אוניברסיטת תל-אביב
הפקולטה למדעים מדויקים
ע״ש ריימונד ובברלי סאקלר
ביה״ס למדעי המחשב

לוגיקה של ידע לא מושלם

חיבור זה הוגש כחלק מהדרישות לקבלת התואר
״מוסמך אוניברסיטה״ – M.Sc. באוניברסיטת תל-אביב

על-ידי

דני ליבנה

העבודה הוכנה בהדרכתו של פרופ׳ ארנון אברון

חשוון, תשס״א

תודות:

ברצוני להודות לפרופ׳ ארנון אברון מאוניברסיטת ת״א על הנחיתו המקצועית ועזרתו הרבה בכתיבת עבודה זו.
כמו-כן רוב תודות לאורנה קלינשטיין על הערותיה החשובות, וכמובן לאשתי היקרה מיכל.

תקציר:

בעבודה זו נעסוק בהרחבה של לוגיקה מסדר ראשון המאפשרת שליטה על תכולת (scope) הכמתים בנוסחאות. בלוגיקה מסדר ראשון תכולת הכמתים "לכל" ו"קיים" תמיד מתחילה מהופעת הכמת בנוסחה, ומסומנת על ידי סוגריים. התחולה חייבת להיות רציפה ומקוננת, כלומר, בין כל שני כמתים תמיד יתקיים יחס של הכלה או זרות מוחלטת, כמו בדוגמאות הבאות:

$$\forall x(x = 0 \vee \exists y(y < x))$$

$$\forall x(x * x > x) \vee \exists y(y * y <= y)$$

בעבודה נסקור שתי גישות לשינוי הסדר הליניארי הקיים בין כמתים בלוגיקות סטנדרטיות. הראשונה פותחה על ידי ליאון הנקין (1961) ונקראת כמתי הנקין. השניה פותחה על ידי הינטיקה (1983) ומתבססת על סמנטיקה של משחקים על גבי נוסחאות. לשתי הגישות ישנם שונים בתחומים פילוסופיים ובמדעי המחשב, במיוחד בניסוח בעיות במודלים סופיים.

Tel Aviv University
Raymond and Beverly Sackler Faculty of Exact Sciences
School of Computer Sciences

Logic of Imperfect Information

Submitted as partial fulfillment of the requirement towards
the M.Sc. degree.

By

Dani Livne

This research work has been conducted under the supervision of
Prof. Arnon Avron

November, 2000

Acknowledgements

I am grateful to Prof. Arnon Avron for helpful discussions on the subject of this paper and to Prof. Gabriel Sandu from the university of Helsinki for his useful comments. I would also like to thank Orna Kleinstein for greatly improving the final presentation, and my wife Michal for supporting me doing this work.

Contents

1 Introduction

In this paper we examine the scope of quantifiers in first-order formulae. We consider extensions to first-order logic in which the ordinary scope of universals is changed. As a result we can control the information flow from a formula to its subformulae in a way that cannot be done in first order logic. Consider for example the following first order sentence ψ:

$$\forall x \exists y \forall z \exists w \phi(x, y, z, w)$$

Tarski's classical semantics for first order logic is recursively defined. ψ is true in a structure $\mathcal{M}$ for an assignment v ($(\mathcal{M}, v) \models \psi$) if and only if for all assignments v' such that for all $r \neq x$, $v'(r) = v(r)$:

$$(M, v') \models \exists y \forall z \exists w \ \phi(x, y, z, w)$$

It follows that for ψ to be true in $(\mathcal{M}, v)$, an imaginary test must be established. We must check all assignments v' (even an infinite number of them) which differ in the value given to x. The subformula $\exists y ...$, in turn, is true in $(\mathcal{M}, v')$, if and only if there exists an assignment v'' such that $v''(r) = v'(r)$ for all $r \neq y$, and

$$(M, v'') \models \forall z \exists w \ \phi(x, y, z, w)$$

The process continues by evaluating all possible values for z and so on.

The value of x was first established by an assignment v', and then we searched for a new assignment v'', that gives a value to y and equals v' on the value given to x. The value given to y *depends* on the value that was given to x, which is known at this stage.

The Skolem form theorem (extensions by definition) highlights this dependency because when applied the existentials are replaced with new function symbols, whose arguments are the previous universals in the formula.

This example demonstrates knowledge flow from a formula to its subformulae. The value of the variable w is a function of the values given to the previous quantified variables x, y and z (since the value for y is a function of x, it is actually only x and z). In this paper we try to control the flow of information using additional operators. The scope of quantified variables is altered using these operators.

Common logical systems always presuppose three assumptions about the scope of quantifiers:

1. A quantifier's scope begins immediately after its appearance.

2. A quantifier's scope is continuous. For any first order formula, we can change variables so that each two quantifiers are connected with different variable symbols (this will have no effect on the semantic meaning of the formula.) In such a formula the scope of each quantifier is continuous and without intervals.

3. Quantifier scopes never partially overlap. If the scope of two quantifiers overlap, then the scope of one of them must lie completely within the scope of the other, namely, they are nested.

These a priori assumptions are not justified in logic textbooks. There are in fact English sentences that do not meet those requirements. Hintikka [8] gives examples of such English sentences:

(a) "Some relative of each villager and some relative of each townsman hate each other."

(b) "Some book by every author is referred to in some essay by every critic."

(c) "The richer the country, the more powerful one of its officials."

In these sentences each existential is within the scope of a single universal. In the first sentence the villager's relative is chosen independently of the townsman, and the townsman's relative is chosen independently of the villager. The relatives, after being chosen, hate each other. In the third sentence, a situation in the world is described. We can independently select an official for each country so that given two countries x and y and their selected officials u and v - if x is richer than y, u is more powerful than v.

The simplest model that satisfies the first sentence is one where everyone hates each other. But "friendlier" models do exist, in which not every townsman hates every villager, and not every villager is a relative of all other villagers.

Are the sentences above expressible in first order logic? We use Skolemization to explain why they aren't. Consider sentence (a). We try to formalize it in first-order logic using $R(x, y, z, w)$ for "$villager(x) \wedge townsman(z) \wedge relative(x, y) \wedge relative(z, w) \wedge Hate(y, w) \wedge Hate(w, y)$".

(i) It is not $\forall x \exists y \forall z \exists w R(x, y, z, w)$ since the Skolem form of it is

$$\exists f \exists g \forall x \forall z R(x, f(x), z, g(x, z))$$

The fourth argument (the relative of each townsman) depends on x (the villager) and z, and not only on z (the townsman).

(ii) It is not $\forall z \forall x \exists y \exists w R(x, y, z, w)$ since its Skolem form is

$$\exists f \exists g \forall x \forall z R(x, f(x, z), z, g(x, z))$$

which is again not what we meant.

We actually need the following to be the 'Skolem form' of sentence (a):

$$(*) \quad \exists f \exists g \forall x \forall z R(x, f(x), z, g(z))$$

The functions f, g choose (independently) for each villager a relative, and for each townsman a relative such that $f(x)$ and $g(z)$ hate each other. There is no first-order sentence whose Skolem form is $(*)$, but using a new operator (called 'Henkin quantifier') will enable us to express such sentences. This operator carries a branching notation of the form:

$$\begin{pmatrix} \forall x & \exists y \\ \forall z & \exists w \end{pmatrix} R(x, y, z, w)$$

We will provide a formal definition for this quantifier, but for now we notice that y depends only on x (they appear on the same row) and w depends only on z.

Controlling the information flow of quantifiers in a sentence, allows to write formulae with more expressive power than first-order logic. We present two languages for dealing with this new concept, the language of Henkin quantifiers and the language of independence friendly logic (IF-logic). These languages have short and simple syntax and semantics. We describe the current state of affairs in this field and the major results achieved. We clarify and simplify many of the definitions, and supply portions missing in theorem proofs. We define Henkin quantifiers semantics without assuming the axiom of choice. A new theorem expressing Henkin quantifiers in IF-logic is proven, and new formulation of the induction scheme in both Henkin quantifiers and IF-logic is presented.

The first chapter deals with Henkin quantifiers. After a formal definition and examples, we examine model theoretical results. The last three sections deal with implementations of Henkin quantifiers:

- Henkin quantifiers in finite models
- Henkin quantifiers and the transitive closure operator
- Truth definition in a fragement of Henkin quantifiers logic

The second chapter deals with IF-logic. The semantics for this logic varies. First we see Hintikka's games semantics. We give a simple and coherent definition for it, and its relation to Henkin quantifiers. Then there is Hodges', Caicedo and Krynicki compositional game semantics. Their variant to IF-logic is much more complex, but it is compositional and allows using a new type of negation - game negation. This type of negation is one of the novelties of IF-logic. It has an intuitive semantic rule, and is stronger then the classical first-order negation in a sense that the class of structures that satisfies a given formula, is not necessarily the class of structures that satisfies its negation.

2 Henkin quantifiers

2.1 Syntax and Semantics

Logic of imperfect information first appeared in Leon Henkin [6] "Some Remarks on Infinitely Long Formulas". Dealing with a special kind of formulae of infinite length, Henkin asked what can possibly be the meaning of a first order formula with an infinite number of quantifiers, alternating an infinite number of times:

$$...(\exists v_5)(\forall v_4)(\exists v_3)(\forall v_2)(\exists v_1)(\phi v_1 v_2 v_3...) \tag{1}$$

A semantic meaning to such formulae can be given with the help of *Skolem functions*. Using Skolemization we can easily show which existential is in the scope of which universals. Formula (1) is true if and only if the following infinite second order formula is true:

$$(\exists g_1 g_3 g_5...)(\forall v_2 v_4 v_6...)\ \phi[g_1(v_2 v_4 v_6...), v_2, g_3(v_4 v_6 v_8...), v_4, g_5(v_6 v_8 v_{10}...)...]$$

Appling this "Skolem functions semantics" to *finite* first order formulae yield the Henkin quantifiers. The following definition is needed in order to alter the scope of universals in a formula.

Definition 2.1 *A* Henkin quantifier Q *is a pair* (A_Q, f_Q) [1]*, where* A_Q *is a finite set of first order quantifiers with distinct variables, and* f_Q *is a function from the set of existentials in* A_Q *to the power-set of the set of universals in* A_Q*.*

We assume that all Henkin quantifiers in a formula contain different variable signs. Let H be the closure of the language of first-order logic under its own formation rules together with the rule that $Q\phi$ is a formula whenever ϕ is a formula and Q is a Henkin quantifier whose variables do not occur bound in ϕ. Let H' be the set of all formulae of H of the form $Q\phi$ for a first order formula ϕ.

In a first order formula, the functions f_Q^i are derived from the scope relationships of the quantifiers. Each existential is mapped to the set of all universals within whose scope it is in. In a Henkin quantifier this does not have to be the case.

For some Henkin quantifiers (will be defined later as "standard Henkin quantifier") we use a branching notation in order to define (A_Q, f_Q). For example, the following Henkin quantifier:

$A_Q = \{\forall x, \exists y, \forall z, \exists u\}$, $f_Q(\exists y) = \{\forall x\}$, $f_Q(\exists u) = \{\forall z\}$
is written as:

$$\begin{pmatrix} \forall x & \exists y \\ \forall z & \exists u \end{pmatrix} \varphi(x, y, z, u) \tag{2}$$

[1] Walkoe [28] use a strong asymetric partial order instead of f_Q.

This notation graphically shows that y depends only on x, u depends only on z, and that y and u are chosen simultaneously. The following definition is a variant of Walkoe's [28] semantics for Henkin quantifiers.

Definition 2.2 (Second Order Semantics)
The Second order form (φ') of a formula φ of H is defined recursively going from outside in as follows:

1. *If $\varphi = (\psi \wedge \chi)$ then $\varphi' \triangleq (\psi') \wedge (\chi')$.*
2. *If $\varphi = \neg\psi$ then $\varphi' \triangleq \neg(\psi')$.*
3. *If $\varphi = \forall x\ \psi(x)$ then $\varphi' \triangleq \forall x\ (\psi(x)')$.*
4. *If $\varphi = Q\psi$ (Q - a Henkin quantifier) we associate a distinct set of function symbols $\{f_{x_i}^{Q\psi}\}$ for each existential quantifier $\exists x_i$ of Q. The function symbols are to be chosen so that distinct formulae are associated with disjoint sets, and they are to be "new" in the sense that none of them occur in any first-order formula. For each existential quantifier $\exists x_i$ of Q, let $\mathbf{s_{x_i}}$ be the sequence of all universal variables in $f_Q(\exists x_i)$. Then*

$$\varphi' \triangleq \exists f_{x_0}^{Q} ... \exists f_{x_n}^{Q} \forall v_0 ... \forall v_k\ (Subf_{x_0}^{Q}(\mathbf{s_{x_0}})/x_0) ... (Subf_{x_n}^{Q}(\mathbf{s_{x_n}})/x_n)\ (\psi)$$

where $A_Q = \{\forall v_0 ... \forall v_k, \exists x_0 .. \exists x_n\}$.

Truth. Given any formula φ of H, any model $\mathcal{M}$ for the vocabulary of φ, and an assignment v, we say that φ is *true* in $(\mathcal{M}, v)$ whenever $(\mathcal{M}, v) \models \varphi'$ (using the standard second order semantics).

Remark. In Walkoe's original paper [28] he used the term "Skolem form semantics" for the semantics of Henkin quantifiers. This name is less appropriate than "second order semantics" because a translation of a first-order formula in H according to definition (2.2), doesn't lead to the familiar Skolem form of first-order logic. For example

$$\forall x \exists y \forall z \exists w\ \psi(x, y, z, w)$$

consider each quantifier as a separate Henkin quantifier. Its second-order semantics is:

$$\forall x\ \exists f_1 \forall z \exists f_2\ \psi(x, f_1/y, z, f_2/w)$$

which is identical to the original formula and not its Skolem-form.

