formula χ can thus be seen true on any structure in which there is a P-object and a non-P-object. Under specific conditions on the nature of the functions appearing at the right-hand side of the $/\!\!/$ device, Partial Information logic is a decidable fragment of first-order logic.

It has been pointed out by various authors [22, 24, 41] that we are not really interested in the actual game playing of semantic games. To the ends we employ them it is very much indifferent what strategy is used, for instance, and whether the game is actually played in a platonic universe. Instead we are interested in the statements we can truthfully make *about* these games, in particular in the existence of winning strategies. There is one viewpoint from which this difference becomes clear, that we will highlight. There is a discrepancy between the complexity of the players and the complexity of the statements we make about them, or—more precisely—the expressive power of the logic required to express the winning conditions of Eloise. We saw that Eloise enjoys an infallible faculty of memory in the semantic games for first-order logic, or the toy fragment thereof. Yet, ipse facto, it takes the first-order sentence ϕ to express whether Eloise has a winning strategy in the semantic game of ϕ on any structure. On the other hand, we hired an agent with a limited number of memory cells to play the semantic games for **H**. As was pointed out in [10, 44], here we have to resort to the expressive power of full Σ^1_1!

Note that such a discrepancy does not always occur. For instance, limit attention to 0-cell agents, i.e., agents that don't see any of their opponent's actions. Then, Henkin quantifiers that are playable by such an agent look like

$$\begin{pmatrix} \exists y_1 \\ \vdots \\ \exists y_n \end{pmatrix},$$

and are clearly defined by the first-order prefix $\exists y_1 \ldots \exists y_n$.

There is no a priori reason to stick to expressive power as the single measure of complexity. Van Benthem [39] takes up the axiomatization of game models with imperfect information, and needs *extra* axioms to enforce perfect information. Yet the axiom system seems to get more simple when k-cell agents are considered.

4 Computation

Fagin [11] gave birth to the area of descriptive complexity, revealing an intimate connection between model theory and the theory of computation. Descriptive complexity concerns itself with connecting up logical languages and complexity classes. This enterprise departs from the insight that with every logical sentence there is a computational cost associated to verifying its semantic value on an arbitrary finite structure; and the other way around, that the particulars of a computing device can be described in logic. The hope is that hard questions from complexity theory (think of P versus NP and NP versus coNP) can be solved by separating the logics they are associated with, see also Section 5.

In this section we will take up the descriptive complexity analysis of **H***. This will give us an algorithmic view on Henkin quantifiers. Furthermore it gives some insight in the way partially ordered quantifiers manifest themselves in the theory of computation. A more general variant of Theorem 7 from this section appeared in an excellent paper by Gottlob [17]. The references we use do not build on any of Gottlob's results nor on his main references.[1] An independent proof, that is. The descriptive complexity of $\mathbf{H}^*$ was raised as an open problem in [5].

First we give a recap of the basics of finite model theory and descriptive complexity.

Let σ be a finite set of relation symbols—a *vocabulary*—each of which comes with an integer, its *arity*. Every vocabulary contains the binary relation symbol =.

Let a σ-structure $\mathfrak{A}$ be an object of the form $\langle A, \langle R^{\mathfrak{A}} \rangle_{R \in \sigma} \rangle$, where A is the universe of $\mathfrak{A}$ and $R^{\mathfrak{A}} \subseteq A^a$, for a the arity of R. The symbol = is rigidly evaluated as the identity relation. If $< \in \sigma$, then $<^{\mathfrak{A}}$ shall be a linear order on A, and $\mathfrak{A}$ is called a *linear ordered structure*. If A is finite, $\mathfrak{A}$ is called a *finite structure*. Here and henceforth, all discussion will be restricted to finite structures unless indicated otherwise.

