Investigating the basic notions of Hintikka's independence friendly logic

TU/e technische universiteit eindhoven

CS-Report 01-016

Investigating the basic notions of Hintikka's Independence Friendly Logic

Francien Dechesne

/ department of mathematics and computer science

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science

Investigating the basic notions of Hintikka's Independence Friendly Logic

by

Francien Dechesne

01/16

ISSN 0926-4515

Reports are available at:
http://www.win.tue.nl/inf/onderzoek

Computer Science Reports 01/16
Eindhoven, November 2001

Investigating the basic notions of Hintikka's Independence Friendly Logic

Francien Dechesne

November 26, 2001

Abstract

This paper presents Game Theoretical Semantics (GTS) and Independence Friendly logic (IF-logic), as introduced by Jaakko Hintikka in [Hin96], on a basic level. We describe some aspects of IF-logic with GTS: the expressive power coincides with Σ_1^1, the law of the excluded middle does not hold, there are at least two candidates for the notion of equivalence.

We aim to stay as close to Hintikka's definitions as possible. Based on these definitions, we show that the syntax of IF-logic should be extended to ensure negation normal form, we discuss an attempt to define implication in IF-logic, and we argue that the real basis for the notion of truth in IF-logic seems to be Skolemization rather than the game theoretical concepts of GTS.

1 Introduction

In his book *The Principles of Mathematics Revisited* ([Hin96]), Hintikka aims to do what Russell did almost a century ago in *The Principles of Mathematics* (1903), which Hintikka paraphrases as follows: "to examine the conceptual problems that arise in the foundations of logic and mathematics, expose the difficulties in the earlier views, and by so doing try to find guidelines for the right approach." ([Hin96], p. vii)
The book can be criticized for its strong claims, which are not always supported by sufficient evidence, and for its misprints and errors. In fact, this is what happens in most of the reviews that have appeared.[1] But still, the basic ideas deserve further investigation.

Hintikka proposes a new logic to be used for the Foundations of Mathematics: (Information) Independence Friendly logic, IF-logic for short. In the book, he claims the following properties for IF-logic:

- it is not compositional (see, for example, pp. 106-112);

[1] The most critical example is probably [Ten98].

- the law of the excluded middle does not hold (e.g. p. 132);
- every IF-first order formula can be translated into a (classical) Σ_1^1-formula *and* vice versa (pp. 61-63);
- (therefore) a truth predicate can be defined on first order level (p. 116);
- the compactness theorem, separation theorem (in a strengthened form), downward Löwenheim-Skolem theorem, and Beth's definability theorem all hold (pp. 59-61);
- the class of valid formulas of the new logic is not axiomatizable, although the class of inconsistent formulas is (pp. 66-68).

Hintikka distinguishes three functions for logic: logic as a means of expressing (mathematical) propositions ('the descriptive function'), logic as the study of relations of logical consequence ('the deductive function'), and logic as a medium for axiomatic set theory. Hintikka considers the descriptive function to be the most important for the foundations of mathematics. Because the expressive power of IF-logic exceeds the expressive power of classical first order logic, Hintikka believes that IF-logic could open new directions in the foundations of mathematics.

This report aims to investigate Hintikka's proposal on a basic level. Before we discuss some of the claims mentioned above, we will first introduce the two main building blocks of Hintikka's proposal: Game Theoretical Semantics and Independence Friendly logic.

1.1 Preliminary remarks

In this paper 'first order logical language' means: a first order logical language containing the connectives '$\wedge, \vee$', the quantifiers '$\forall, \exists$', and a negation sign '$\sim$'. The connective '$\rightarrow$' is assumed not to be a primitive connective in the language; implication should be defined in terms of the other connectives and the negation sign.

The following notational conventions are used. The Greek letters φ and ψ are used to indicate first order formulas; their capitals Φ and Ψ to indicate second order formulas. In the examples, P, Q and R are used as predicate symbols (with no fixed arity), the binary relation symbol '$=$' is used with the standard interpretation; c and k are used as individual constants, x, y, z, u, v as individual variables and f and g as function variables.

Where in this paper the symbol '$\rightarrow$' occurs in a second order formula, it is to be read in the usual definition of material implication. Furthermore, in second order formulas (which will always be interpreted by 'classical' semantics), we will write '$x \neq y$' rather than '$\sim(x = y)$', and use $\neg$ as negation symbol rather than '$\sim$'.

Recall that a closed formula is called a **sentence**.