If, on the other hand, we consider $(\forall x \exists y \forall z \exists w)$ as a single compound Henkin quantifier, then using the above definition we would get the familiar Skolem form of a first-order formula

$$\exists f_1 \exists f_2 \forall x \forall z\ \psi(x, f_1(x)/y, f_2(x, z)/w)$$

We can generalize this by considering any two adjacent Henkin quantifiers and replacing them with a single one:

Proposition 2.3 ([28]) *For any two Henkin quantifiers Q_1 and Q_2 there is a Henkin quantifier Q_3 such that whenever a formula ϕ of H has a subformula of the form $(Q_1\ Q_2\chi)$, and ψ is the formula obtained by replacing any given occurrence of $(Q_1\ Q_2\chi)$ in ϕ by $Q_3\chi$, $\models \phi \leftrightarrow \psi$.*

The following example demonstrates how to prove this proposition. The formula

$$\begin{pmatrix} \forall x_1 & \exists y_1 \\ \forall x_2 & \exists y_2 \end{pmatrix} \begin{pmatrix} \forall x_3 & \exists y_3 \\ \forall x_4 & \exists y_4 \end{pmatrix} \varphi(x_1, ..., x_4, y_1, ..., y_4)$$

is true in (M, v) iff:

$$(M, v) \models \exists f_1 \exists f_2 \forall x_1 \forall x_2 \exists f_3 \exists f_4 \forall x_3 \forall x_4 \varphi(x_1, ..., x_4, f_1(x_1), ..., f_4(x_4))$$

Note that f_3 and f_4 depend on x_1 and x_2. We can change the order of quantifiers in this formula to get an equivalent formula:

$$\begin{aligned} (M, v) \quad \models \quad & \exists f_1 \exists f_2 \exists f_3' \exists f_4' \forall x_1 \forall x_2 \forall x_3 \forall x_4 \\ & \varphi(x_1, ..., x_4, f_1(x_1), f_2(x_2), f_3'(x_1, x_2, x_3), f_4'(x_1, x_2, x_4)) \end{aligned}$$

This formula is equivalent to the formula $Q\varphi$ with a single Henkin quantifier Q. Its domain is $\{\forall x_1, \forall x_2, \forall x_3, \forall x_4, \exists y_1, \exists y_2, \exists y_3, \exists y_4\}$ and

$$f_Q(\exists y_1) = \{\forall x_1\},$$

$$f_Q(\exists y_2) = \{\forall x_2\},$$

$$f_Q(\exists y_3) = \{\forall x_1, \forall x_2, \forall x_3\},$$

$$f_Q(\exists y_4) = \{\forall x_1, \forall x_2, \forall x_4\}$$

2.2 Henkin Quantifiers Examples

Example 1
The following formula

$$\begin{pmatrix} \forall x & \exists y \\ \forall z & \exists u \end{pmatrix} \varphi(x, y, z, u)$$

is true in (M, v) if the following second order formula is:

$$\exists f_1 \exists f_2 \forall x \forall z S(x, [\frac{f_1(x)}{y}], z, [\frac{f_2(z)}{u}])$$

i.e., there are two functions f_1, f_2 such that for all x and z, S holds for $x, f_1(x), z, f_2(z)$. The second term depends only on x, and the fourth - only

on z. Thus the branching quantifier actually means that for all x there exists y (depending only on x), and at the same time, for all z there exists u (depending only on z) such that S holds. The values of y and u are given simultaneously, and are independent of each other (*independent quantifiers*).

Example 2

Consider the following first-order formula:

$$\neg\forall x\exists y P(x,y)$$

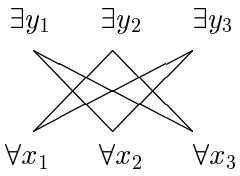
According to De-Morgan laws, it is equivalent to $\exists x\forall y\neg P(x,y)$, but if we first apply the second-order translation (taking $\forall x\exists y$ as a Henkin quantifier), we get $\forall f\exists x\neg P(x,f(x))$. The two resulting formulae are equivalent in second-order logic.

Example 3

If $A_Q = \{\forall x_1, \forall x_2, \forall x_3, \exists y_1, \exists y_2, \exists y_3\}$ and f_Q is given by

$$f_Q(\exists y_j) = \{\forall x_i\}_{i\neq j}$$

as illustrated in the following figure:



Then $Q\phi(x_1,x_2,x_3,y_1,y_2,y_3)$ means that there are binary functions Y_1, Y_2, Y_3 such that the following formula holds

$$\forall x_1\forall x_2\forall x_3\phi(x_1,x_2,x_3,Y_1(x_2,x_3),Y_2(x_1,x_3),Y_3(x_1,x_2)) \tag{3}$$

Example 4

The language H allows the nesting of Henkin quantifiers and negation. For example:

$$\begin{pmatrix} \forall x_1 & \exists y_1 \\ \forall x_2 & \exists y_2 \end{pmatrix} \neg \begin{pmatrix} \forall x_3 & \exists y_3 \\ \forall x_4 & \exists y_4 \end{pmatrix} \phi(x_1,...,x_4,y_1,...,y_4)$$

which means

$$\exists f_1\exists f_2\forall x_1\forall x_2\forall f_3\forall f_4\exists x_3\exists x_4\neg\phi(x_1,...,x_4,f_1(x_1),...,f_4(x_4))$$

Example 5 [16]

Well ordering [2] is not definable in a first-order language (Lindström [17]). Its negation is definable in H' using the well known fact that a partially ordered set $\langle A, \leq \rangle$ is well ordered if and only if there is no function $f : A \to A$ such that

1. For all $a \in A : f(a) \leq a$, and
2. $\{a \in A | f(a) < a\}$ is non-empty, and
3. For all $a \in A$ if $f(a) < a$ then $f(f(a)) < f(a)$.

We use the common notation Σ_1^1 for the class of second order sentences of the sort:

$$\exists\ f_1 ... f_n\ \varphi$$

Where f_i are second order variables and all the quantifiers in φ are of a first-order. Π_1^1 is defined similarly using $\forall\ f_1 ... f_n$.
The negated Σ_1^1 formula that meets provisions 1 - 3 is:

$$\neg\exists f \Big[(\forall x_1\ f(x_1) \leq x_1) \wedge (\exists x_3\ f(x_3) < x_3) \wedge (\forall x_2\ f(x_2) < x_2 \to f(f(x_2)) < f(x_2))\Big]$$

Later on we see that every Σ_1^1 formula has a H' equivalent (theorem 2.8) which means that well ordering is equivalent to a negated H' sentence.

Example 6 [7]

Henkin quantifiers can be used to formulate the property of a function to be uniformly differentiable in the interval $[x_1, x_2]$.

f is differentiable in each point of the interval $[x_1, x_2]$ if and only if

$$\forall x\ \exists y\ \forall \varepsilon > 0\ \exists \delta\ \forall z \Big[(x_1 < x < x_2\ \wedge\ |z| < |\delta|) \to (|\tfrac{f(x+z)-f(x)}{z} - y| < \varepsilon)\Big]$$

The function is uniformly differentiable if and only if the same condition holds, but with $\exists\delta$ being independent of $\forall x$, i.e., using the Henkin quantifier

$$\begin{pmatrix} \forall x & \exists y \\ \forall \varepsilon & \exists \delta \end{pmatrix}$$

In this special case, a first order equivalent can in fact be found. Since the variable y in this sentence has a concrete meaning i.e., $f'(x)$, we can replace $\forall x \exists y ...$ with $\forall x \exists y! ...$ in the formula above. The property of a function which is uniformly differentiable can then be expressed by the conjunction of the following two formulae:

1. f is differentiable in each point of the interval.

[2] A set is well ordered if it is partially ordered and any of its non-empty subsets contains a first element.

2. $\forall \varepsilon > 0\ \exists \delta\ \forall x\ \forall y\ \forall z$

$$\Big[\text{“}y = f'(x)\text{”} \rightarrow (x_1 < x < x_2 \wedge\ |z| < \delta \rightarrow (|\tfrac{f(x+z)-f(x)}{z} - y| < \varepsilon))\Big]$$

where “$y = f'(x)$” is a short form of

$$\forall \epsilon_1 > 0\ \exists \delta_1 \Big(\forall z_1\ \Big[|z_1| < |\delta_1| \wedge (x_1 < x < x_2) \rightarrow |\tfrac{f(x+z_1)-f(x)}{z} - y| < \epsilon_1\Big]\Big)$$

This assures that y keeps its original role of being the single value $f'(x)$ and so it is independent of the previous quantifiers.

2.3 Second Order Semantics - Not Assuming the Axiom of Choice

Walkoe's Definition (2.2) tacitly assumes the axiom of choice. [3] Indeed without this axiom, the fact that “$\forall x \exists y ...$” does not necessarily imply that a corresponding function f exists. It implies only that there is a non-empty set of y's for each x, but not that a function that *chooses* one such y for each x in a systematic way exists.

There is an alternative way to define the semantics of a Henkin quantifier, which does not presuppose the axiom of choice. As far as we know, this definition is not mentioned in the literature.

Definition 2.4 (Second Order Semantics - Without AC)
We replace item 4 *in definition (2.2) with the following:*

> *If* $\varphi = Q\psi$ *(Q - a Henkin quantifier) we associate a set of function symbols, with distinct function symbol* $F_{x_i}^{Q\psi}$ *for each existential quantifier* $\exists x_i$ *of* Q*. The function symbols are to be chosen so that distinct formulae are associated with disjoint sets, and they are to be “new” in the sense that none of them occurs in any first-order formula. For each existential quantifier* $\exists x_i$ *of* Q*, let* $\mathbf{s_{x_i}}$ *be the sequence of all universal variables in* $f_Q(\exists x_i)$*. Then*
>
> $\varphi' \triangleq \exists F_{x_0}^{Q} ... \exists F_{x_n}^{Q} \forall v_0 ... \forall v_k [(\bigwedge_i F_{x_i}^{Q\phi}(\mathbf{s_{x_i}}) \neq \emptyset) \wedge$
>
> $\forall y_0, ..., y_n :\ \ (y_0 \in F_{x_0}^{Q\phi}(\mathbf{s_{x_0}}) \wedge ... y_n \in F_{x_n}^{Q\phi}(\mathbf{s_{x_n}})) \rightarrow \phi(\frac{y_0}{x_0}, \frac{y_1}{x_1} ...)]$
>
> $A_Q = \{\forall v_0, ..., \forall v_k, \exists x_0, ..., \exists x_n\}$.
> F_{x_i} *- functions of third order language.* [4]

[3] We shall see in the next chapter that Hintikka's game semantics for IF-logic relies on the axiom of choice as well.

[4] F_{x_i} is a function whose domain is the set of tuples $\mathbf{s_{x_i}}$ and its target is a partial set of the domain, $F_{x_i} : D^n \rightarrow P(D)$. A partial set is a function $f : D \rightarrow Bool$ and by UnCurrying we can in fact translate F_{x_i} to a function of a second order language.

The difference is in the intended range of the interpreted functions, which is now $\mathcal{P}(\mathcal{D})$ rather than $\mathcal{D}$ (the domain under discourse). The interpretations of the functions $F_{x_i}^{Q\phi}$ are well defined (without AC) since they assign to every sequence of objects from the domain a set rather than choosing one element. This definition is equivalent to the original one if we assume the axiom of choice.

2.4 Model Theoretical Results

Henkin [6] poses the questions whether the simplest kind of Henkin quantifier is expressible in first-order logic and, if not, whether first-order logic enriched by the following Henkin quantifier is recursively axiomatizable:

$$\begin{pmatrix} \forall x & \exists v \\ \forall y & \exists u \end{pmatrix} \varphi(x, y, v, u) \tag{4}$$

Ehrenfeucht [6] answered both questions negatively by showing how to express the quantifier "for infinitely many x's" in this enriched logic.

Theorem 2.5 (Ehrenfeucht) *In a first order predicate calculus enriched with the most simple branching quantifier (4) one can define a quantifier R such that $R(x)\varphi(x)$ is true if and only if there are infinitely many values of x for which $\varphi(x)$ is true.*

proof
For any formula φ, let $(Sxy; vw)\varphi$ be the formula:

$$\begin{pmatrix} \forall x & \exists v \\ \forall y & \exists w \end{pmatrix} \Big[\Big((x = y) \leftrightarrow (v = w)\Big) \wedge \varphi\Big]$$

Then the following three formulae are logically equivalent:

(i) $(Sxy; vw)\varphi(x, y, v, w)$

(ii) $(\exists g, h)(\forall x, y)\Big[\big(x = y \leftrightarrow g(x) = h(y)\big) \wedge \varphi(x, y, g(x), h(y))\Big]$

(iii) $(\exists g, h)\Big[(\forall x, y)\Big(x = y \rightarrow g(x) = h(y)\Big) \wedge$
$(\forall x, y)\Big(g(x) = h(y) \rightarrow x = y\Big) \wedge (\forall x, y)\varphi(x, y, g(x), h(y))\Big]$

Furthermore, $(\forall x, y)\Big(x = y \rightarrow g(x) = h(y)\Big)$ is equivalent to $(\forall x)\Big(g(x) = h(x)\Big)$ and hence to $g = h$. Thus $(Sxy; vw)\varphi(x, y, v, w)$ is equivalent to

$$(\exists g)\Big[(\forall x, y)(g(x) = g(y) \rightarrow x = y) \wedge (\forall x, y)\varphi(x, y, g(x), g(y))\Big]$$

i.e. to

$$(\exists g)\Big[g \ is \ one-one \ \wedge (\forall x,y)\varphi(x,y,g(x),g(y))\Big]$$

If we define for every formula ψ, $(Tx)\psi(x)$ as

$$(\exists z)\Big[\psi(z)\wedge(Sxy;vw)\big[\psi(x)\rightarrow\big(\psi(v)\wedge v\neq z\big)\big]\Big]$$

we see that $(Tx)\psi(x)$ is equivalent to

$$(\exists z)\Bigg[\psi(z)\wedge(\exists g)\Big[g \ is \ one-one\wedge(\forall x)\Big(\psi(x)\rightarrow\Big(\psi(g(x))\wedge g(x)\neq z\Big)\Big)\Big]\Bigg]$$

thus $(Tx)\psi(x)$ is true if and only if there is a one-one function which maps the set of individuals satisfying ψ into one of its proper subsets. Hence $T = R$, and we have shown how to define R in terms of the simplest Henkin quantifier. □

The quantifier R is not definable in first order logic. Moreover, it follows from this theorem that the system of valid formulae of H' is not axiomatizable. This is due to Motowski [18] who shows that predicate calculus enriched by the quantifier R is not axiomatizable (this set of formulas is not even recursively enumerable).