Sometimes when we write $\mathfrak{A}$ we actually mean *the binary encoding of* $\mathfrak{A}$. We refer the reader to Immerman's textbook [26], in which a detailed account is given of how one can encode structures in binary. For our ends, it suffices to take notice of the fact that the length of the binary encoding of a σ-structure $\mathfrak{A}$, symbolically $||\mathfrak{A}||$, is of size A^c, for some constant c depending on σ.

Let $\mathcal{K}$ be a class of σ-structures. A *property* Π over $\mathcal{K}$ is a function assigning a truth value $\Pi(\mathfrak{A}) \in \{\mathit{false}, \mathit{true}\}$ to every structure $\mathfrak{A}$ from $\mathcal{K}$. Let $\mathbf{L}$ be a logic, i.e., a set of sentences, for which the satisfaction relation $\models$ is defined. Every $\mathbf{L}$-sentence ϕ defines a property Π_ϕ on $\mathcal{K}$, where

$$\Pi_\phi(\mathfrak{A}) = \mathit{true} \qquad \text{iff} \qquad \mathfrak{A} \models \phi,$$

for every $\mathfrak{A} \in \mathcal{K}$. We say that ϕ and $\mathbf{L}$ *express* Π_ϕ. So the sentence ζ expresses the graph-property of 3-colorability.

Let $\mathbf{L}$ and $\mathbf{L}'$ be two languages over the same vocabulary. Then, write $\mathbf{L} \leq_{\mathcal{K}} \mathbf{L}'$ to indicate that every property over $\mathcal{K}$ expressible in $\mathbf{L}$ is expressible in $\mathbf{L}'$. Define $=_{\mathcal{K}}$ and $<_{\mathcal{K}}$ in the standard way.

Let C be a complexity class [14, 31]. We say that $\mathbf{L}$ *captures at least* C over $\mathcal{K}$, if each C-decidable property over $\mathcal{K}$ can be expressed by a sentence from $\mathbf{L}$ in the vocabulary of σ. We say that the *query complexity* of $\mathbf{L}$ over $\mathcal{K}$ is in C, if for every sentence ϕ in $\mathbf{L}$ in the vocabulary σ, the property Π_ϕ over $\mathcal{K}$ is decidable in C. Here it should be borne in mind, that the size of ϕ is constant. The complexity of Π_ϕ is measured solely by the size of the structures. Finally, say that $\mathbf{L}$ *captures* C over $\mathcal{K}$, if $\mathbf{L}$ captures at least C over $\mathcal{K}$ and the query complexity of $\mathbf{L}$ over $\mathcal{K}$ is in C.

[1] Following the reviewer's suggestion we tag presented proofs of already published results, that differ from the ones given in the literature, with our name.

Descriptive complexity began with *Fagin's Theorem*, in which the complexity class NP is captured.

Theorem 3 ([11])**.** *Over graphs,* Σ^1_1 *captures* NP.

The result can be extended so as to hold for arbitrary structures, cf. [26]. Blass and Gurevich [5] drew upon the connection with Henkin quantifiers and obtained that $\mathbf{H}$ captures NP. This result is readily obtained in virtue of the fact that $\mathbf{H} = \Sigma^1_1$, due to [10, 44]. The remainder of this section is dedicated to the descriptive complexity of $\mathbf{H}^*$.

For future reference, we lay down an easy Prenex normal form result.

Proposition 4. *Every* $\mathbf{H}^*$*-sentence* ϕ *is equivalent to an* $\mathbf{H}^*$*-sentence of the following form:*

$$\pm_1 \mathsf{H}_{(1)}\mathbf{x}_1 \ldots \pm_n \mathsf{H}_{(n)}\mathbf{x}_n\ \psi,$$

where $\pm_i \in \{\neg, \neg\neg\}$ *and* ψ *is first-order.*

Proof. A standard inductive proof suffices, the only non-trivial case being the conjunction. But $\mathsf{H}_{(1)}\mathbf{x}\ \phi_1(\mathbf{x}) \wedge \mathsf{H}_{(2)}\mathbf{y}\ \phi_2(\mathbf{y})$ is easily seen to be equivalent to $\mathsf{H}_{(1)}\mathbf{x}\mathsf{H}_{(2)}\mathbf{z}\ (\phi_1(\mathbf{x}) \wedge \phi_2(\mathbf{z}))$, where $\mathbf{z}$ is a string of variables none of which appear in $\mathbf{x}$. □