In this paper, we aim to stay close to the ideas as presented in [Hin96]. Some of the sections in this paper are concluded by one or more comments and questions concerning these ideas, indicating possible directions for future research.

2 Game Theoretical Semantics

The game theoretical approach to semantics can be used for a wide range of logical systems. It has been used since the early 1960's for the semantics of infinitary languages, and for the analysis of natural language and dialogue structures ([HS97], p. 51-52). On a less formal level, the interpretation of logical formulas as a question-answer game has a long history. A familiar example from mathematics is the explanation of the notion of continuity of a function f in x ("if you tell me how close you want $f(y)$ to be to $f(x)$, then I tell you how close y has to be to x").

Hintikka argues that the combination of game theory and logic is very natural and even unavoidable.[2]

2.1 Semantical games

When describing the concepts of Game Theoretical Semantics (GTS), we will use the following notation: if φ is a (classical) first order formula, and M a model suitable for the language of φ, then

- $FV(\varphi)$ denotes the set of variables occurring free in φ;
- v is a **valuation in** M if v is a *partial* function from the set of variables to $Dom(M)$. For every valuation v in M and $a \in Dom(M)$, we define a valuation in M

$$v[x/a] \overset{d}{=} \begin{cases} v \cup \{\langle x, a\rangle\} & \text{if } x \notin Dom(v) \\ (v \setminus \{\langle x, v(x)\rangle\}) \cup \{\langle x, a\rangle\} & \text{if } x \in Dom(v) \end{cases}$$

  If v is a valuation in M and $FV(\varphi) \subset Dom(v)$, we say that v **is a valuation for** φ **in** M. We use the notation v_φ^M to denote a valuation for φ in M. However, if it is clear from the context in which model the values for the free variables are chosen, we write v_φ rather than v_φ^M.

[2] See, for example [Hin96], p. 29.

Definition 1: semantical games
Given a first order formula φ, a suitable model M, a valuation v_φ for ϕ in M, and $k \in \{0,1\}$, we associate a 2-player **semantical game**, with players P_0 and P_1:[3]

$$G_M(\varphi, v_\varphi, k).$$

The game is played by the following rules:

rule	if φ of the form	move is choice of	by player	next stage
$(\forall)$	$\forall x[\psi]$	$a \in Dom(M)$	P_{1-k}	$G_M(\psi, v_\varphi[x/a], k)$
$(\exists)$	$\exists x[\psi]$	$a \in Dom(M)$	P_k	$G_M(\psi, v_\varphi[x/a], k)$
$(\wedge)$	$\psi_1 \wedge \psi_2$	$i \in \{1,2\}$	P_{1-k}	$G_M(\psi_i, v_\varphi, k)$
$(\vee)$	$\psi_1 \vee \psi_2$	$i \in \{1,2\}$	P_k	$G_M(\psi_i, v_\varphi, k)$
$(\sim)$	$\sim\psi$	(no move)	–	$G_M(\psi, v_\varphi, 1-k)$
(At)	atomic formula	(no move)	–	(game over) P_k wins if $(M, v_\varphi) \models \varphi$; P_{1-k} wins otherwise

If φ is a sentence (and hence: $FV(\varphi) = \emptyset$), we abbreviate $G_M(\varphi, \emptyset, 1)$ to

$$G_M(\varphi)$$

and call this: **the semantical game for φ in M.** ◁

We will use the rest of this section to give an informal explanation of this definition.

In $G_M(\varphi, v_\varphi, k)$, the parameter k determines the roles of the players (with respect to φ): P_k has the role of 'Verifier' with respect to φ, which means he plays to show that

$$(M, v_\varphi) \models \varphi,$$

while P_{1-k} has the role of 'Falsifier' with respect to φ, and plays to show that

$$(M, v_\varphi) \not\models \varphi,$$

The player in the role of Verifier makes the moves associated with $\exists$ or $\vee$, the player in the role of Falsifier makes the moves associated with $\forall$ or $\wedge$.

[3] In his appendix to [Hin96] (p. 255), Gabriel Sandu calls the two players Nature and Myself. We choose the more neutral names P_0 and P_1, in order to make it easier to indicate the respective roles of the players during the game.