Given a formula in H', its semantics is given by an existential second order formula, namele Σ^1_1. Walkoe [28] proves that the opposite is true as well, i.e., we can translate any Σ^1_1 formula to an equivalent one in H'. We need the following definitions:

Definition 2.6 *The Standard Henkin Quantifier is a Henkin quantifier that satisfies the following:*
For each universal $\forall x_i$ in A_Q there is no more than one existential $\exists y_i$ in A_Q that satisfy $\forall x_i \in f_Q(\exists y_i)$.

Using the branching notation we denote the standard Henkin quantifier by:

$$\begin{pmatrix} \forall x_1^1 \ ... & \forall x_{n_1}^1 & \exists a_1 \\ \forall x_1^2 \ ... & \forall x_{n_2}^2 & \exists a_2 \\ \vdots & & \\ \forall x_1^m \ ... & \forall x_{n_m}^m & \exists a_m \end{pmatrix}$$

where x_i^j - different variables.

Definition 2.7 (Representability) *Let L_P be a first-order language with identity, a k-place predicate symbol P and no other function, predicate or constant symbols other than P. Let Q_1 be a Henkin quantifier with exactly k variables $v_1...v_k$ and let Q_2 be an arbitrary Henkin quantifier. We say that Q_1 is representable by Q_2, if there is a quantifier free formula ϕ of L_P such that $Q_2\phi$ is a sentence and*

$$\models Q_1 P(v_1, ..., v_k) \leftrightarrow Q_2\phi.$$

Theorem 2.8 (Walkoe)
1) In any second order language L, every Σ^1_1 formula is equivalent [5] *to a formula in H', i.e. a formula $Q\psi$ where ψ is a quantifier free first-order formula of L, and Q is a standard Henkin quantifier.*

As a consequence we get:
2) Normal form theorem:
Over infinite domains, all Henkin quantifiers are representable by a Henkin quantifier of the form:

$$Q_2 = \begin{pmatrix} \forall x_1 \ \forall y_1 \exists a_1 \\ \vdots \\ \forall x_n \ \forall y_n \exists a_n \end{pmatrix}$$

or a Henkin quantifier of the form [6] *:*

$$Q_3 = \begin{pmatrix} \forall x_1 \ ... \ \forall x_i \ \exists a_1 \ ... \ \exists a_j \\ \forall y_1 \ ... \ \forall y_i \ \ \exists b_1 \ ... \ \exists b_j \end{pmatrix}$$

proof
We prove 1) by providing an effective procedure to translate any Σ^1_1 formula ϕ to an H' formula $Q\psi$.
Let ϕ be
$\exists f_0 \exists f_1 \forall \mu_0 ... \forall \mu_4$
$$P(\mu_0, .., \mu_4, f_0(f_1(\mu_2), \mu_4), f_0(\mu_2, \mu_4), f_1(f_1(\mu_0)), f_1(\mu_1), f_1(\mu_2))$$

we list the terms of ϕ (other than variables) without placing any term before any of its subterms:

$$f_0(\mu_2, \mu_4) \ , \ f_1(\mu_1) \ , \ f_1(\mu_2) \ , \ f_1(\mu_0) \ , \ f_0(f_1(\mu_2) \ , \mu_4) \ , \ f_1(f_1(\mu_0))$$

[5] The second order form of the Henkin formula and the Σ^1_1 formula are logically equivalent. We consider only sentences in the proof. If ϕ has free variables then the two formulae 'are equivalent' with refer to any given assignment.

[6] Michal Krynicki [15] improves Walkoe's result by showing that over infinite domains every Henkin quantifier is equivalent to the following Henkin quantifier:

$$\begin{pmatrix} \forall x_1 \ ... \ \forall x_n \exists y \\ \forall z_1 \ ... \ \forall z_k \ \exists t \end{pmatrix}$$

for some $n, k \in \omega$.

Define

$$Q = \begin{pmatrix} \forall\mu_2^0 & \forall\mu_4^0 & \exists w_0^0 \\ & \forall\mu_1^1 & \exists w_1^1 \\ & \forall\mu_2^2 & \exists w_1^2 \\ & \forall\mu_0^3 & \exists w_1^3 \\ \forall x_2^4 & \forall\mu_4^4 & \exists w_0^4 \\ & \forall x_3^5 & \exists w_1^5 \\ \forall\mu_0^0 & \forall\mu_1^0 & \forall\mu_3^0 \end{pmatrix}$$

and let ψ be

$$\begin{aligned} &(\mu_2^0 = x_2^4 \wedge \mu_4^0 = \mu_4^4 \rightarrow w_0^0 = w_0^4) \\ &\wedge(\mu_1^1 = \mu_2^2 \rightarrow w_1^1 = w_1^2) \\ &\wedge(\mu_1^1 = \mu_0^3 \rightarrow w_1^1 = w_1^3) \\ &\wedge(\mu_1^1 = x_3^5 \rightarrow w_1^1 = w_1^5) \\ &\wedge[(\mu_1^0 = \mu_1^1 \wedge \mu_2^0 = \mu_2^2 \wedge \mu_0^0 = \mu_0^3 \wedge \mu_4^0 = \mu_4^4 \wedge x_2^4 = w_1^2 \wedge x_3^5 = w_1^3) \\ &\qquad\qquad \rightarrow P(\mu_0^0, ..., \mu_4^0, w_0^4, w_0^0, w_1^5, w_1^1, w_1^2)] \end{aligned}$$

μ_j^i corresponds to variable μ_j in term i in the list above.
w_j^i corresponds to function f_j in term i in the list above.
x_j^i corresponds to a subterm of the term i in the list above. This subterm is term j in the list above.

Then $Q\psi$ is equivalent to

$$\begin{aligned} &\exists f_0^0 \exists f_1^1 \exists f_1^2 \exists f_1^3 \exists f_0^4 \exists f_1^5 \\ &\quad [\forall\mu_2^0 \forall x_2^4 \forall\mu_4^0 \forall\mu_4^4 \ \ (\mu_2^0 = x_2^4 \wedge \mu_4^0 = \mu_4^4) \rightarrow f_0^0(\mu_2^0, \mu_4^0) = f_0^4(x_2^4, \mu_4^4)] \ \ \wedge \\ &\quad \forall\mu_1^1 \forall\mu_2^2 \ \ (\mu_1^1 = \mu_2^2 \rightarrow f_1^1(\mu_1^1) = f_1^2(\mu_2^2)) \ \ \wedge \\ &\quad \forall\mu_1^1 \forall\mu_0^3 \ \ (\mu_1^1 = \mu_0^3 \rightarrow f_1^1(\mu_1^1) = f_1^3(\mu_0^3)) \ \ \wedge \\ &\quad \forall\mu_1^1 \forall x_3^5 \ \ (\mu_1^1 = x_3^5 \rightarrow f_1^1(\mu_1^1) = f_1^5(x_3^5)) \ \ \wedge \\ &\quad \forall\mu_1^0 \forall\mu_1^1 \forall\mu_2^0 \forall\mu_2^2 \forall\mu_0^0 \forall\mu_0^3 \forall\mu_4^0 \forall\mu_4^4 \forall x_2^4 \forall x_3^5 \forall\mu_3^0 \\ &\quad [(\mu_1^0 = \mu_1^1 \wedge \mu_2^0 = \mu_2^2 \wedge \mu_0^0 = \mu_0^3 \wedge \mu_4^0 = \mu_4^4 \wedge x_2^4 = f_1^2(\mu_2^2) \wedge x_3^5 = f_1^3(\mu_0^3)) \\ &\qquad\qquad \rightarrow P(\mu_0^0, ..., \mu_4^0, f_0^4(x_2^4, \mu_4^4), f_0^0(\mu_2^0, \mu_4^0), f_1^5(x_3^5), f_1^1(\mu_1^1), f_1^2(\mu_2^2))] \end{aligned}$$

This reduces to

$$\begin{aligned} &\exists f_0^0 \exists f_1^1 \forall\mu_0^0 ... \forall\mu_4^0 \\ &\qquad P(\mu_0^0, ..., \mu_4^0, f_0^0(f_1^1(\mu_2^0), \mu_4^0), f_0^0(\mu_2^0, \mu_4^0), f_1^1(f_1^1(\mu_0^0)), f_1^1(\mu_1^0), f_1^1(\mu_2^0)) \end{aligned}$$

which is clearly equivalent to ϕ. □

To summarize the procedure of translating a Σ_1^1 formula to a Henkin formula:

1. Move the Σ_1^1 formula to a prenex normal form.
2. Start with a new (empty) Henkin quantifier Q.
3. Line up in a list, without repetitions, the complex terms in the formula (terms with second order variables). Do not place any term before any of its subterms. Lets assume there are n terms.

4. Let term i in the list ($i = 1...n$) be

$$f_x(v_m, f_y(...))$$

where v_m is a variable and $f_y(...)$ is a complex term (that appears earlier on the list). The term consists of only two parameters for the simplicity of the example.

Add the following row to the standard Henkin quantifier Q:

$$\forall x_k^i \ \forall \mu_m^i \ \exists w_x^i$$

where k is the place of the term $f_y(...)$ in the list.

Construction of the formula ψ:

5. For each two rows in Q with the same subscript x in w_x^i (they correspond to terms with the same function) add a condition of the following form:

$$\mu_n^m = \mu_k^j(x_k^j) \wedge x_l^m = x_r^j \ \rightarrow w_x^m = w_x^j$$

6. The formula proceeds by equating parameters. if two terms in the list have the same variables, we add equations of the form:

$$\mu_j^i = \mu_m^n$$

and if few of their subterms are the same, we add equations of the form $x_j^i = x_j^k$ or of the form $x_j^i = w_x^j$.

7. We add a row of the form $(\forall \mu_0^0 \forall \mu_1^0 ...)$ which include all the variables in $P(v_0, v_1, ...)$ that haven't appeared yet.

8. The rest of the formula is the original formula with the new variables w_x^i and μ_j^i substituted.

Proof of the normal form theorem
By definition, every formula in H' is equivalent to a Σ_1^1 formula in a language with the same signature. Hence it is a consequence of part 1) of this theorem that every Henkin quantifier is representable by a standard Henkin quantifier. This means that we may assume without loss of generality that Q_1 is:

$$Q_1 = \begin{pmatrix} \forall x_1^1 \ ... \ \forall x_{n_1}^1 & \exists a_1 \\ \forall x_1^2 \ ... \ \forall x_{n_2}^2 & \exists a_2 \\ \vdots & \\ \forall x_1^m \ ... \ \forall x_{n_m}^m & \exists a_m \end{pmatrix}$$

for some $m, n_1, ..., \ n_m$.

The second order form of

$$Q_1 P(x_1^1, ..., x_{n_m}^m, a_1, ..., a_m)$$

is
$\exists f_1 ... \exists f_m \forall x_1^1 ... \forall x_{n_m}^m P(x_1^1, ..., x_{n_m}^m, f_1(x_1^1, ..., x_{n_1}^1), ..., f_m(x_1^m, ..., x_{n_m}^m))$

Suppose $n_1 > 2$. We use the fact that

$$\begin{array}{l} \exists f_1 ... \exists f_m \forall x_1^1 ... \forall x_{n_m}^m P(x_1^1, ..., x_{n_m}^m, f_1(x_1^1, ..., x_{n_1}^1), ..., f_m(x_1^m, ..., x_{n_m}^m)) \\ \quad \leftrightarrow \\ \exists f_1 ... \exists f_{m+1} \forall x_1^1 ... \forall x_{n_m}^m P(x_1^1, ..., x_{n_m}^m, f_{m+1}(f_1(x_1^1, ..., x_{n_1-1}^1), x_{n_1}^1), \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad f_2(x_1^2, ..., x_{n_2}^2), ..., f_m(x_1^m, ..., x_{n_m}^m)) \end{array}$$

to obtain

$$\models Q_1 P(x_1^1, ..., x_{n_m}^m, a_1, ..., a_m) \leftrightarrow$$
$$\left(\begin{array}{llll} \forall x_1^1 & ... & \forall x_{n_1-1}^1 & \exists a_1 \\ \vdots & & & \\ \forall x_1^m & ... & \forall x_{n_m}^m & \exists a_m \\ \forall x_1^{m+1} & & \forall x_{n_1}^1 & \exists a_{m+1} \end{array} \right) (x_1^{m+1} = a_1 \rightarrow P(x_1^1, ..., x_{n_m}^m, a_{m+1}, a_2, ..., a_m))$$

By repeated application of this procedure, and by adding dummy variables when necessary, we eventually obtain a standard quantifier Q_2 with no more than two universals in any row, such that Q_1 is representable by Q_2. [7]

That Q_1 is representable by a quantifier of the form of Q_3 follows from the fact that Q_1 is representable by a quantifier of the form of Q_2 and the following lemma:

Lemma 2.9 *The Henkin quantifier Q_2 is representable by the Henkin quantifier Q_3.*

The lemma follows from the following equivalence:

$$\models \left(\begin{array}{lll} \forall x_1 & \forall y_1 & \exists a_1 \\ & \vdots & \\ \forall x_n & \forall y_n & \exists a_n \end{array} \right) P(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{a}}) \leftrightarrow$$
$$\left(\begin{array}{llll} \forall x_1 & ... \forall x_n & \forall y_1 & ... \forall y_n \\ \forall z_1 & ... \forall z_n & \forall w_1 & ... \forall w_n \end{array} \begin{array}{ll} \exists a_1 & ... \exists a_n \\ \exists b_1 & ... \exists b_n \end{array} \right) (x_1 = z_1 \wedge y_1 = w_1 \rightarrow a_1 = b_1) \wedge$$
$$\vdots$$
$$(x_n = z_n \wedge y_n = w_n \rightarrow a_n = b_n) \wedge$$
$$P(\bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{a}})$$

[7] In his paper, Walkoe doesn't mention that the normal form theorem is applicable only to infinite domains. The problem with finite domains is that decomposing the function f_1 to two functions f_{m+1} and f_1 is not possible. A simple example in a domain with only two elements {0,1} and a three place function $f_1(x, y, z)$ will show that.