Our main observation concerns the computational complexity of $\mathbf{H}^*$, that is associated with the complexity class $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$.[2] This denotes the class of properties decidable in deterministic polynomial time with the help of an NP-oracle that can be asked a polynomial number of queries in parallel only once. The action of querying the oracle takes only one time step. Further, P^{NP} contains those problems decidable in deterministic polynomial time with an NP-oracle. Some grasp a complexity class best by its complete problems, that is, its problems to which every problem in the complexity class can be reduced (by means of a polynomial time, many one reduction). Wagner [42] showed that the graph-property of having an odd chromatic number is $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$-complete. Denote the class of graphs with an odd chromatic number by ODD-COLOR.

Theorem 5 ([17])**.** *The query complexity of* $\mathbf{H}^*$ *is in* $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$.

Proof (M. Sevenster). It suffices to show that for an $\mathbf{H}^*$-sentence ϕ in the vocabulary σ, deciding whether ϕ is true on a finite σ-structure $\mathfrak{A}$ can be done in $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$.

[2] Gottlob's [17] theorem is cast in terms of $\mathrm{LOGSPACE}^{\mathrm{NP}}$, that is, the class of problems decidable in logarithmic space with an NP-oracle. Recall that $\mathrm{LOGSPACE}^{\mathrm{NP}} = \mathrm{P}^{\mathrm{NP}}_{\shortparallel}$, due to [43].

First we describe an algorithm that computes whether ϕ is true on $\mathfrak{A}$. Thereafter we observe that this algorithm can be implemented on a Turing machine that works in $\mathrm{P}^{\mathrm{NP}}_{||}$.

As for the algorithm, due to Proposition 4 we may assume without loss of generality that ϕ has the form:

$$\pm_1 \mathsf{H}_{(1)}\mathbf{x}_1 \ldots \pm_n \mathsf{H}_{(n)}\mathbf{x}_n \, \psi(\mathbf{x}),$$

where ψ is a first-order formula over the variables $\mathbf{x} = \mathbf{x}_1, \ldots, \mathbf{x}_n$. Let the algorithm start off by writing down all variable assignments in $A^{\mathbf{x}}$, and label every such assignment $\mathbf{a}$ with *true* if $\langle \mathfrak{A}, \mathbf{a}\rangle \models \psi(\mathbf{x})$, and *false* otherwise. Note that consequently ψ's truth conditions on $\mathfrak{A}$ are completely spelled out. Since ψ is first-order this can be done in LOGSPACE.

Put $i = n$ and $\chi_{i+1} = \phi$. For every i from n through 1, proceed as follows for $\pm_i \mathsf{H}_{(i)}\mathbf{x}_i$ in ϕ:

- Write down all assignments in $A^{\mathbf{x}_1,\ldots,\mathbf{x}_{i-1}}$.
- For every assignment $\mathbf{a} \in A^{\mathbf{x}_1,\ldots,\mathbf{x}_{i-1}}$ ask the oracle whether $\langle \mathfrak{A}, \mathbf{a}\rangle \models \mathsf{H}_{(i)}\mathbf{x}_i \, \chi_{i+1}$.
- Label $\mathbf{a}$ 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 $A^{\mathbf{x}_1,\ldots,\mathbf{x}_i}$ and let the current list of assignments fully specify the truth conditions of $\chi_i(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1})$; that is, let χ_i be the formula that holds of an assignment $\mathbf{a}$ on $\mathfrak{A}$ if and only if $\mathbf{a}$ is labeled *true*.