We will usually discuss the semantical game $G_M(\varphi) = G_M(\varphi, \emptyset, 1)$ for some specific first order sentence φ and appropriate model M. In this situation, P_1 is called **the Verifier** (with respect to φ), and player P_0 **the Falsifier.**
After each stage of a game $G_M(\varphi)$ for a given sentence φ and an appropriate model M, this game continues as a game $G_M(\psi, v_\psi, k)$ for some subformula ψ of φ, and valuation v_ψ for ψ in M.
If ψ is atomic, the game ends and the winner is determined on the basis of the truth value of ψ in M with the valuation v_ψ.[4]
If during the game a negation sign ('$\sim$') is encountered, none of the players makes a move. Instead, the roles of the players are exchanged with respect to the *subformula* of φ behind the negation sign.[5]
Note that two successive changes of roles do not affect the course of the game: a game of the form $G_M(\sim\sim\varphi, v_\varphi, k)$ continues as $G_M(\sim\varphi, v_\varphi, 1-k)$, which in turn continues as $G_M(\varphi, v_\varphi, 1-(1-k)) = G_M(\varphi, v_\varphi, k)$. We conclude that in GTS like in classical logic, but unlike in intuitionistic logic, double negation cancels out.

In [Hin98], a different treatment for negation in GTS is proposed:

> "Negation can be dealt with by pushing the negation signs as deep into the formulas as they can go". ([Hin98], p. 309, item (3.9))

In other words, before the start of the semantical game for a formula φ, we should rewrite the formula into negation normal form, i.e. with the negation signs prefixed to atomic formulas only. This leads us to the question whether (with our definition of the negation rule) the games for φ and for its negation normal form correspond in some sense.
In the standard semantics of first order logic, De Morgan's laws enable us to translate an arbitrary formula into an equivalent formula in negation normal form:[6]

$$\begin{aligned} \sim\forall x[\psi] &\equiv \exists x[\sim\psi] \\ \sim\exists x[\psi] &\equiv \forall x[\sim\psi] \\ \sim(\psi_1 \wedge \psi_2) &\equiv \sim\psi_1 \vee \sim\psi_2 \\ \sim(\psi_1 \vee \psi_2) &\equiv \sim\psi_1 \wedge \sim\psi_2 \end{aligned}$$

[4] Note that a semantical game for a traditional first order formula is always won by one of the players: after finitely many moves an atomic formula is reached, and given the valuation determined by the moves of the players, the truth value of this atomic formula in the model is fixed.

[5] The role each player has with respect to φ itself remains the same, although the role may be changed with respect to a subformula of φ during the game.

[6] Note that the first rule is intuitionistically invalid. In intuitionistic logic not every formula has an equivalent in negation normal form.

It turns out that these laws also hold in GTS, because the changes of the quantifiers and the connectives correspond precisely to the changes of the roles of the players.[7] To see this, compare for instance $G_M({\sim}\forall x[\psi], v, k)$ and $G_M(\exists x[{\sim}\psi], v, k)$, where v is some fixed valuation in M with domain containing $FV(\psi) \setminus \{x\}$, and $k \in \{0,1\}$:

$$\begin{array}{ll}
G_M({\sim}\forall x[\psi], v, k) & G_M(\exists x[{\sim}\psi], v, k) : \\
\Downarrow & \mathsf{P_k} \text{ chooses } a \in Dom(M) \\
G_M(\forall x[\psi], v, 1-k) : & \Downarrow \\
\mathsf{P_{1-(1-k)}} = \mathsf{P_k} \text{ chooses } a \in Dom(M) & G_M({\sim}\psi, v[x/a], k) \\
\Downarrow & \Downarrow \\
G_M(\psi, v[x/a], 1-k) & G_M(\psi, v[x/a], 1-k)
\end{array}$$

The first two stages of both games are similar: a domain element is chosen by player $\mathsf{P_k}$ and the roles of the players are exchanged. After that, the games continue in exactly the same way.

This example illustrates how writing a first order formula into negation normal form complies with the definition of the negation rule as the change of roles of the players. Hence, the alternative treatment of negation as prescribed in [Hin98] does not lead to different semantical games.

Remarks and further investigation:

- We are aware of the fact that our 'definition' of semantical games, does not thoroughly *define* the concept of semantical game as a mathematical object. It is rather a *description* and it heavily depends on intuitions on what constitutes a game in general: players, rules, moves, winning conditions etc.

  A prerequisite for use of GTS in the foundations of mathematics is to choose a formalization of the concept of 'game', in which framework the concept of 'semantical game' can be formulated. The lack of a formal framework shows itself to be a particularly big handicap when we attempt to 'define' the concept of strategy in section 2.3.
- As demonstrated, the semantical games for different formulas like ${\sim}\forall x[\psi]$ and $\exists x[{\sim}\psi]$ are very similar. We tend to talk about these games as being 'the same game'. We could try to formalize this intuitive notion by defining an equivalence relation on the class of semantical games.