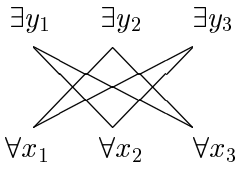
The second order form of the left-hand side is
(1) $\exists \bar{\mathbf{f}} \forall \bar{\mathbf{x}} \forall \bar{\mathbf{y}} P(\bar{\mathbf{x}}, \bar{\mathbf{y}}, f_1(x_1, y_1), ..., f_n(x_n, y_n))$

The second order form of the right-hand side is
(2) $\exists \bar{\mathbf{h}} \exists \bar{\mathbf{g}} \forall \bar{\mathbf{x}} \forall \bar{\mathbf{y}} \forall \bar{\mathbf{z}} \forall \bar{\mathbf{w}}$
$[(x_1 = z_1 \wedge y_1 = w_1 \rightarrow h_1(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = g_1(\bar{\mathbf{z}}, \bar{\mathbf{w}})) \wedge$
$\vdots$
$(x_n = z_n \wedge y_n = w_n \rightarrow h_n(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = g_n(\bar{\mathbf{z}}, \bar{\mathbf{w}})) \wedge$
$P(\bar{\mathbf{x}}, \bar{\mathbf{y}}, h_1(\bar{\mathbf{x}}, \bar{\mathbf{y}}), ..., h_n(\bar{\mathbf{x}}, \bar{\mathbf{y}}))$

(1) $\Rightarrow$ (2) is shown by defining $h_i(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = g_i(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = f_i(x_i, y_i)$.

(2) $\Rightarrow$ (1) is shown by defining $f_i(x_i, y_i) = h_i(x_i, ..., x_i, y_i, ..., y_i)$. Note that $h_i(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ and $g_i(\bar{\mathbf{z}}, \bar{\mathbf{w}})$ depend only on the i'th component (a result of the i'th conjunct in (2)). □

Example. The non-standard Henkin quantifier we had on page 9 which was illustrated by



is representable by the following <u>standard</u> Henkin quantifier:

$$\begin{pmatrix} \forall x_2 & \forall x_3 & \exists y_1 \\ \forall x_1 & \forall x_3' & \exists y_2 \\ \forall x_1' & \forall x_2' & \exists y_3 \end{pmatrix} [x_1 = x_1' \wedge x_2 = x_2' \wedge x_3 = x_3' \rightarrow \phi(x_1, x_2, x_2, y_1, y_2, y_3)]$$

Corollary 2.10 *The induction scheme of first-order logic is equivalent to a negated H' formula.*

proof
The second order version of the axiom of mathematical induction is the following Π_1^1 sentence:

$$\forall X \Big[\Big(X(0) \wedge \forall y (X(y) \rightarrow X(y+1) \Big) \rightarrow \forall z X(z) \Big]$$

This axiom is equivalent to the negation of the following Σ_1^1 sentence:

$$\exists X \Big[X(0) \wedge \big(\forall y (X(y) \rightarrow X(y+1)) \big) \wedge \exists z \neg X(z) \Big] \Leftrightarrow$$

$$\exists X \forall y \Big[X(0) \wedge (X(y) \rightarrow X(y+1)) \wedge \exists z \neg X(z) \Big] \Leftrightarrow$$

$$\exists X \exists z \forall y \forall x \Big[(x = 0 \to X(x)) \wedge (X(y) \wedge (x = y + 1) \to X(x)) \wedge (x = z \to \neg X(x)) \Big]$$

Using Walkoe method (theorem 2.8) for translating a Σ_1^1 formula to Henkin formula we get:

$$\neg \begin{pmatrix} \forall x \ \exists w_1 \\ \forall y \ \exists w_2 \\ \quad \exists z \end{pmatrix} \begin{array}{l} (w_1 = 0 \vee w_1 = 1) \wedge (w_2 = 0 \vee w_2 = 1) \wedge \\ (x = y \to w_1 = w_2) \wedge (x = 0 \to w_1 = 1) \wedge \\ (w_2 = 1 \wedge x = y + 1 \to w_1 = 1) \wedge (x = z \to w_1 = 0) \end{array}$$

□

This is a special kind of a Henkin quantifier, where the interpreted functions can have only Boolean values. It is called a *narrow Henkin quantifier.* We return to this kind of quantifier later in definition 2.18.

Walkoe's theorem shows that Q_2 and Q_3 are sufficient in infinite domains, in a sense that they can represent any other Henkin quantifier. If our language contains the language of Peano arithmetic we can reduce Q_2 to

$$\begin{pmatrix} \forall x \ \exists z \\ \forall y \ \exists w \end{pmatrix} \varphi(..., x_i, ..., y_i, ..., z, w)$$

due to the fact that Peano arithmetic is rich enough to code finite sequences inside the language itself.

We learn from the normal form theorem that H' is equivalent to Σ_1^1. The next question is how high the language H is in the hierarchy theory [8] Enderton [3] shows that it is contained in Σ_2^1 and Π_2^1.

Definition 2.11 *Using the symbols Σ_n^m and Π_n^m of hierarchy theory, we classify the ordinary quantifier-prefixes and sentences of higher-order logic as follows. The* character *of any universal quantifier is $\forall$; the* character *of any existential quantifier is $\exists$. A sequence of quantifiers is a Σ_n^m prefix if the highest order of any of its variables is $m+1$, its first quantifier is existential, and it has n quantifiers including the first which are of adifferent character from their immediate predecessors. A Π_n^m prefix is the dual of a Σ_n^m prefix. A second-order sentence is a Σ_n^m sentence (respectively, a Π_n^m sentence) if it is of the form $Q\phi$ where Q is a Σ_n^m prefix (Π_n^m prefix) and ϕ is a formula all of whose quantified variables have order $\leq m$.*

[8] H is the language that contains nested Henkin quantifiers together with negation.

Theorem 2.12 (Enderton) *Given any formula $\varphi \in H$ we can effectively find a Σ^1_2 and a Π^1_2 formula, both equivalent to φ.*

The theorem follows from the following lemma by induction.

Lemma 2.13 *Given a Π^1_2 (Σ^1_2) formula ψ and a Henkin quantifier Q, the formula $Q\psi$ yields another formula which is again, up to logical equivalence, a Π^1_2 (Σ^1_2).*

proof
We use the following principle [1]:

$$\exists F_1 ... F_n \forall x_1 ... x_m \Big[\forall G\ \exists H\ \ \phi(\bar{x}, F_1(\bar{x}_1), ..., F_n(\bar{x}_n), G, H)\Big] \leftrightarrow$$

$$\forall G \exists F_1 ... F_n\ \exists H \forall x_1 ... x_m\ \ \phi(\bar{x}, F_1(\bar{x}_1), ... F_n(\bar{x}_n), \lambda t. G(\bar{x}, F_1(\bar{x}_1), ..., F_n(\bar{x}_n), t), H)$$

$\bar{x}_i$ - a sublist of $x_1, ..., x_n$,
ϕ - a first-order formula.
The left-hand side is a Π^1_2 formula prefixed with second order existentials, which we have shown to be equivalent to a standard Henkin quantifier. The right-hand side is again a Π^1_2 formula.

We show the above equivalence for the case $n = m = 1$. The general case then follows by repeated application of the simple case. We need to show that

$$\exists F \forall x \forall G \exists H \phi(x, F(x), G, H) \leftrightarrow \forall G' \exists F \exists H \forall x \phi(x, F(x), \lambda t. G'(x, F(x), t), H)$$

$\rightarrow$
If there is F_1 such that for all x and for all the functions G, $\exists H \phi$ is true, then for all the functions G' we can find a corresponding F (it is F_1) such that for all x, $\exists H \phi(x, F(x), \lambda t. G'(x, F(x), t)$ holds (note that $\lambda t. G'(x, F(x), t)$ is a special case of a function G). We now use the second order equivalence

$$\forall x \exists H \Phi(x, H) \leftrightarrow \exists H' \forall x \Phi(x, H(x))$$

to change the order of x and H.
$\leftarrow$
It is more convenient to show the negated version of this implication, i.e.,

$$\forall F \exists x \exists G \forall H\ \phi(x, F(x), G, H) \rightarrow \exists G' \forall F \forall H \exists x\ \phi(x, F(x), \lambda t. G'(x, F(x), t), H) \tag{5}$$

For all x, y such that $\exists G \forall H\ \phi(x, y, G, H)$ holds we choose (AC) one such $G_{x,y}$. For any other x, y we choose $G_{x,y} = \lambda t.0$.
The function $\lambda x, y, t. G_{x,y}(t)$ is a G' function such that

$$\forall F \exists x \forall H\ \phi(x, F(x), \lambda t. G'(x, F(x), t), H)$$

holds (assuming the left-hand side in (5)).
Since we proved that there is such a function, we can add

$$\exists G' \forall F \exists x \forall H \ \phi(x, F(x), \lambda t.G'(x, F(x), t), H)$$

We move the quantifier $\forall H$ outwards, using a same principle as above.

The proof for the case of Σ_2^1 formula is simpler since we have the equivalence:

$\exists F \forall x \exists G \forall H \ \phi(x, F(x), G, H) \leftrightarrow \exists F \exists G \forall H \forall x \ \phi(x, F(x), G(x), H)$ □

2.5 Henkin Quantifiers in Finite Models

Many fields of computer sciences deal only with finite models, for example finite graph theory, Ramsey theory and finite group theory. Blass and Gurevich [2] use Henkin quantifiers to formulate problems on finite models.

Definition 2.14 *An* l-ary global predicate *for a vocabulary σ is a function π assigning to each finite σ-structure S an l-ary predicate π^s such that*

$$\pi^s : S^l \to \{\mathbf{true}, \mathbf{false}\}$$

The decision problem for π is:
Instance: A finite σ-structure S, and $\bar{a} \in S^l$
Question: Is $\pi^s(\overline{a})$ get the value true?

Examples

1. Any formula ϕ with n free variables defines an n - ary global predicate. When $n = 0$ the decision problem for this global predicate is testing for any given structure S, whether ϕ is true in S.

2. Let GRAPH be the class of finite graphs seen as a set of structures with exactly one binary relation which is irreflexive and symmetric. The following define GRAPH-global predicates:

   - The graph is connected (a global predicate of arity 0, $\bar{a}$ is empty).
   - Node x has at most $log(n)$ neighbors, where n is the number of nodes (a global predicate of arity 1).
   - There is a path from node x to node y (a global predicate of arity 2).

3. Let GROUP be the class of finite groups. The following GROUP-global relations are of arities 0,1, and 3, respectively:

   - The group is Abelian.
   - The index of the subgroup generated by an element x is at most $log(n)$ where n is the number of elements.

- The subgroup generated by the elements x and y contains the element z.

Using this definition we can characterize the computational complexity of Henkin quantifiers:

Theorem 2.15 *For any global predicate π, the following are equivalent:*
1) The decision problem for π is in $\mathcal{NP}$.
2) π is expressible by a Σ^1_1 formula.
3) π is expressible by formula $Q\phi$ where Q is a standard Henkin quantifier and ϕ is a quantifier-free formula.

Proof: The equivalence (1) $\leftrightarrow$ (2) is a theorem of Fagin [4] , and the equivalence (2) $\leftrightarrow$ (3) is a result of theorem 2.8. $\Box$

As an example, using a Henkin quantifier we can express a sentence ϕ which holds only if a finite graph defined by a predicate E is 3-colorable. This is a known NP-complete problem. Ordinary first-order quantifiers cannot express this since first-order definable predicates are in L (log-space computable), unless $L = P = NP$.
The construction of ϕ is as follows:

In a vocabulary σ with a binary predicate symbol E and three constant symbols 0,1,2, let ψ be the conjunction of:

- $x_1 = x_2 \rightarrow y_1 = y_2$
- $y_1 = 0 \vee y_1 = 1 \vee y_1 = 2$
- $E(x_1, x_2) \rightarrow y_1 \neq y_2$

Then, in any σ structure in which 0,1,2 denote distinct elements, we have

Proposition 2.16

$$\phi = \begin{pmatrix} \forall x_1 & \exists y_1 \\ \forall x_2 & \exists y_2 \end{pmatrix} \psi$$

holds if and only if the graph defined by E is 3-colorable.

proof
$\leftarrow$. If the graph is 3-colorable we need to show that ϕ holds, i.e., that the following Σ^1_1 sentence is true in σ:

$$\phi^* \equiv \exists f_1 \exists f_2 \forall x_1 \forall x_2 \psi(x_1, x_2, [\frac{f_1(x_1)}{y_1}], [\frac{f_2(x_2)}{y_2}])$$

Indeed, if the graph defined by E is 3-colorable, we can take the values both for f_1 and f_2 in the formula above to be the colors of the vertices x_i and x_j defined by the graph E. All of the conjuncts of the matrix of ϕ^* then become true.