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 $n \cdot |A^{\mathbf{x}}|$ steps, which is clearly polynomial in the size of the input, $||\mathfrak{A}||$, because the number of variables in $\mathbf{x}$ is constant. Since $\mathbf{H}$ captures NP it is sufficient (and necessary) to employ an NP-oracle. This renders the algorithm in P^{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 NP-oracle. (Recall that the size of ϕ is constant.) In [6] it was shown that a constant number of rounds of polynomially many queries to an NP-oracle is equivalent to one round of parallel queries. Therefore, the algorithm sits in $\mathrm{P}^{\mathrm{NP}}_{||}$. □

Let $\mathbf{H}^+$ be the *first-order closure* of $\mathbf{H}$. That is, the closure of $\mathbf{H}$ under boolean operations and existential quantification (but not under application of Henkin quantifiers). More formally, $\mathbf{H}^+$ is generated by the following grammar:

$$\phi \quad ::= \quad \psi \mid \neg\phi \mid \phi \vee \phi \mid \exists x\, \phi,$$

where ψ ranges over the $\mathbf{H}$-formulae. The first-order closure of (fragments of) Σ^1_1 was taken up in [2]. In this publication, the authors observe that the first-order closure of Σ^1_1 captures $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$, on linear ordered structures. Since $\mathbf{H} = \Sigma^1_1$, the following result follows directly.

Proposition 6. *Over linear ordered structures,* $\mathbf{H}^+$ *captures* $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$.

It is readily observed from the languages' grammars that every sentence in $\mathbf{H}^+$ is a sentence in $\mathbf{H}^*$ as well. Therefore, for every class of structures $\mathcal{K}$, $\mathbf{H}^+ \leq_{\mathcal{K}} \mathbf{H}^*$. This is actually the last step we have to make to establish the main result.

Theorem 7 ([17])**.** *Over linear ordered structures,* $\mathbf{H}^*$ *captures* $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$.

Proof (M. Sevenster). Let $\mathcal{L}$ denote the class of linear ordered structures. By Theorem 5 we have that $\mathbf{H}^*$'s query complexity is in $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$, also over $\mathcal{L}$. It remains to be proved therefore that $\mathbf{H}^*$ captures at least $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$. To this end, let Π be an arbitrary $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$-decidable property over $\mathcal{L}$. In virtue of Proposition 6, we obtain that there is a sentence ϕ from $\mathbf{H}^+$ that expresses Π over $\mathcal{L}$. As we concluded right before this theorem, for every class of structures $\mathcal{K}$, $\mathbf{H}^+ \leq_{\mathcal{K}} \mathbf{H}^*$. So in particular it is the case that $\mathbf{H}^+ \leq_{\mathcal{L}} \mathbf{H}^*$. Whence, Π is expressible in $\mathbf{H}^*$ as well, and the claim follows. □

We wish to warn the reader who is about to jump to conclusions about parallel computation and partially ordered quantification. Admittedly, the complexity class $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$ is based on parallel Turing machines and it is captured by $\mathbf{H}^*$, on linear ordered structures. However, this does not mean that verifying a single $\mathbf{H}$-formula $\mathsf{H}\mathbf{x}\, \phi$ can be done by parallel means, as this requires 'simply' an NP-machine. The parallel way of computing comes in effect only when we compute the semantic value of several $\mathbf{H}$-formulae at the same moment in time. For instance, if $\mathsf{H}\mathbf{x}\, \phi(y)$ is an $\mathbf{H}$-formula with one free variable y, then verifying all of

$$\langle \mathfrak{A}, a_1\rangle \models \mathsf{H}\mathbf{x}\, \phi(y) \quad \dots \quad \langle \mathfrak{A}, a_m\rangle \models \mathsf{H}\mathbf{x}\, \phi(y)$$

for objects $a_1, \dots, a_m \in A$, can be done in one round of m parallel queries to an NP-oracle. It is this principle that underlies the fact that $\mathbf{H}^*$'s query complexity is in $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$.