2.2 Tree representation

When discussing the semantical game for a formula in a given model, it can be useful to have an overview of the different possible courses of the game. For this purpose, we associate with a semantical game $G_M(\varphi, v_\varphi, k)$ a labeled tree

$$T_M[\varphi, v_\varphi, k],$$

[7] See also [Hin96], p. 133-134.

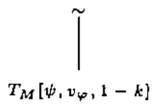
which we define inductively below.

Definition 2: tree representation
Let φ, M, v_φ and k be as in definition 1. We distinguish the following cases:

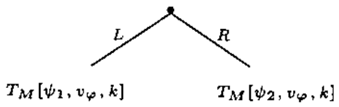
(At) **φ is an atomic formula:** then $T_M[\varphi, v_\varphi, k]$ is defined by

$$\begin{cases} \bullet \quad \text{if} \quad \mathsf{P}_1 \text{ wins, i.e.} \\ \qquad\qquad k = 1 \text{ and } (M, v_\varphi) \models \varphi \\ \qquad\qquad \text{or} \\ \qquad\qquad k = 0 \text{ and } (M, v_\varphi) \not\models \varphi \\ \circ \quad \text{if} \quad \mathsf{P}_0 \text{ wins, i.e.} \\ \qquad\qquad k = 1 \text{ and } (M, v_\varphi) \not\models \varphi \\ \qquad\qquad \text{or} \\ \qquad\qquad k = 0 \text{ and } (M, v_\varphi) \models \varphi \end{cases}$$

$(\sim)$ **$\varphi = \sim\psi$ for some formula ψ:** then $T_M[\varphi, v_\varphi, k]$ is defined by

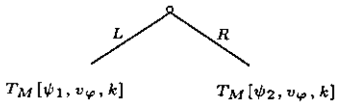


$(\vee)$ **$\varphi = \psi_1 \vee \psi_2$ for formulas ψ_1 and ψ_2:** then, if $k = 1$, $T_M[\varphi, v_\varphi, k]$ is defined by



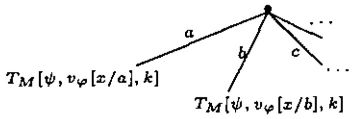
If $k = 0$, the tree $T_M[\varphi, v_\varphi, k]$ is defined similarly, except that the top node is a circle ($\circ$) instead of a bullet.

$(\wedge)$ **$\varphi = \psi_1 \wedge \psi_2$ for formulas ψ_1 and ψ_2:** then, if $k = 1$, $T_M[\varphi, v_\varphi, k]$ is defined by



If $k = 0$, the tree $T_M[\varphi, v_\varphi, k]$ is defined similarly, except that the top node is a bullet ($\bullet$) instead of a circle.

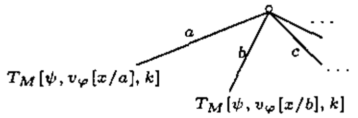
(∃) $\varphi = \exists x[\psi]$ **for some formula** ψ: then, if $k = 1$, $T_M[\varphi, v_\varphi, k]$ is defined by


a
b
c
. . .
. . .
$T_M[\psi, v_\varphi[x/a], k]$
$T_M[\psi, v_\varphi[x/b], k]$

with one branch for every $a \in Dom(M)$.[8]

If $k = 0$, the tree $T_M[\varphi, v_\varphi, k]$ is defined similarly, except that the top node is a circle (∘) instead of a bullet.

(∀) $\varphi = \forall x[\psi]$ **for some formula** ψ: then, if $k = 1$, $T_M[\varphi, v_\varphi, k]$ is defined by


a
b
c
. . .
. . .
$T_M[\psi, v_\varphi[x/a], k]$
$T_M[\psi, v_\varphi[x/b], k]$

with one branch for every $a \in Dom(M)$.

If $k = 0$, the tree $T_M[\varphi, v_\varphi, k]$ is defined similarly, except that the top node is a bullet (•) instead of a circle.

If φ is a sentence and M a suitable model for φ, we will write $T_M[\varphi]$ for $T_M[\varphi, \emptyset, 1]$. ◁

If we call the •-nodes **1-nodes**, and the ∘-nodes **0-nodes**, we can formulate the following claim for our tree representation of semantical games: the k-nodes correspond to the stages in $G_M(\varphi)$ where $\mathsf{P_k}$ makes a move, and the k-leafs to the winning end positions for $\mathsf{P_k}$ in $G_M(\varphi)$.[9]

As an example, in figure 1 the tree diagram is drawn corresponding to the semantical game for $\varphi = \forall x[P(x)] \vee \sim \forall x[P(x)]$ in the model $M = \langle\{0, 1, 2\}, P \mapsto \text{"even"}\rangle$.