$\rightarrow$. If ϕ is true then so is ϕ^*. According to the first conjunct, f_1 and f_2 are identical. Hence the following second order formula is true in this structure:
$\exists f\Big(\ \forall x(f(x) = 0 \vee f(x) = 1 \vee f(x) = 2) \wedge$
$$(\forall x_1 \forall x_2 [E(x_1, x_2) \rightarrow f(x_1) \neq f(x_2)])\ \Big)$$
This is exactly the definition of 3-coloring a graph defined by E. $\square$

2.6 Henkin Quantifiers and Transitive Closure

Transitive closure (Immerman [14], Gurevich [5]) is an operator that allows to define inside the language the transitive closure of any binary relation definable in the language (the transitive closure itself is again a binary relation). For examlpe, we can express relations like "X is a descendant of Y" using the relation "W is a son or a daughter of Z".

Definition 2.17 *Let σ be a signature for a first-order language with equality. The language $L^n_{TC}(\sigma)$ is defined like the usual first-order language which is based on φ, but with the addition of the following clause:*

1. *If φ is a well formed formula, $x_1, ..., x_k, y_1, ..., y_k$ $(k \leq n)$ are $2k$ distinct variables and $t_1, ..., t_k, s_1, ..., s_k$ are $2k$ terms, then*

$$(TC^k_{x_1,...,x_k,y_1,...,y_k}\varphi)(t_1, ..., t_k, s_1, ..., s_k)$$

is a well formed formula.
In this formula all the free occurrences of $x_1, .., x_k$,and $y_1, ..., y_k$ in φ are bound.

2. *The semantics of these new formulae is defined as follows. Given a structure I for σ with domain D and an assignment v in D for the variables of the language, define:*

$$I, v \models (TC^k_{\bar{x},\bar{y}}\varphi)(\bar{t}, \bar{s})$$

if and only f there exists $\bar{a}_0, \bar{a}_1, ..., \bar{a}_n \in D$ (where $n \geq 1$) such that $\bar{a}_0$ is the interpretation of $\bar{t}$ relative to I and v, $\bar{a}_n$ - the interpretation of $\bar{s}$, and for $i = 0, 1, ..., n-1$ we have:

$$I, v(\bar{x} := \bar{a}_i, \bar{y} := \bar{a}_{i+1}) \quad \models \varphi$$

Example. There is no finite set of first-order axioms that is categorical with the standard model of the natural numbers together with the standard $*$ and $+ < \mathbb{N}, +, * >$.

The following finite set of axioms is easily seen to be categorical with $< \mathbb{N}, +, * >$ as their unique model:

Define	$t < s \equiv_{Df} (TC^1_{x,y} y = S(x))(t, s)$
N1	$\forall x(S(x)) \neq 0$
N2	$\forall x \forall y(S(x) = S(y) \rightarrow x = y)$
N3	$\forall x(x \neq 0 \rightarrow 0 < x)$[9]
N4	$\forall x(x + 0 = x)$
N5	$\forall x(x + S(y) = S(x + y))$
N6	$\forall x(x * 0 = 0)$
N7	$\forall x \forall y(x * S(y) = x * y + x)$

A result of this is that $L^1_{TC}(< 0, S, +, * >)$ is not arithmetical (and so is not decidable or recursively enumerable).
The following logical rule is a general form of induction as a deduction rule in $L^1_{TC}(< 0, S, +, * >)$:

$$\frac{\Gamma, \psi,\ y = S(x) \quad \Rightarrow \quad \Delta, \psi(y/x)}{\Gamma, \psi(0/x), (TC^1_{x,y} y = S(x))(0, t) \ \Rightarrow \Delta, \psi(t/x)} \tag{6}$$

First order logic enriched with the TC operator, is strongly connected to Henkin quantifiers. Following Blass and Gurevich [2] we can express TC and a special form of Henkin quantifier (*Narrow Henkin Quantifier*) in terms of each other.

Definition 2.18 (Narrow Henkin Quantifiers)

1. *We allow in our logic Boolean variables $\alpha, \beta, ...$, which range over {0,1}, i.e., two-sorted language whose second sort is to be interpreted as {0,1} in all structures. A narrow Henkin quantifier is:*

$$\begin{pmatrix} \forall \bar{x} & \exists \alpha \\ \forall \bar{y} & \exists \beta \end{pmatrix}$$

Where α and β are Boolean variables, while $\bar{x}$ and $\bar{y}$ are tuples *of individuals or Boolean variables. In the second order form semantics of such a quantifier the range of the functions is {0,1}.*

We say that $\bar{x}$ and $\bar{y}$ are compatible *if they have the same length and have variables of the same type at corresponding positions. We write $\bar{x} = \bar{y}$ as an abbreviation for $\bigwedge_i x_i = y_i$ provided that $\bar{x}$ and $\bar{y}$ are compatible; if they are incompatible then $\bar{x} = \bar{y}$ means false.*

[9] This rule is the main difference to first-order logic. In order to categorize the standard model we need some sort of induction in the system, i.e, expressing that all elements are successors of the first element (and no other non-standard numbers exists).

2. *An* equality bound *narrow Henkin quantifier is a narrow Henkin quantifier whose second order form is equivalent to the following formula:* [10]

$$\exists f \forall x \forall y \phi(x, y, f(x), f(y))$$

A narrow Henkin quantifier is a special case of the standard Henkin quantifier, since in a language with two different constants (say 0 and 1) [11] we can define:

$$\begin{pmatrix} \forall \bar{x} & \exists \alpha \\ \forall \bar{y} & \exists \beta \end{pmatrix} \phi \equiv \begin{pmatrix} \forall \bar{x} & \exists i \\ \forall \bar{y} & \exists j \end{pmatrix} \Big((i = 0) \vee (i = 1)\Big) \wedge \Big((j = 0 \vee (j = 1)\Big) \wedge \phi(\tfrac{i}{\alpha}, \tfrac{j}{\beta})$$

Yet, the complexity classes corresponding to these quantifiers are different. In [5] it is shown that in finite models the decision problems that are expressible using narrow Henkin quantifiers are in co-nondeterministic logspace class, where the standard Henkin quantifier expresses decision problems that are in NP (as seen in theorem 2.15).

Lemma 2.19 *Narrow Henkin quantifiers and equality bound narrow Henkin quantifiers can be expressed in terms of each other.*

proof
An equality bound narrow Henkin quantifier is equivalent to the following narrow Henkin quantifier

$$\begin{pmatrix} \forall x & \exists \alpha \\ \forall y & \exists \beta \end{pmatrix} (x = y \to \alpha = \beta) \wedge \phi(x, y, \alpha, \beta)$$

For the other direction - the second order form semantics of the general narrow Henkin quantifier

$$\begin{pmatrix} \forall \bar{x} & \exists \alpha \\ \forall \bar{y} & \exists \beta \end{pmatrix} \phi(\bar{x}, \bar{y}, \alpha, \beta)$$

is:

$$\exists f \exists g \forall \bar{x} \forall \bar{y} \phi(\bar{x}, \bar{y}, f(\bar{x}), g(\bar{y})) \tag{7}$$

We define the following equality bound narrow Henkin quantifier:

$$\exists h \forall \bar{x} \gamma \forall \bar{y} \delta(\gamma = 1 \wedge \delta = 0 \to \phi(\bar{x}, \bar{y}, h(\bar{x}\gamma), h(\bar{y}\delta)) \tag{8}$$

(8) imlies (7) using the following definitions:
$f(\bar{x}) = h(\bar{x}, 1)$
$g(\bar{y}) = h(\bar{y}, 0)$ □

[10] Krynicki and Väänänen [16] refer to these kind of quantifiers as "function quantifiers".
[11] If the language contains the equality predicate we can skip this provision.

Theorem 2.20 (Blass & Gurevich)
Narrow Henkin quantifiers and transitive closures can be expressed in terms of each other, i.e., positive occurrences of either can be expressed by negative occurrences of the other.

proof
The TC operator can be expressed by a second order formula. In fact, $(TC^k_{\bar{u},\bar{v}}\psi)(\bar{x},\bar{y})$ is logically equivalent to:

$$\exists \bar{z_0} ... \exists \bar{z_n} (\bar{z_0} = \bar{z} \wedge \psi(\bar{z_0}, \bar{z_1}) \wedge ... \wedge \psi(\bar{z_{n-1}}, \bar{z_n}) \wedge \bar{z_n} = \bar{y})$$

which is equivalent to

$$\neg \exists f [f(\bar{x}) = 1 \wedge f(\bar{y}) = 0 \wedge \forall \bar{u} \forall \bar{v} (f(\bar{u}) = 1 \wedge \psi(\bar{u}, \bar{v}) \rightarrow f(\bar{v}) = 1)]$$

Note that f equals 1 over the set of elements which are accessible via ψ from $\bar{x}$.
This second order formula is equivalent to the following narrow Henkin quantifier formula:

$$\neg \begin{pmatrix} \forall \bar{u} & \exists \alpha \\ \forall \bar{v} & \exists \beta \end{pmatrix} [(\bar{u} = \bar{v} \rightarrow \alpha = \beta) \wedge (\bar{u} = \bar{x} \rightarrow \alpha = 1) \wedge (\bar{u} = \bar{y} \rightarrow \alpha = 0)$$

$$\wedge (\alpha = 1 \wedge \psi(\bar{u}, \bar{v}) \rightarrow \beta = 1)]$$

For the other direction:

Using lemma 2.19 we want to express in terms of TC a general formula with an equality bound narrow Henkin quantifier. Its second order semantics is:

$$\exists f \forall \bar{x} \forall \bar{y} \ \phi(\bar{x}, f(\bar{x}), \bar{y}, f(\bar{y})) \tag{9}$$

f - is a Boolean function.

Let us assume that (9) holds, namely that such a function f exists. If, for certain $\bar{x}, \alpha, \bar{y}$ and β, $\neg\phi(\bar{x}, \alpha, \bar{y}, 1-\beta)$ is true, then from the two hypothesis:

- $\phi(\bar{x}, f(\bar{x}), \bar{y}, f(\bar{y}))$ is true.
- $\phi(\bar{x}, \alpha, \bar{y}, 1-\beta)$ is false.

if $f(\bar{x})$ is assigned the value α then $f(\bar{y})$ must be assigned the value β, or else it would not satisfy $\phi(\bar{x}, f(\bar{x}), \bar{y}, f(\bar{y}))$. The value α for $f(\bar{x})$ "forces" the value β for $f(\bar{y})$. The same holds if $\neg\phi(\bar{y}, 1-\beta, \bar{x}, \alpha)$ is true.
We use $\hat{}$ for a concatenation sign. Let

$$\psi(\bar{x}\hat{}\alpha, \bar{y}\hat{}\beta) \triangleq \neg\phi(\bar{x}, \alpha, \bar{y}, 1-\beta) \vee \neg\phi(\bar{y}, 1-\beta, \bar{x}, \alpha)$$

Whenever $\psi(\bar{x}^\frown\alpha, \bar{y}^\frown\beta)$ holds, the value α for $f(\bar{x})$ "forces" the value β for $f(\bar{y})$. Since this forcing is clearly a reflexive transitive relation on tuples of $k+1$ elements $(\bar{x}^\frown\alpha)$ we see that $\bar{a}$ forces $\bar{b}$ whenever $(TC^{k+1}_{\bar{x}\alpha,\bar{v}\beta}\psi)(\bar{a},\bar{b})$ holds.

In particular, if for certain $\bar{t}$

$$(TC^{k+1}_{\bar{x}\alpha,\bar{y}\beta}\psi)(\bar{t}^\frown\gamma, \bar{t}^\frown(1-\gamma))$$

holds, then $f(\bar{t})$ cannot be assigned the value γ. Otherwise we would get a contradiction using the "forcing" relation which states that $f(\bar{t})$ should be assigned the value $1-\gamma$. Therefore a necessary condition for such an assignment f to exists is:

$$\neg\exists\bar{t}\exists\gamma[(TC^{k+1}_{\bar{x}\alpha,\bar{y}\beta}\psi)(\bar{t}^\frown\gamma, \bar{t}^\frown(1-\gamma)) \wedge (TC^{k+1}_{\bar{x}\alpha,\bar{y}\beta}\psi)(\bar{t}^\frown(1-\gamma), \bar{t}^\frown\gamma)] \qquad (10)$$

The first conjunct states that $f(\bar{t})$ cannot be assigned the value γ, and the second conjunct states that $f(\bar{t})$ cannot be assigned the value $1-\gamma$ (but f is a Boolean function...).

We complete the proof by showing that this necessary condition is also sufficient. This is the negated TC-sentence which is the equivalent to the equality bound narrow Henkin quantifier whose semantics is given in (9).

As a consequence of the definition of the TC operator, $(TC^{k+1}_{\bar{x}^\frown\alpha,\bar{y}^\frown\beta}\psi)$ defines a pre-ordering (i.e. a reflexive transitive relation $\leq$) on the tuples $\bar{x}^\frown\alpha$. Furthermore, if the structure satisfies the necessary condition (10), then we never have $\bar{x}^\frown\alpha$ and $\bar{x}^\frown(1-\alpha)$ each $\leq$ the other.