On the other hand, it is noteworthy that the very fact that a polynomial number of parallel queries suffice is due to the fact that $\mathbf{H}^*$-formulae do only contain first-order variables. This, namely, makes 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. Interestingly, full second-order logic captures the *Polynomial Hierarchy*, whereas $\mathbf{H}^*$ 'gets stuck' at $\mathrm{P}^{\mathrm{NP}}_{\|}$. In this sense Theorem 5 provides the computational upper-bound of partially ordered, yet first-order, quantification.

One way to appreciate the fact that the logics $\mathbf{H}^+$ and $\mathbf{H}^*$ coincide on linear ordered structures is by means of the *Henkin depth* of $\mathbf{H}^*$-formulae:

$$\begin{array}{rcl} hd(\phi) & = & 0, \quad \text{for first-order } \phi \\ hd(\neg\phi) & = & hd(\phi) \\ hd(\phi \vee \psi) & = & \max\{hd(\phi), hd(\psi)\} \\ hd(\exists x\, \phi) & = & hd(\phi) \\ hd(\mathsf{H}^k_n \mathbf{x}\, \phi) & = & hd(\phi) + 1, \end{array}$$

reading $\mathsf{H}^0_n x_1 \ldots x_n$ as $\exists x_1 \ldots \exists x_n$.

Clearly every $\mathbf{H}^+$-sentence has a Henkin depth of at most one. Therefore, by Theorem 7 we get that for every $\mathbf{H}^*$-sentence ϕ there exists an $\mathbf{H}^+$-sentence ψ, such that $hd(\psi) \leq 1$ and on the class of linear ordered structures ϕ and ψ define the same property. Put differently, on linearly ordered structures granting Henkin quantifiers to nest does not yield greater expressive power. Gottlob proves an even stronger normal form for $\mathbf{H}^*$ on linear ordered structures. In Gottlob's terminology, an $\mathbf{H}^*$-sentence ϕ is in *Stewart normal form*, if it is of the form

$$\exists \mathbf{x}\ \left(\mathsf{H}_{(1)}\mathbf{y}\ \phi_1(\mathbf{x}, \mathbf{y})\ \wedge\ \neg\mathsf{H}_{(2)}\mathbf{z}\ \phi_2(\mathbf{x}, \mathbf{z})\right),$$

where ϕ_1 and ϕ_2 are first-order. This normal form is inspired by the work of Stewart [36, 37], 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 $\mathbf{H}^*$-sentence ϕ there exists an $\mathbf{H}^*$-sentence ψ in Stewart normal form, that expresses the same property.

This result cries out for an effective translation procedure from $\mathbf{H}^*$ into $\mathbf{H}^+$ of course, but unfortunately we 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

$$\begin{pmatrix} \forall u_1 & \exists v_1 \\ \forall u_2 & \exists v_2 \end{pmatrix} \begin{pmatrix} \forall x_1 & \exists y_1 \\ \forall x_2 & \exists y_2 \end{pmatrix} \phi \tag{3}$$

is equivalent to

$$\begin{pmatrix} & & \forall u_1 & \exists v_1 \\ & & \forall u_2 & \exists v_2 \\ \forall u_1 & \forall u_2 & \forall x_1 & \exists y_1 \\ \forall u_1 & \forall u_2 & \forall x_2 & \exists y_2 \end{pmatrix} \phi, \tag{4}$$

see also [5]. 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 [7] raise the question whether $\mathbf{H}^*$ captures $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$ over *unordered* structures. Surprisingly, it turns out that $\mathbf{H}^*$ does not capture $\mathrm{P}^{\mathrm{NP}}_{\shortparallel}$ 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.