[8] Hence, if $Dom(M)$ is uncountable, $T_M[\varphi, v_\varphi, k]$ has uncountably many branches.

[9] It does not seem hard to prove this property, but it does require a more formal approach to game theoretical concepts like 'move', or 'run of a game' than we provide in this paper.

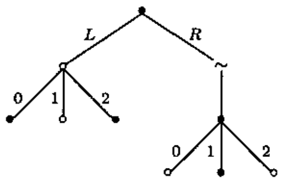


Figure 1: game-tree for $G_M(\forall x[P(x)]\vee{\sim}\forall x[P(x)])$

2.3 Truth in GTS

As intended, the tree representation defined in the previous section is an extensive one: it shows all the possible courses of a semantical game in a given model. But not all courses will be of interest for the players, assuming each player plays to win. This is where the notion of strategy appears on the scene.

Intuitively, a strategy is a method prescribing what to do in order to reach a goal. A standard game theoretical definition does not exist. Hintikka says the following about his interpretation ([Hin96], p. 27): "In my sense, a strategy for a player is a rule that determines which move that player should make in any possible situation that can come up in the course of the play". For the time being, to stay close to this statement, we define the notion of strategy as follows:[10]

Definition 3: strategies
Let φ be a first order sentence, and M a model suitable for the language of φ. A **strategy-function for a move** of player $\mathsf{P_k}$ in the game $G_M(\varphi)$ is a function f, whose arguments are the values for variables of φ, that were previously chosen by $\mathsf{P_{1-k}}$. If the move is induced by a connective ($\vee$ or $\wedge$), f ranges over $\{1,2\}$; otherwise, if the move is induced by a quantifier, f ranges over $Dom(M)$.
A **strategy for player $\mathsf{P_k}$ in $G_M(\varphi)$** is a finite set of strategy-functions for $\mathsf{P_k}$, with a fixed bijective correspondence to the moves $\mathsf{P_k}$ possibly has to make in $G_M(\varphi)$.[11]
A strategy for player $\mathsf{P_k}$ in $G_M(\varphi)$ is called **winning** if $\mathsf{P_k}$ wins every run of $G_M(\varphi)$ by making the occurring moves as prescribed by the corresponding strategy-function in the strategy. $\triangleleft$

[10] In his examples, Hintikka only treats strategies for the player in the role of Verifier. We don't see any objection however to treat strategies for the player in the role of Falsifier uniformly.

[11] In other words: for each move $\mathsf{P_k}$ possibly has to make in $G_M(\varphi)$, a strategy-function is available, *and* $\mathsf{P_k}$ knows which function should be applied in which move. Note that, as there are only finitely many quantifiers and connectives in a first order formula, the number of moves for both players is finite as well.

In section 5.3, we come back to Hintikka's concept of strategy and argue that it seems to be more closely related to the logical notion of Skolemization[12] than to our intuitions about playing games.

The notion of winning strategy is the basis for the definition of truth in game theoretical semantics:

Definition 4: truth and falsity in GTS
If φ is a first order sentence and M a model suitable for the language of φ, then[13]

(t) φ is **true in** M if and only if there exists a winning strategy for the Verifier in $G_M(\varphi)$;

(f) φ is **false in** M if and only if there exists a winning strategy for the Falsifier in $G_M(\varphi)$.

$\lhd$

Combining this definition with the definition of strategy, we see that the **truth (and falsity) condition** for a sentence is a statement containing existential quantifications over functions. This will be discussed in more detail in sections 4.1 and 5.3.
It is important to notice that, in general, the non-existence of a winning strategy for one player in a 2-player game, does not automatically imply the existence of a winning strategy for the other player. We will return to this interesting aspect of the game-theoretical truth definition in section 4.3.

Remarks and further investigation:

- We would like to give a formal definition of strategy, making the number of strategy-functions for each player explicit in terms of the syntax of φ, as well as the arguments for each strategy-function.
- This is needed to be able to give a direct formal proof that game-theoretical truth and truth in the classical sense coincide for standard first order formulas.
- Should a previous choice by the opponent concerning a connective ($\wedge$ or $\vee$) be an argument in a strategy-function? One could say that a connective-move determines *which* strategy-functions will be used in the rest of the game, rather than being an argument for them. Hintikka is not explicit about this (and avoids the problem by mainly considering sentences in prenex normal form).