Lemma 2.21 *Let $(X,\leq)$ be a pre-ordered set, and let $i : X \to X$ be an order-reversing function such that for every $x \in X$ $i(i(x)) = x$ and either $x \not\leq i(x)$ or $i(x) \not\leq x$ or both. Then there is a subset T of X, closed upward for $\leq$, containing exactly one of x and $i(x)$ for each $x \in X$.*

First, make the preliminary observation that $\psi(\bar{x}^\frown\alpha, \bar{y}^\frown\beta)$, and therefore also $(TC^{k+1}_{\bar{x}^\frown\alpha,\bar{y}^\frown\beta}\psi)$, are invariant when interchanging $\bar{x}$ and $\bar{y}$ while simultaneously replacing α with $(1-\beta)$ and β with $(1-\alpha)$.

Granting the lemma for the moment, we apply it to the set of tuples $\bar{x}^\frown\alpha$ pre-ordered according to the sentence $(TC^{k+1}_{\bar{x}^\frown\alpha,\bar{y}^\frown\beta}\psi)$, with $i(\bar{x}^\frown\alpha) = \bar{x}^\frown(1-\alpha)$. The preliminary observation guarantees that i is an order-reversing function.

Note that the necessary condition (10) permits the last assumption of lemma 2.21. We obtain a set T as in the conclusion of the lemma. Define $f(x) = 1$ if and only if $\bar{x}^\frown 1 \in T$, or similarly (property of T) $\bar{x}^\frown 0 \notin T$. Thus, we always have $\bar{x}^\frown f(x) \in T$ and $\bar{x}^\frown(1-f(x)) \notin T$. This assignment f has the desired property that $\phi(\bar{x}, f(\bar{x}), \bar{y}, f(\bar{y}))$ holds for all $\bar{\mathrm{x}}$ and $\bar{\mathrm{y}}$.

To see this, suppose $\bar{x}$ and $\bar{y}$ were a counter-example, i.e., the following holds:

$$\neg\phi(\bar{x}, f(\bar{x}), \bar{y}, f(\bar{y}))$$

Then $\psi(\bar{x}^\frown f(x), \bar{y}^\frown(1-f(y)))$ would hold and, since ψ implies $(TC^{k+1}_{\bar{x}\alpha,\bar{y}\beta}\psi)$, we would have $\bar{x}^\frown f(x) \leq \bar{y}^\frown(1-f(y))$.

But $\bar{x}^\frown f(X) \in T$ and $\bar{y}^\frown(1-f(y)) \notin T$, and since T is closed upward, it is a contradiction. □

Lemma 2.21 is proved by induction on the number of elements of X, the case $X = \emptyset$ being vacuously true. If $X \neq \emptyset$, consider an arbitrary element $x \in X$. Replacing x by $i(x)$ if necessary, we may assume $x \not\leq i(x)$. We put into (respectively out of) T all elements $\geq x$ (respectively $\leq i(x)$); there is no conflict here as $x \not\leq i(x)$. Note that an element y has been put into T if and only if $i(y)$ has been put out of T. The set Y of elements whose membership is still in doubt is smaller than X (as $x \notin Y$). Apply the induction hypothesis to Y (with the reduction of $\leq$ and i) to get $T' \subseteq Y$ satisfying the conclusion of the lemma for Y. Finally, put all members of T' into T and take all members of $Y - T'$ out of T. It is easy to verify that T has all the required properties.

A compactness argument proves that the lemma holds without the hypothesis that X is finite. □

Earlier in corollary 2.10 we used Walkoe's theorem to express the induction axiom in a language with Henkin quantifiers. Although unforeseen, we will now show that by using another method, i.e. the TC operator, we get the same result.

Proposition 2.22 *The TC axiom $N3$: $\forall x(x \neq 0 \rightarrow 0 < x)$, when translated into Henkin quantifiers using the theorem above, gives the same Henkin formula from corollary 2.10.*

proof
The following is the TC version of axiom $N3$:

$$\forall z\ (z = 0) \vee (TC_{a,b}\, b = a + 1)(0, z)$$

Using theorem 2.20 we get a translation to the following Henkin quantifier:

$$\forall z \neg \begin{pmatrix} \forall u & \exists \alpha \\ \forall v & \exists \beta \end{pmatrix} (u = v \rightarrow \alpha = \beta) \wedge (u = 0 \rightarrow \alpha = 1) \wedge (u = z \rightarrow \alpha = 0) \\ \wedge(\alpha = 1 \wedge v = u + 1 \rightarrow \beta = 1)$$

The case $(z = 0)$ is included due to the second and the third conjuncts.
This formula is equivalent to

$$\neg \begin{pmatrix} \forall u & \exists \alpha \\ \forall v & \exists \beta \\ & \exists z \end{pmatrix} (u = v \rightarrow \alpha = \beta) \wedge (u = 0 \rightarrow \alpha = 1) \wedge (u = z \rightarrow \alpha = 0) \\ \wedge(\alpha = 1 \wedge v = u + 1 \rightarrow \beta = 1)$$

and this is exactly the Henkin sentence from corollary 2.10. □

2.7 Truth Definition using Σ_1^1

Tarski's impossibility result (1935) and Gödel's incompleteness theorems (1931) show that for a formal language L of first-order Peano arithmetic, one cannot give an explicit definition of truth in the standard model $\mathbb{N}$ of L.

We can define a truth predicate for the language H' using a Σ_1^1 formula. Using Walkoe algorithm we can transfer this formula to a formula in H'. Thus we have a truth predicate for H' within the same language.

This truth definition is achieved due to the lack of negation in H'. Tarski's theorem on the undefinability of truth assume that the object language is closed under classical negation. In this section we show how to define such a truth predicate.

In order to define Gödel numbers we assume that the language H' has arithmetical language as its sub-language. Thus we have the unary function $S(x)$ (successor of x), a constant 0 to denote zero, and the arithmetical axioms. The basic idea of Gödel numbering (arithmetization) is to code the syntax of the given language with elementary arithmetic, i.e. for each formula there is a corresponding unique term in the arithmetical subsystem (a numeral whose interpretation is a natural number). For the details of such coding using primitive recursive functions see Smorynski [24].

$\ulcorner\varphi\urcorner$ is the Gödel number of a sentence φ, and $\underline{n}$ is the numeral corresponding to the natural number n. From now on we shall simply write $\ulcorner\varphi\urcorner$ in the syntax instead of $\underline{\ulcorner\varphi\urcorner}$.

Lemma 2.23 *There is a Σ_1^1-formula $\Psi(x)$ with one free variable in the signature of PA which defines truth in $\mathbb{N}$ for the first-order language of PA, i.e. for any first-order sentence φ in the signature of PA we have:*

$$\mathbb{N} \models \Psi(\ulcorner\varphi\urcorner) \quad \Leftrightarrow \quad \mathbb{N} \models \varphi$$

Sketch of the proof
Let the recursive predicates "x is a Gödel number representing a sentence of PA", "y is a variable", "x is a formula of PA", "x is a true atomic formula" be abbreviated by "$Sent_{PA}(x)$", "$Var(x)$", "$Form_{PA}(x)$, "$Tatomic(x)$". We also use a recursive function sub that has the property:

$$sub(\ulcorner\varphi\urcorner, \ulcorner x\urcorner, n) = \ulcorner\varphi(\underline{n})\urcorner$$

Using standard practice, we abbreviate $sub(\ulcorner\psi\urcorner, \ulcorner y\urcorner, x)$ by $\ulcorner\psi(\dot{x})\urcorner$.

The second-order truth predicate has obvious intuitive meaning. When applied to the Gödel number n of a sentence, it says that there is a one-place predicate X which behaves in the way a truth predicate should, and that n has this predicate property. In other words, the truth predicate has the form:

$$\Psi(x) \triangleq (\exists X)(L[X] \wedge X(x)) \tag{11}$$

Where $L(X)$ is the conjunction of the following formulae:

a. $\forall z(X(z) \rightarrow Sent_{PA}(z))$

b. $\forall z(Tatomic(z) \rightarrow X(z))$

c. $\forall z \forall \psi \ ((Sent_{PA}(\psi) \wedge z = \ulcorner \neg \psi \urcorner) \rightarrow (X(z) \leftrightarrow \neg X \ulcorner \psi \urcorner))$

d. $\forall z \forall \psi, \chi \ ((Sent_{PA}(\psi) \wedge Sent_{PA}(\chi) \wedge \ z = \ulcorner \psi \wedge \chi \urcorner) \rightarrow (X(z) \leftrightarrow (X(\ulcorner \psi \urcorner) \wedge X(\ulcorner \chi \urcorner))))$

e. $\forall z \forall \psi \forall y((Form_{PA}(\psi) \wedge Var(y) \wedge \ z = \ulcorner \exists y \psi \urcorner) \rightarrow (X(z) \leftrightarrow \exists v X(\ulcorner \psi(\dot{v}) \urcorner)))$

□

Ψ defines truth predicate for the first-order sentences in the signature of PA using a Σ^1_1 formula. The following is the truth predicate for for all the sentences in H' (using only the signature of PA):

$$Tr(x) = (Sent_{PA}(x) \rightarrow \Psi(x)) \ \vee$$
$$\exists \psi, t, u, v, w \Big[Form_{PA}(\psi) \wedge Var(t) \wedge ... \wedge Var(w) \wedge x = \ulcorner \begin{pmatrix} \forall t & \exists u \\ \forall v & \exists w \end{pmatrix} \psi \urcorner$$
$$\wedge \begin{pmatrix} \forall x_0 & \exists x_1 \\ \forall y_0 & \exists y_1 \end{pmatrix}^{(*)} \Psi(\ulcorner \psi(\dot{x_0}, \dot{x_1}, \dot{y_0}, \dot{y_1}) \urcorner) \Big]$$

(*) We first translate this Henkin quantifier into the second order form semantics, to yield a Σ^1_1 formula. $Tr(x)$ can then be moved into prenex normal form and translated into a formula in H' using Walkoe's algorithm. For more details refer to Sandu & Hyttinen [23].

3 IF-Logic

Another approach to controlling the scope of quantifiers in a first-order formula is Independence Friendly Logic (IF-Logic ; Hintikka and Sandu [11]). This logic is an extension of first-order logic (only one operator added), which has intuitive semantics, based on a game between two imaginary opponents. In the semantic game we explicitly specify the scope of each universal in the formula. This logic is strongly connected with Henkin quantifiers and Σ^1_1.

We begin with a simple form of IF-logic in which negation can appear only in front of atomic formulae. No negation signs are allowed within a formula. Allowing negation inside a formula is a complex matter, as we shall see, which requires extensive changes to the syntax and semantics.

3.1 Syntax of IF-Logic

We extend the syntax of first-order logic with a new slash operator '/', which like the Henkin quantifier, captures the notion of "independent quantifier".

The set of well formed formulae over an IF-language L contains the set of first-order formulae in negation normal form over the language L plus those which arise from the first-order formulae as follows:

1. Make sure that all quantified variables are pairwise disjoint (if not replace them by new symbols.)

2. If $\exists x$ occurs within the scope of $\forall y_1 ... \forall y_n$, then it may be replaced by $(\exists x / \forall y_1, ..., \forall y_n)$.

3. If $\forall x$ occurs within the scope of $\exists y_1 ... \exists y_n$, then it may be replaced by $(\forall x / \exists y_1, ..., \exists y_n)$. [12]

3.2 Game Theoretical Semantics (GTS)

With each IF-formula φ, each model M of this language, and each assignment v (restricted to the free variables of φ), a semantic game $G(\varphi, M, v)$ is played on M, whose domain is denoted by D. The game is played by two players, player $\exists$ and player $\forall$. Player $\exists$ has the role of verifier and player $\forall$ has the role of falsifier. Each symbol which occurs in φ and belongs to the set $\{(\exists x / \forall y_1, ..., \forall y_n), (\forall x / \exists y_1, ..., \exists y_n),\ \vee, \wedge\}$ prompts a move in $G(\varphi, M, v)$ according to the following rules:

- If φ is a literal, then neither player makes any move in $G(\varphi, M, v)$ and the game is over. If $(M, v) \models \varphi$ then the verifier wins the game. If $(M, v) \not\models \varphi$ then the falsifier wins the game.

[12] Van Benthem [27] suggests variations for the syntax of IF-logic with slash on other logical connectives, not only on quantifiers.

- If $\varphi = (\exists x/\forall y_1, ..., \forall y_n)\psi$ then the verifier chooses an individual $a \in D$ and the game continues as $G(\psi, M, v')$ where v' is like v except that $v'(x) = a$.

- If $\varphi = (\forall x/\exists y_1, ..., \exists y_n)\psi$ then the falsifier chooses an individual $a \in D$ and the game continues as $G(\psi, M, v')$ where v' is the same as v except that $v'(x) = a$.

- If $\varphi = \varphi_1 \vee \varphi_2$ then the verifier chooses $m \in \{1, 2\}$ and the game continues as $G(\varphi_m, M, v)$.

- If $\varphi = \varphi_1 \wedge \varphi_2$ then the falsifier chooses $m \in \{1, 2\}$ and the game continues as $G(\varphi_m, M, v)$.

Game $G(\varphi, M, v)$ is a sequence of choices made as prescribed by the above game rules. It ends with an atomic formula or its negation and an assignment v' that is the same as v except that it may give new values to the variables in the formula.

Strategy S for a player in the game $G(\varphi, M, v)$ is a set of functions. Each function $f_Q \in S$ corresponds to a move of the respective player in the game prompted by the symbol Q of φ. The function value is the choice for that move.

1. If Q is $\exists x/\forall y_1, ..., \forall y_k$ (similarly $\forall x/\exists y_1, ..., y_k$) in the scope of the universal quantifiers $\forall y_1, ..., \forall y_k \forall z_1, ..., \forall z_m$ ($\exists y_1, ..., \exists y_k \exists z_1, ..., \exists z_m$) respectively, then the arguments of f_Q are all sequences of the form $(z_1, ..., z_m)$ with values in D. f_Q range is D.