Further still, a study by Hyttinen and Sandu [25] implies that essentially one has to make use of the finiteness of the structures. Consider the logical languages

$$\begin{aligned} \mathbf{H}^+_1 &= \mathbf{H} \\ \mathbf{H}'_k &= \text{first-order closure of } \mathbf{H}^+_k \\ \mathbf{H}^+_{k+1} &= \{\mathsf{H}\mathbf{x}\,\phi \mid \phi \in \mathbf{H}'_k\}. \end{aligned}$$

Clearly the Henkin depth of any sentence from $\mathbf{H}^+_k$ is k, and $\bigcup_k \mathbf{H}^+_k = \mathbf{H}^*$. The authors prove that on the standard model of arithmetic the language $\mathbf{H}^+_{k+1}$ has strictly stronger expressive power than $\mathbf{H}^+_k$, for every $k \geq 1$.

For the sake of concreteness, consider the property Odd-color over graphs. By Theorem 7, the similar property over linear ordered graphs is expressible in $\mathbf{H}^*$ (and $\mathbf{H}^+$). A linear ordered graph $\mathfrak{G}$ is a structure $\langle G, R^{\mathfrak{G}}, <^{\mathfrak{G}}\rangle$ such that $\langle G, R^{\mathfrak{G}}\rangle$ is a graph and $<^{\mathfrak{G}}$ is a linear order on G. We claim that ξ expresses Odd-color on linear ordered graphs, where ξ is

$$\exists x_1 \exists x_2\, (\mathit{EVEN}(x_2) \wedge \mathit{SUC}(x_1, x_2) \wedge \mathit{COLOR}(x_2) \wedge \neg \mathit{COLOR}(x_1)).$$

In ξ, *EVEN* is the predicate that holds for exactly those objects that are even with respect to $<$, and *SUC* holds for every pair of objects x_1, x_2 such that x_2 is the immediate $<$-successor of x_1. *EVEN* and *SUC* are clearly expressible in Σ^1_1 and consequently in $\mathbf{H}$. Intuitively, *COLOR* holds for all objects x such that the graph at hand is n-colorable, where n is the number of objects $<$-preceding x. Formally, we define $\mathit{COLOR}(x)$ as follows:

$$\begin{pmatrix} \forall y_1 & \exists z_1 \\ \forall y_2 & \exists z_2 \end{pmatrix} (y_1 = y_2) \rightarrow (z_1 = z_2) \ \wedge\ R(y_1, y_2) \rightarrow (z_1 \neq z_2) \ \wedge\ (z_1 < x) \ \wedge\ (z_2 < x),$$

in spirit akin to ζ. We leave it for the reader to check that ξ indeed expresses ODD-COLOR. It is readily observed that ξ can be cast as a $\mathbf{H}^{+}$-sentence, that is not in Stewart normal form. Yet by the Prenex normal form result, Proposition 4, we can extract the Henkin quantifiers from $\mathit{EVEN}(x_2)$, $\mathit{SUC}(x_1, x_2)$ and $\mathit{COLOR}(x_2)$, and obtain an equivalent formula of the form $\mathsf{H}_{(1)}\mathbf{x}\mathsf{H}_{(2)}\mathbf{y}\mathsf{H}_{(3)}\mathbf{z}\ldots$. By merging these, as we got from (3) to (4), we get an equivalent formula with one Henkin quantifier $\mathsf{H}_{(4)}\mathbf{u}\ldots$. The formula that results after replacing $\mathsf{H}_{(4)}\mathbf{u}\ldots$ in ξ is in Stewart normal form.

5 Concluding remarks

As we hoped to have shown, Henkin's idea has exciting manifestations in game theory, model theory, and computational complexity. Each of these manifestations shows a different face of the Henkin quantifier: interaction in the absence of full information, expressive power on formal structures, and algorithmic verification. Our results provide another instance when the disciplines at stake are strongly intertwined. Our Propositions 1 and 2 are cases in point. But admittedly, our approach was not highly systematic. We meandered from non-absentmindedness to absentmindedness, and from partially ordered quantification to Partial Information logic. Improving our understanding of the sparkling interface of logic and game theory is definitely worthwhile.