[12] See section 4.1.

[13] Throughout this paper, we use this typography to distinguish the truth value of a formula in the game theoretical sense from the truth value in the traditional (Tarskian) sense.

2.4 Example: $\forall x \exists y[x \neq y]$

To illustrate the notions introduced above, we consider the semantical game for a simple classical first order sentence on two different models. Let φ be the formula $\forall x \exists y[\sim(x = y)]$. We draw the game tree $T_M[\varphi]$ for $M_1 = \langle\{0,1\},=\rangle$ and $M_2 = \langle\{0\},=\rangle$:

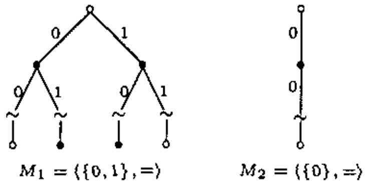

0
1
0
1
0
1
~
~
~
~
$M_1 = \langle\{0,1\},=\rangle$
0
0
~
$M_2 = \langle\{0\},=\rangle$

Figure 2: $T_M[\forall x \exists y[\sim(x = y)]]$ for two different models M.

In the latter case, a one element model, the formula is `false`: the only possible course of the game ends in a win for the Falsifier, P_0. In the first case the formula is `true`: in the second and last move of the game the Verifier, P_1, can choose a value for y different from the value for x previously chosen by the Falsifier, and thereby win the game.

For arbitrary models, using the notions of the previous section, the truth condition for this formula can be formulated as the classical second order formula:

$$\exists f \forall x[x \neq f(x)].$$

On the model $\langle\{0,1\},=\rangle$, the (only[14]) winning strategy for the Verifier can be expressed as the function $f : \{0,1\} \to \{0,1\}$, defined by $f(0) = 1$ and $f(1) = 0$.
It is crucial that the Verifier is allowed to use the value assigned to x, otherwise the function f is unusable as a method prescribing a choice. In other words: the game has to be one of perfect information. Games for classical first order formulas are games of perfect information, contrary to games for Hintikka's (Information) Independence Friendly formulas. These will be introduced in the following chapter.

3 IF-logic

3.1 "Frege's fallacy"

In classical logic (as introduced in Frege's *Begriffsschrift*), the scope of a quantifier is defined as follows: if $\forall x[\psi]$ appears as a (sub)formula, then ψ

[14] In a three-element model we would have several winning strategies for the Verifier.

is called the *scope* of $\forall x$, and similarly for $\exists$. (See, for instance: [Ham88], p. 54.)

In classical logic, scopes are always either nested or non-overlapping, as for example the scopes of $\forall x$ and $\forall z$ in the following two formulas respectively:

$$\forall x \exists y[\psi_1(x,y) \wedge \forall z \exists u[\psi_2(z,u)]];$$

$$\forall x \exists y[\psi_1(x,y)] \wedge \forall z \exists u[\psi_2(z,u)].$$

This results in a restriction of the expressive power of the traditional first order language, as will be shown in the next paragraph. It is exactly this restriction, referred to by Hintikka as *Frege's fallacy*, that IF-logic aims to withdraw.

Consider the first-order sentence

$$\forall x \exists y \forall z \exists u[Q(x,y,z,u)],$$

where Q is a 4-place predicate symbol of the logical language. We can use the process of Skolemization (which will be described in more detail in section 4.1) to translate this first order sentence into a second order formula (introducing the function symbols f and g for the Skolem-functions):

$$\exists f \exists g \forall x \forall z[Q(x,f(x),z,g(x,z))].$$

We will call this second order translation the **Skolemization of** the first order formula.[15]
In the latter formula, the scopes of the two universal quantifiers become explicit: for instance, the function symbol f, representing the choice of y, does not have z as an argument, corresponding to the fact that $\exists y$ was no part of the scope of $\forall z$.