   In this case we say that the move prompted by Q is *informational independent* of the moves prompted by $\forall y_1, ..., \forall y_k$ ($\exists y_1, ..., \exists y_k$) and *informational dependent* on the moves prompted by $\forall z_1, ..., \forall z_m$ ($\exists z_1, ..., \exists z_m$).

2. If Q is $\phi_1 \vee \phi_2$ ($\phi_1 \wedge \phi_2$) in the scope of $\forall y_1, ..., \forall y_k$ ($\exists y_1, ..., \exists y_k$) then the arguments of f_Q are all sequences of the form $(y_1, ..., y_k)$ with values in D. f_Q range is {1,2}

A strategy for a player in game $G(\varphi, M, v)$ is a winning one, if by using it in the game the player wins every play of the game regardless of the opponent's moves.

Our definition of a strategy causes these semantic games to be of imperfect information. Whenever a slash is involved in the game, the player must make his choice independent of the values already given to variables under the slash, i.e., this information is 'hidden' from him at that point of the game. This demonstrates the control of information flow in a formula.

Definition 3.1 (Truth)
(i) A formula φ of the IF-language is true in M with respect to the assignment v ($(M, v) \models_{GTS} \varphi$) if and only if there is a winning strategy for player $\exists$ in $G(\varphi, M, v)$.
(ii) A formula φ of the IF-language is false in M with respect to the assignment v ($(M, v) \not\models_{GTS} \varphi$) if and only if there is a winning strategy for player $\forall$ in $G(\varphi, M, v)$.

Assuming the axiom of choice we have the following

Theorem 3.2 *(Hintikka [10]; Hodges [13].) For any first-order sentence φ an assignment v and a model M, Tarski-type truth and games semantics truth coincide, i.e.,*

$$(M, v) \models_{Tarski} \varphi \quad \leftrightarrow \quad (M, v) \models_{GTS} \varphi$$

Remark. According the game theoretical semantics, a strategy function cannot use information gained from *all* earlier moves. We exclude a players own moves from his/her information base. In IF-logic, a strategy function is defined only over the opponent's earlier moves and not over a player's own earlier moves. For example, moves connected with existential quantifiers are always independent of earlier moves with existential quantifiers. This restriction helps to prevent "forbidden" dependencies of moves on earlier moves. This is referred to as the 'signaling' phenomenon. Consider the sentence:

$$\forall x \forall z (\exists y / \forall z)(\exists u / \forall x) \varphi(x, y, z, u)$$

Without the restriction, u is still dependent on x since it depends on y (which depends on x).

In the next section we introduce a wider IF-logic (it has negation and is compositional) where this restriction is removed, thus the signaling phenomenon occurs. This logic is more intuitive when considering human games since a player can gain information from all earlier moves.

3.3 IF-Logic Examples

Example 1
A winning strategy for player $\exists$ in the game $\forall x \exists y (x = y)$ is a function f_1 where for each possible value given to x, it assigns the same value for y.

Example 2
A strategy for player $\exists$ in the game $\forall x \exists y [(x = y) \vee (x + 1 = y)]$ consists of two strategy functions f_1 and f_2. f_1 is the same function $\lambda x.x$ from the previous example, and f_2 is $\lambda x.1$, i.e., it chooses the first disjunct to continue the game with.

Example 3
In IF-logic, we have sentences that are neither true nor false in a given model (the law of excluded middle doesn't hold in the meta-language). Consider the following sentence

$$\forall x(\exists y/\forall x)\ x = y$$

This formula is neither true nor false in a domain with more than one element. 'True in a model' means that the verifier has a winning strategy, and 'false in a model' means that the falsifier has a winning strategy. But neither is the case in any model with more than two elements, since the falsifier (who chooses x) cannot guarantee that the verifier (who chooses y) will not choose the other element. The verifier (who doesn't know what was assigned to x according to the slash rule) doesn't know which element to choose in order that y equals x.

3.4 IF-Logic and Henkin Quantifiers

We already saw that both Henkin quantifiers and IF-logic capture the notion of independence of quantifiers i.e., choices made in parallel. We are now formally going to prove that the languages H' (formulae with Henkin quantifiers initially) and IF-logic have the same expressive power with respect to truth in a model.

Theorem 3.3 *Given an IF-sentence φ in a language L we can find a Σ_1^1 sentence ψ over the language L such that for all (M, v):*

$$(M, v) \models_{GTS} \varphi \quad iff \quad (M, v) \models \psi$$

proof
We perform the following syntactical operations on φ:

(i) Replace all the occurrences of the form $(\forall x/\exists y_1, ..., \exists y_k)$ in φ by $(\forall x)$. Call the result $\varphi^{(1)}$.

(ii) On $\varphi^{(1)}$ perform the following:
drop all the quantifiers of the form $(\exists x/\forall y_1, ..., \forall y_k)$, and replace the free variable x in the result by $f_x(z_1, ..., z_n)$, where $\forall z_1, ..., \forall z_k$ are all the universal quantifiers within which the scope $(\exists x/\forall y_1, ..., \forall y_k)$ is, except $\forall y_1, ..., \forall y_k$; then prefix the result with $(\exists f_x)$; Call the result ψ.

The equivalence (in terms of truth in a model) between φ and $\varphi^{(1)}$ is guaranteed since the falsifier strategy is irrelevant to the truth of a formula. The equivalence between $\varphi^{(1)}$ and ψ follows directly from the definition of the functions f_x in connection with strategy. For example, when we replace each $(\exists x/\forall y_1, ..., \forall y_k)$ in $\varphi^{(1)}$ with a function $f_x(z_1, ..., z_k)$ and a prefix $\exists f_x$ we get a Σ_1^1 sentence. This sentence is true in a model only if such a function exists,

and we can take it as a strategy function that depends only on quantifiers not under the slash. Conversely, a strategy function for φ can be refer to show the truth of ψ. □

The sentence ψ is equivalent to φ only in the weak sense - they are true in the same models. This does not mean however that they are false in the same models.

Theorem 2.8 (Walkoe's theorem) states that every Σ^1_1 sentence is equivalent to a first order sentence prefixed with a standard Henkin quantifier. The converse is true as well, since Henkin quantifiers semantics is defined using the second-order existential form which is a Σ^1_1 sentence. Theorem 3.3 shows that an IF-formula has a Σ^1_1-equivalent. To complete the circle, we show that the standard Henkin quantifier is definable in IF-logic.

Theorem 3.4 *Given a Henkin formula $\varphi \in H'$ we can find an IF-formula ψ over the first-order language of H' such that for all (M, v)*

$$(M, v) \models \varphi \quad \Leftrightarrow \quad (M, v) \models_{GTS} \psi$$

proof
A general formula in H' is of the form

$$\begin{pmatrix} \forall x^1_1 \ ... \ \forall x^1_{n_1} \exists a_1 \\ \forall x^2_1 \ ... \ \forall x^2_{n_2} \exists a_2 \\ \vdots \\ \forall x^m_1 \ ... \ \forall x^m_{n_m} \exists a_m \end{pmatrix} \sigma$$

its IF-logic equivalent is the following formula:

$$\begin{array}{l} \Big(\forall x^1_1 \ ... \ \forall x^1_{n_1} \exists a_1\Big) \\ \Big(\forall x^2_1 \ ... \ \forall x^2_{n_2} (\exists a_2 / \forall x^1_1 \ ... \ \forall x^1_{n_1})\Big) \\ \vdots \\ \Big(\forall x^m_1 \ ... \ \forall x^m_{n_m} (\exists a_m / \forall x^1_1 \ ... \ \forall x^1_{n_1} ... \forall x^{m-1}_1 \ ... \ \forall x^{m-1}_{n_{m-1}})\Big)\sigma \end{array}$$

We use a same argument from theorem 3.3 to show this. The strategy functions of the IF-formula are of the same form of the existentials in the Σ^1_1 interpretation of the Henkin formula. They have the same order of appearance and the same arguments. □

The last two theorems together with Walkoe's theorem show that Σ^1_1, the logic H' (Henkin quantifiers) and IF-logic are languages with the same expressive power.

Example. We can use the last two theorems to express a sentence which is true in a model M if and only if M has infinitely many elements. This sentence is not expressible in a first-order language.

A model M is infinite is there exists a 1-1 function f which is not onto, i.e., M is infinite iff the following Σ_1^1 sentence is true:

$$\exists f \exists g \exists z \forall x \ (x = g(f(x)) \ \wedge \ f(x) \neq z)$$

using Walkoe's algorithm to translate any Σ_1^1 sentence to Henkin quantifiers we get

$$\begin{pmatrix} \forall x \ \exists y \\ \forall t \ \exists w \\ \exists z \end{pmatrix} (t = y) \rightarrow [x = w \wedge y \neq z]$$

using theorem 3.4 we get

$$\exists z \forall x \exists y \forall t \exists w / \forall x \ (t = y) \rightarrow [x = w \wedge y \neq z]$$

3.5 Negation in IF-Logic

The language of IF-logic presented so far allows negation signs only in front of atomic formulae. The fact that the law of excluded middle doesn't hold (in the meta-language) shows that trying to add negation to IF-logic will give a different kind of behavior than the classical first-order negation. We will see that adding negation to IF-logic is a complex matter, which requires changing syntax and semantic rules.

Negation as Role-Switching. The common first-order game semantics for negation is switching the roles of the players (for example in [12]). In IF-logic, however, there is no intuitive meaning for a slashed quantifier with negation in front of it. Consider for example

$$\forall x \exists y \neg \forall z (\exists w / \forall x) \varphi(x, y, z, w)$$

If we want the negation sign to prompt for a role switch in the game, it is player $\forall$'s turn to play when we get to $(\exists w / \forall x)$. But what exactly does this mean? Since we defined a strategy function to range only over the opponent's earlier moves this stage of the game is meaningless. We haven't defined how can player $\forall$ make a move independent of his earlier choice to x. The first change to IF-logic is to allow strategy functions to gain information from all earlier moves, and to remove quantification signs after the slash. The above formula becomes

$$\forall x \exists y \neg \forall z (\exists w / x) \varphi(x, y, z, w)$$

The Principle of Compositionality. The next semantical problem is preserving compositionality. The principle of compositionality states that the semantic interpretation of a complex expression is determined by the semantic interpretations of its constituent expressions in addition to the way they are combined in it (Gottlob Frege seems to be the source of this principle).

Compositional semantics (like first-order semantics) enables us to find equivalent formulae and to instantiate one in place of the other in a complex expression. Compositional semantics for IF-logic is needed in order to show its equivalence to H' (the logic of Henkin quantifiers) and to Σ^1_1. This result was proven for IF-logic without negation (theorems 3.3, 3.4).

Using compositionality we can prove a negation normal form theorem for IF-logic. The equivalence to Σ^1_1 is then an immediate result (since we can eliminate all negation signs we can transform the resulting formula to a Σ^1_1 formula as we did before).

In order to preserve compositionality we must define the truth of a formula φ in a structure $\mathfrak{U}$ with regard to a *set* V of valuations (not only a single one as before). This is the major change made to the semantics.

Following Hodges [13] [13] , Caicedo and Krynicki [29] without proofs, we briefly review the changes needed to support negation.

Syntax. Given a first order signature τ (namely, a set of relation, function and constant symbols, with arities assigned to the first two kinds), the *terms* and *atomic formulae* of the language with equality $L_{ii}(\tau)$ are defined as in first-order logic. The set of *formulae* of $L_{ii}(\tau)$ is defined as the least set F such that:

- Each atomic formula of $L_{ii}(\tau)$ belongs to F.
- If $\varphi, \psi \in F$ then $\neg\varphi, (\varphi \vee \psi) \in F$.
- If $\varphi \in F$ and $\bar{y}$ is a finite list of distinct variables not containing x, then $(\exists x/\bar{y})\varphi \in F$.
- We assume that in every formula the quantified variables are pairwise disjoint and different from the free variables of the formula.

$(\forall x/\bar{y})\varphi$ is an abbreviation of $\neg(\exists x/\bar{y})\neg\varphi$.
$(\varphi \wedge \psi)$ is an abbreviation of $\neg(\neg\varphi \vee \neg\psi)$.

The definition of a *subformula* is the same as in elementary logic (it must include the position of the subformula in the syntactical tree of the formula).

The set $Fv(\varphi)$ of *free variables* of φ is defined inductively as usual, except for the following clause:

$$Fv(\exists x/y_1, ..., y_k\psi) = (Fv(\psi) - x) \cup \{y_1, ..., y_k\}$$

We need to relativize this notion to the subformulae of a formula. The set $Fv^{\varphi}(\psi)$ (free variables of ψ *relatively* to φ) is defined inductively from the outside in as follows:

[13] Hodges uses a different terminology. He refers to valuations as 'deals of cards', a set of deals as a 'trump', and an independent function as a 'uniform strategy function'.

- $Fv^{\varphi}(\varphi) = Fv(\varphi)$
- If ψ occurs after a negation symbol then $Fv^{\varphi}(\psi) = Fv^{\varphi}(\neg\psi)$
- If ψ is a disjunct in a disjunction then $Fv^{\varphi}(\psi) = Fv^{\varphi}(\psi \vee \psi') = Fv^{\varphi}(\psi' \vee \psi)$, where ψ' is the other disjunct (this rule is the main difference from the previous definition).
- If ψ occurs after the quantifier prefix $\exists x/\bar{y}$ then $Fv^{\varphi}(\psi) = Fv^{\varphi}(\exists x/\bar{y}\psi) \cup \{x\}$.