For instance what kind of game-theoretic underpinning can we give for $\mathbf{H}^{*}$? What does its game-theoretic semantics look like? And can it maybe inspire us to define an *interactive protocol* [16] kind of computing device that computes $\mathrm{P}_{\shortparallel}^{\mathrm{NP}}$? After all, interactive protocols are games with imperfect information.

An intriguing question was raised in [15] related to the finite model theory of Carnap's first-order modal logic $\mathbf{C}$. It is shown that even over finite structures, $\mathbf{C} < \mathbf{H}^{*}$, but what complexity class is actually captured by $\mathbf{C}$ is left as an open question. To this problem we may add the issue of developing a game-theoretic foundation for $\mathbf{C}$.

Finally we mention a game-theoretic gap that needs to be filled in the interest of logic and descriptive complexity. We used the computational result saying that every constant series of parallel queries can be reduced to one session of parallel queries [6]. The logical face of this theorem is the *flatness result*, holding that over linear ordered structures a $\mathbf{H}^{*}$-sentence of arbitrary Henkin depth has an equivalent $\mathbf{H}^{*}$-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* à la Ehrenfeucht and Fraïssé [9, 13]. 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 NP from coNP, for instance. Although considerable progress has been made along these lines [2, 12] the big questions from complexity theory are still unanswered. A fertile approach to prove non-expressibility results is to simplify model comparison games, in order to develop a library of intuitive tools for separating logics, cf. [1, 3]. Along these lines the flatness result concerning Henkin quantifiers may give rise to less complicated, but powerful, games.

References

[1] M. Ajtai and R. Fagin. Reachability is harder for directed than for undirected graphs. *Journal of Symbolic Logic*, 55(1):113–150, 1990.

[2] M. Ajtai, R. Fagin, and L. Stockmeyer. The closure of monadic NP. *Journal of Computer and System Sciences*, 60(3):660–716, 2000.

[3] S. Arora and R. Fagin. On winning strategies in Ehrenfeucht–Fraïssé games. *Theoretical Computer Science*, 174(1–2):97–121, 1997.

[4] J. Barwise. On branching quantifiers in English. *Journal of Philosophical Logic*, 8:47–80, 1979.

[5] A. Blass and Y. Gurevich. Henkin quantifiers and complete problems. *Annals of Pure and Applied Logic*, 32(1):1–16, 1986.

[6] S. R. Buss and L. Hay. On truth-table reducibility to SAT. *Information and Computation*, 91(1):86–102, 1991.

[7] A. Dawar, G. Gottlob, and L. Hella. Capturing relativized complexity classes without order. *Mathematical Logic Quarterly*, 44(1):109–122, 1998.

[8] F. Dechesne. *Game, Set, Maths: Formal investigations into logic with imperfect information*. PhD thesis, SOBU, Tilburg university and Technische Universiteit Eindhoven, 2005.

[9] A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.

[10] H. B. Enderton. Finite partially ordered quantifiers. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 16:393–397, 1970.

[11] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp, editor, *SIAM-AMS Proceedings, Complexity of Computation*, volume 7, pages 43–73, 1974.

[12] R. Fagin. Monadic generalized spectra. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 21:89–96, 1975.

[13] R. Fraïssé. Sur quelques classifications des systèmes de relations. Publications Scientifiques, Série A, 35–182 1, Université d'Alger, 1954.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Company, San Francisco, 1979.

[15] A. Gheerbrant and M. Mostowski. Recursive complexity of the Carnap first order modal logic C. *Mathematical Logic Quarterly*, 52(1):87–94, 2006.

[16] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[17] G. Gottlob. Relativized logspace and generalized quantifiers over finite ordered structures. *Journal of Symbolic Logic*, 62(2):545–574, 1997.

[18] L. Hella. Definability hierarchies of generalized quantifiers. *Annals of Pure and Applied Logic*, 43(3):235–271, 1989.