Conversely, considering the similar second order formula

$$\exists f \exists g \forall x \forall z[Q(x,f(x),z,g(z))], \tag{1}$$

the question arises whether this formula is the Skolemization of some first order formula. In classical logic, the answer is negative, because writing

[15] Note that to speak of a second order *equivalent* of the first order formula, we need the axiom of choice (to assert that the truth of the first order formula implies the truth of its Skolemization). Without AC, we can prove that a first order formula is contradictory if and only if its Skolemization is (see [Ham88], p. 71). In other words: without AC we can prove that a first order formula and its Skolemization are *equisatisfiable*, with AC we can prove them to be *equivalent*.

down the two universal quantifiers linearly, their scopes cannot be otherwise than nested. To solve this, we could allow a two-dimensional notation:[16]

$$\left.\begin{matrix}\forall x \exists y \\ \forall z \exists u\end{matrix}\right\} [Q(x,y,z,u)]$$

This 2 × 2-array of quantifiers has been introduced by Henkin in [Hen61], and is usually referred to as the Henkin-quantifier.[17]

Hintikka generalizes this idea by introducing a notation, that not only allows to liberate *quantifiers* from the scope of other quantifiers, but does the same for the *connectives.*

In a game theoretical perspective, we could describe the first order version we have in mind for formula (1) as an imperfect information game: the value for the second position in Q can be based on the value for x, but has to be independent of the value of z, and vice versa for the fourth position. Our classical first order logic however is equipped for perfect information games only. The slash notation, that will be introduced in the following section, frees us from this restriction.

The resulting logic is called: (Information) Independence Friendly logic, or **IF-logic** for short.

3.2 Syntax for IF-logic

In [Hin96], the syntax for IF-first order formulas is given (p. 52) as a procedure to build them from classical first order sentences. We will adopt Wilfrid Hodges's notation and write $(\ldots/x)$ instead of $(\ldots/\forall x)$ ([Hod97], p. 551).

Definition 5: IF-formulas

Let φ be a formula of ordinary first order logic in negation normal form. A formula of IF-first order logic is obtained from φ by any finite number of the following steps:

- If $\exists y[\psi]$ occurs in φ within the scope of a number of universal quantifiers which include $\forall x_1, \forall x_2, \ldots$ then it may be replaced by:

$$(\exists y/x_1, x_2, \ldots)[\varphi]$$

[16] In the prenex form of the usual linear notation, the quantifiers are linearly ordered. In a two dimensional notation, we can express every *partial* ordering of quantifiers. It is usually referred to as *branching quantification.*

[17] By a result from Ehrenfeucht quoted in the same paper ([Hen61], pp. 182), we know that the Henkin-quantifier is not expressible in classical first order logic, and that a logic enriched with the Henkin-quantifier is not axiomatizable.

- If $\psi_1 \vee \psi_2$ occurs in φ within the scope of a number of universal quantifiers which include $\forall x_1, \forall x_2, ...$ then it may be replaced by:

$$\psi_1(\vee/x_1, x_2, ...)\psi_2$$

$\triangleleft$

The objective of this notation is to be able to free existential quantifiers and disjunctions from the scope of universal quantifiers. If $\forall x[\psi]$ occurs as a (sub)formula in IF-logic, the scope of $\forall x$ is no longer simply ψ: the scope of $\forall x$ contains only the existential quantifiers and disjunctions in ψ for which the variable x is not 'slashed out'.

In terms of game theoretical semantics, the slash must be read as follows: the player who makes the move associated with the $\exists$-quantifier or the $\vee$-connective, is not allowed to use the chosen values for the variables under the slash. In other words: a strategy function for that move does not have the values of the variables under the slash as arguments. Semantical games on IF-formulas are, in contrast to semantical games for classical first order formulas, games of **imperfect information**.
To illustrate that this is a significant difference, consider the IF-formula ψ:

$$\forall x \exists y/x[\sim(x = y)].$$

We can build ψ from the classical first order formula φ in section 2.4. The strategy f that was winning for the Verifier in the game $G_M(\varphi)$ on the two-element model is no longer allowed in the semantical game for the IF-formula ψ: the Verifier's strategy-function for y can't have x as an argument. In fact, a strategy for the Verifier in the game for ψ must be a function with no arguments, i.e. a constant. The truth condition for ψ in GTS is the (classical) formula

$$\exists c \forall x[x \neq c],$$

which does not hold (in the classical sense) in any model. Hence, ψ is not `true` in any model.[18]

Hintikka defines the slash notation for formulas in negation normal form[19] only and claims that this is no restriction because "rules for transforming formulas into negation normal form and out are the same in IF first order logic as in its traditional variant" ([Hin96], p. 52). In section 5.1, we show

[18] In fact, in models containing more than one element, ψ is not `false` either; see section 4.3.