Semantics. Let $\mathfrak{U}$ be a structure of signature τ with domain $|\mathfrak{U}|$, and let X be a set of variables. By *valuation* of X in $\mathfrak{U}$ we mean a function $\bar{a} \in |\mathfrak{U}|^X$. For any variable x and any element $b \in |\mathfrak{U}|$, let $\bar{a}(x/b) = \bar{a} \upharpoonright (X - \{x\}) \cup \{(x, b)\}$. Note that the variable x is not necessarily in X.

To each formula φ and any nonempty set of valuations $V \subseteq |\mathfrak{U}|^X$, where $Fv(\varphi) \subseteq X$, we associate a game $G(\mathfrak{U}, \varphi, V)$ [14] between two players: $\exists$ and $\forall$.

The Game $G(\mathfrak{U}, \varphi, V)$**.** We assume that a third agent, the "nature", chooses from V the initial valuation needed to start the game. After the first random move, the players consider consecutively a subformula ψ of φ and an associated valuation $\bar{a}_{\psi} \in |\mathfrak{U}|^{X \cup Fv^{\varphi}(\psi)}$. One of the players chooses the next subformula to play or modify the received valuation, according to the following rules:

1. When the game starts player $\exists$ is the verifier, and player $\forall$ is the falsifier.
2. If ψ is of the form $\neg\psi'$ the game proceeds with the subformula ψ', $\bar{a}_{\psi} = \bar{a}_{\neg\psi}$, and the players switch their roles.
3. If ψ is of the form $\exists x/\bar{y}\psi'$ then the verifier chooses an element b from $|\mathfrak{U}|$. The game proceeds with the subformula ψ' and $\bar{a}_{\psi'} = \bar{a}_{\psi}(x/b)$.
4. If ψ is of the form $\psi_1 \vee \psi_2$ then the verfier chooses $m \in \{1, 2\}$. The game proceeds with ψ_m and $\bar{a}_{\psi_m} = \bar{a}_{\psi}$.
5. If ψ is atomic the game ends and a final valuation $\bar{a}_{\psi}$ is obtained. If $\mathfrak{U} \models \varphi[\bar{a}_{\psi}]$ (classically) then the verifier wins. If $\mathfrak{U} \not\models \varphi[\bar{a}_{\psi}]$ then the falsifier wins.

Strategies. Let $\bar{a}, \bar{b} \in |\mathfrak{U}|^X$ and let Y be set of variables. We say that $\bar{a}$ and $\bar{b}$ *coincide out of* Y if $\bar{a}(x) = \bar{b}(x)$ for all $x \in X - Y$.

[14] Defining the game with respect to a set of valuations V instead of a single valuation is needed to have a truly compositional game semantics and the correct notion of game theoretical equivalence.

Given a set of valuation V with domain X, a function $f : V \to |\mathfrak{U}|$ is *independent of* Y if $f(\bar{a}) = f(\bar{b})$ whenever $\bar{a}$ and $\bar{b}$ coincide out of Y. In the opposite case, we say that f *depends on* Y.

A subformula ψ of φ is positive (negative) if it is in the scope of an even (odd) number of negations. $Sub^+(\varphi)$ ($Sub^-(\varphi)$) denotes the set of positive (negative) non-atomic subformulae of φ. [15]

A *strategy* for $\exists$ in the game $G(\mathfrak{U}, \varphi, V)$ is a set of functions $S = \{f_\psi\}_{\psi \in Sub^+\varphi}$ such that

- $f_\psi : |\mathfrak{U}|^{X \cup Fv^\varphi(\psi)} \to |\mathfrak{U}|$ when $\psi = \exists x/\bar{y}\psi'$, f_ψ is independent of $\bar{y}$.
- $f_\psi : |\mathfrak{U}|^{X \cup Fv^\varphi(\psi)} \to \{1, 2\}$ when $\psi = \psi_1 \vee \psi_2$.

A *strategy* for $\forall$ is defined similarly, as a set of functions $\{f_\psi\}_{\psi \in Sub^-(\varphi)}$.

A player *applies* a strategy $S = \{f_\psi\}_{\psi \in Sub^+\varphi}$ if he chooses the element $f_\psi(\bar{a}_\psi)$ when the subformula $\psi = \exists x/\bar{y}\psi'$, and chooses the first (second) disjunct if $f_\psi(\bar{a}_\psi) = 1$ (respectively, 2) when $\psi = \psi_1 \vee \psi_2$.

S is a *winning strategy* for a player in the game $G(\mathfrak{U}, \varphi, V)$ if the player wins any play of the game in which he applies S.

Game Satisfaction. Define for any formula φ, structure $\mathfrak{U}$, and non empty set of valuation $V \subseteq |\mathfrak{U}|^X$, $Fv(\varphi) \subseteq X$:

1. φ is *true in* $\mathfrak{U}$ *under* V ($\mathfrak{U} \models^+_G \varphi[V]$) if and only if $\exists$ has a winning strategy in the game $G(\mathfrak{U}, \varphi, V)$.

2. φ is *false in* $\mathfrak{U}$ *under* V ($\mathfrak{U} \models^-_G \varphi[V]$) if and only if $\forall$ has a winning strategy in the game $G(\mathfrak{U}, \varphi, V)$.

3. φ is *undetermined in* $\mathfrak{U}$ *under* V if and only if non-of the players has a winning strategy in the game $G(\mathfrak{U}, \varphi, V)$.

The following lemma shows basic facts about game satisfaction and its compositional character. It gives us some clues on how to prove a negation normal form theorem. First we need additional definitions.

A set of valuation V is *independent of* $y_1, ..., y_n$ *for* x if whenever $\bar{a}, \bar{b} \in V$ coincide out of $\{y_1, ..., y_n, x\}$ then $\bar{a}(x) = \bar{b}(x)$.
If $V \subseteq |\mathfrak{U}|^X$ and x is a variable (not necessarily in X), let

$$V^x = \{\bar{a}(x/b) : \bar{a} \in V, b \in |\mathfrak{U}|\}$$

V^x - the expansion of V to $X \cup \{x\}$.

[15] Note that the notion positive/negative formula helps to keep track of the verifier/falsifier roles.

Lemma 3.5

a) For atomic φ,
$\mathfrak{U} \models^+_G \varphi[V]$ *if and only if* $\mathfrak{U} \models \varphi[\bar{a}]$ *for all* $\bar{a} \in V$.
$\mathfrak{U} \models^-_G \varphi[V]$ *if and only if* $\mathfrak{U} \not\models \varphi[\bar{a}]$ *for all* $\bar{a} \in V$.

b) $\mathfrak{U} \models^+_G \neg\varphi[V]$ *if and only if* $\mathfrak{U} \models^-_G \varphi[V]$.
$\mathfrak{U} \models^-_G \neg\varphi[V]$ *if and only if* $\mathfrak{U} \models^+_G \varphi[V]$.

c) $\mathfrak{U} \models^+_G (\varphi_1 \vee \varphi_2)[V]$ *if and only if* $\mathfrak{U} \models^+_G \varphi_i[V_i]$ *for some* V_i, $i = 1, 2$, *such that* $V = V_1 \cup V_2$.

d) $\mathfrak{U} \models^-_G (\varphi_1 \vee \varphi_2)[V]$ *if and only if* $\mathfrak{U} \models^-_G \varphi_i[V]$, $i = 1, 2$.

e) $\mathfrak{U} \models^+_G \exists x/\bar{y}\varphi[V]$ *if and only if* $\mathfrak{U} \models^+_G \varphi[V_x]$ *for some set of valuations* $V_x \subseteq V^x$, V_x *independent of* $\bar{y}$ *for* x.

f) $\mathfrak{U} \models^-_G \exists x/\bar{y}\varphi[V]$ *if and only if* $\mathfrak{U} \models^-_G \varphi[V^x]$.

g) If φ *is a first-order sentence then for any structure* $\mathfrak{U}$*:*

$$\mathfrak{U} \models^+_G \varphi \quad iff \quad \mathfrak{U} \models \varphi$$

Game Equivalence. $\varphi \equiv \psi$, φ and ψ are *G-equivalent*, if and only if for any set V of valuations of $Fv(\varphi) \cup Fv(\psi) \subseteq X$ in a structure $\mathfrak{U}$, we have both:

$$\mathfrak{U} \models^+_G \varphi[V] \Leftrightarrow \mathfrak{U} \models^+_G \psi[V], \text{ and}$$
$$\mathfrak{U} \models^-_G \varphi[V] \Leftrightarrow \mathfrak{U} \models^-_G \psi[V]$$

Substituting G-equivalent formulae in a given L_{ii} formula yields a G-equivalent formula (Substitution Principle). The following two theorems are the results of all the definitions above:

Theorem 3.6 *(Henkin Quantifiers and L_{ii})*
For any Henkin quantifier Q there exists a quantifier prefix of the language L_{ii}, $\bar{Q}$ such that for any first order formula φ, arbitrary structure $\mathfrak{U}$ and a valuation $\bar{a}$ in $\mathfrak{U}$ we have: $\mathfrak{U} \models Q\varphi[\bar{a}]$ *if and only if* $\mathfrak{U} \models^+_G \bar{Q}\varphi[\{\bar{a}\}]$.

Theorem 3.7 *(Σ^1_1 Equivalence)*
For any formula ϕ of L_{ii} of signature τ and set of variables X there is a Σ^1_1-translation ϕ^* of the same signature such that for any structure $\mathfrak{U}$ and $V \subseteq |\mathfrak{U}|^X$:
$\mathfrak{U} \models^+_G \phi[V]$ *if and only if* $\mathfrak{U} \models \phi^*[\bar{a}]$ *for all* $\bar{a} \in V$.

References

[1] J. W. Addison and S. C. Kleene, 1957. A Note on Function Quantification. Proceedings Amer. Math. Soc. 8, 1002 - 1006.

[2] A. Blass and Y. Gurevich, 1986. Henkin Quantifiers and Complete Problems, Annals of pure and applied logic 32 1-6.

[3] H. B. Enderton, 1970. Finite Partially-Ordered Quantifiers, Zeitschr. f. math. Logik und Grundlagen d. math. Bd 16, pg.393-397

[4] R. Fagin, 1974. Generalized First-Order Spectra and Polynomial-Time Recognizable sets, in: R.Karp,ed., Complexity of Computation, SIAM-AMS Proc. 7 pg. 43-73.

[5] Y. Gurevich, 1988. Logic and the Challenge of Computer Science, Börger E, Trends in Theoretical Computer Science, Computer Science press Inc, Rockville, Maryland USA, pg. 1-58.

[6] L. Henkin, 1961. Some Remarks on Infinitely Long Formulas, in: Infinitistic Methods (Warsaw) 167-183.

[7] J. Hintikka, 1996. The Principles of Mathematics Revisited, Boston University, Cambridge University Press.

[8] J. Hintikka, 1974. Quantifiers vs. Quantification Theory, Linguistic Inquiry 5, pg. 153-177.

[9] J. Hintikka, 1997. Game Theoretical Semantics, Handbook of Logic and Language, Elsevier science, pg. 361-410.

[10] J. Hintikka and J. Kulas, 1983. The Game of Language, D. Reidel, Dordrecht (1983).

[11] J. Hintikka and G. Sandu, 1989. Informational Independence as a Semantical Phenomenon, in J. E. Fenstad et al (eds) Logic, methodology and Philosophy of Science VIII, Elsevier, pp. 571-589.

[12] W. Hodges in The Handbook of Philosophical Logic, 1983. Volume I.

[13] W. Hodges, 1997. Compositional Semantics For a Language Of Imperfect Information, Log. J. IGPL, 5(4):539-563 (electronic).

[14] Immerman N, 1983. Language Which Capture Complexity Classes, in: 15th Symposium on Theory of Computing, Association for Computing Machinery, pg. 347-354

[15] M. Krynicki, 1993. Hierarchies of Partially Ordered Connectives and Quantifiers, in: Math. Log. Quart. 39 pg. 287-294.

[16] M. Krynicki and J. Väänänen, 1989. Henkin and Function Quantifiers. In Annals of pure and applied logic 43, pg. 273 - 292.

[17] P.Lindström, 1966. First Order Logic With Generalized Quantifiers, Theoria 32, pg. 186-195.

[18] A. Mostowski, 1957. On a Generalization of Quantifiers, Fund. Math. 44, pg. 12-36.

[19] G. Sandu, On the Logic of Informational Independence and its Applications In J.of Philosophical Logic 22:29-60 1993

[20] G. Sandu and J. Väänänen, 1992. Partially Ordered Connectives, Z. für Math. Logic und Grundlagen der Math. 38, pg. 361-372.

[21] G. Sandu, 1997. The Logic of Information Independence and Finite Models, in J. of the IGPL, Vol 5 No. 1, pg. 79-95.

[22] G. Sandu, 1997. IF-Logic and Truth-Definition, J. of Philosophical Logic 00: 1-21.

[23] G. Sandu and Tapani Hyttinen, 1999. If Logic and Foundations of Mathematics. http://www.valt.helsinki.fi/kfil/sandu.htm

[24] C. Smorynski, 1977. The Incompleteness Theorems, in "The Handbook of mathematical logic".

[25] N. Tennant, 1990. Natural Logic. Edinburg University Press, 2nd ed., pg. 35-37.

[26] J. Väänänen, 1999. On the Semantics of Informational Independence.

[27] J. Van Benthem, 1998. In a course on games in logic given at the Dutch academy of science, Amsterdam. Slides are available on the Web at: http://turing.wins.uva.nl/ johan/Phil.298.html

[28] W. Walkoe, 1970. Finite Partially-Ordered Quantification, J. Symbolic Logic 35, pg. 535-555.

[29] X. Caicedo and M. Krynicki, 1991. Quantifiers for Reasoning With Imperfect Information and Σ^1_1-Logic.