[19] L. Henkin. Some remarks on infinitely long formulas. In P. Bernays, editor, *Infinitistic Methods. Proceedings of the Symposium on Foundations of Mathematics*, pages 167–183, Oxford and Warsaw, 1961. Pergamon Press and PWN.

[20] J. Hintikka. Quantifiers vs. quantification theory. *Linguistic Inquiry*, 5:153–177, 1974.

[21] J. Hintikka. *Principles of mathematics revisited*. Cambridge University Press, 1996.

[22] J. Hintikka. Hyperclassical logic (a.k.a. IF logic) and its implications for logical theory. *Bulletin of Symbolic Logic*, 8(3):404–423, 2002.

[23] J. Hintikka and G. Sandu. Game-theoretical semantics. In J. F. A. K. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 361–481. North Holland, Amsterdam, 1997.

[24] W. Hodges. Formal aspects of compositionality. *Journal of Logic, Language and Information*, 10(1):7–28, 2001.

[25] T. Hyttinen and G. Sandu. Henkin quantifiers and the definability of truth. *Journal of Philosophical Logic*, 29(5):507–527, 2000.

[26] N. Immerman. *Descriptive Complexity*. Graduate texts in computer science. Springer, New York, 1999.

[27] M. Krynicki and M. Mostowski. Henkin quantifiers. In M. Krynicki, M. Mostowski, and L.W. Szczerba, editors, *Quantifiers: Logics, Models and Computation*, volume I of *Synthese library: studies in epistemology, logic, methodology, and philosophy of science*, pages 193–262. Kluwer Academic Publishers, The Netherlands, 1995.

[28] M. Krynicki and J. Väänänen. Henkin and function quantifiers. *Annals of Pure and Applied logic*, 43(3):273–292, 1989.

[29] M. Mostowski. Arithmetic with the Henkin quantifier and its generalizations. In F. Gaillard and D. Richard, editors, *Seminaire de Laboratoire Logique, Algorithmique et Informatique Clermontoise*, volume II, pages 1–25. 1991.

[30] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[31] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, Reading, Massachusetts, 1994.

[32] R. Parikh and J. Väänänen. Finite information logic. *Annals of Pure and Applied Logic*, 134(1):83–93, 2005.

[33] G. Sandu. On the logic of informational independence and its applications. *Journal of Philosophical Logic*, 22(1):29–60, 1993.

[34] G. Sandu and J. Väänänen. Partially ordered connectives. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 38(4):361–372, 1992.

[35] M. Sevenster and T. Tulenheimo. Partially ordered connectives and Σ_1^1 on finite models. In A. Beckmann, U. Berger, B. Löwe, and J. V. Tucker, editors, *Proceedings of the 2nd Computability in Europe Conference (CiE 2006), Logical Approaches to Computational Barriers*, volume 3988 of *LNCS*, pages 515–524. Springer, 2006.

[36] I. A. Stewart. Logical characterization of bounded query class I: Logspace oracle machines. *Fundamenta Informaticae*, 18:65–92, 1993.

[37] I. A. Stewart. Logical characterization of bounded query class II: Polynomial-time oracle machines. *Fundamenta Informaticae*, 18:93–105, 1993.

[38] J. Väänänen. On the semantics of informational independence. *Logic Journal of the IGPL*, 10(3):337–350, 2002.

[39] J. F. A. K. van Benthem. Games in dynamic-epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001.

[40] J. F. A. K. van Benthem. The epistemic logic behind IF games. In R. Auxier, editor, *Jaakko Hintikka*, Library of Living Philosophers. Carus Publishers, 2004.

[41] J. F. A. K. van Benthem. Logic and games, lecture notes. Draft version, unpublished.

[42] K. W. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51(1–2):53–80, 1987.

[43] K. W. Wagner. Bounded query classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.

[44] W. Walkoe. Finite partially-ordered quantification. *Journal of Symbolic Logic*, 35(4):535–555, 1970.