[19] Cf. section 2.1, page 6.

that this claim conflicts with his definition of the slash-notation for $\exists$ an $\vee$ only, and that this problem is solved by extending the slash-notation to $\forall$ and $\wedge$ as well.
For the time being, we work with the definition of IF-formulas given above. The negation normal form of the IF-sentences secures that $\exists$- and $\vee$-moves are always made by the (initial) Verifier, P_1. The negation normal form is also needed for the Skolemization procedure in section 4.1.

Remarks and further investigation:

- A definition for the syntax of IF-formulas that does not use classical first order formulas, seems preferable: it would be more direct, and could possibly facilitate a formal definition of scope in IF-logic. Note also that a new kind of free variables emerges in subformulas of the IF-formulas as defined by Hintikka: the unbound variables occurring under a slash of an existential quantifier, that do not occur in the scope of the existential quantifier, as for example the variable y in $\exists x/y[P(x)]$. (Similarly: the unbound variables occurring under the slash of a disjunction, but not in the subformulas connected by the disjunction.)

3.3 An example of the expressive power of IF-logic: infinity

In classical logic there is a number of properties of a model that we cannot express on a first order level (provided we use nothing more than the logical constants, variables and equality). For example: the equicardinality of two predicates, the non well-foundedness of an ordering, the countability or the infinity of the domain of a model.
These properties *can* be expressed by a *second* order formula, using equality and existential quantification over functions. This is exactly what Hintikka claims to have incorporated on a first order level with IF-logic, as we will see in section 4.2.

As an example we give an IF-formula that expresses the infinity of a model M, i.e. the Verifier has a winning strategy for the game on M if and only if the domain is infinite. The domain of a model is infinite exactly if there is an injective, non-surjective function from the domain to itself:

$$\exists f \exists y \forall x_1 \forall x_2 [f(x_1) \neq y \wedge (f(x_1) = f(x_2) \rightarrow x_1 = x_2)].$$

This is expressed by the following IF-first order formula:

$$\begin{gathered}\exists y \forall x_1 \forall x_2 (\exists y_1 / x_2)(\exists y_2 / x_1)[(x_1 = x_2 \rightarrow y_1 = y_2) \wedge (y_1 \neq y) \wedge \\ (y_1 = y_2 \rightarrow x_1 = x_2)].\end{gathered} \tag{2}$$

This formula should be understood as follows: by the use of the slash, the choice by the Verifier for the value of y_i is only allowed to depend on the value chosen by the Falsifier for x_i. Hence, the strategy functions f_i for these

two moves will both be unary functions, just like the injective, non-surjective function whose existence we aim to assert with formula (2).
Indeed, if we skolemize this formula -as will be explained in section 4.1- we get its truth condition:

$$\exists f_1 \exists f_2 \exists c \forall x_1 \forall x_2 [(x_1 = x_2 \rightarrow f_1(x_1) = f_2(x_2)) \wedge (f_1(x_1) \neq c)$$

$$\wedge (f_1(x_1) = f_2(x_2) \rightarrow x_1 = x_2)].$$

The first conjunct forces f_1 and f_2 to denote the same function f; we need the two function symbols f_i because one Skolemfunction cannot occur with different sequences of arguments (x_1 and x_2 in this example).
The last conjunct expresses the injectivity of f. Formula (2) is an improved (the intended?) version of the incorrect formulas Hintikka gives on pages 64 and 187 of [Hin96].

In sections 4.1 and 4.2 we will describe the procedures that we applied here to translate IF-first order formulas into (classical) existential second order formulas (Σ_1^1), and vice versa.

Remarks and further investigation:

- This was an example using the representation of a Henkin-quantifier in IF-logic.[20] It seems harder to find a useful example in which a slashed $\vee$-sign is used.

4 Properties of the combination GTS-IF logic

4.1 Skolemization

Recall that a "Σ_1^1-sentence has the form of a sequence of second order existential quantifiers followed by a first order formula". ([Hin96], p. 61)

Every IF-first order formula can be translated into a Σ_1^1-formula that expresses its truth condition. For this, we will generalize the procedure of Skolemization[21] in order to make it applicable for IF-logic. We write it out informally but in some detail, because of its central position in Hintikka's arguments.

Skolemization for classical first order formulas formalizes the following idea: if for all $x_1, \ldots, x_k$ at least one y exists such that some relation holds on $(x_1, \ldots, x_k, y)$, then there exists a function that assigns to each k-tuple

[20] In fact, the capacity of the Henkin-quantifier to express infinity proves that the Henkin-quantifier is not first order expressible; cf. footnote on page 13.

[21] See for example [Ham88](pp. 70-72).

$(x_1, \ldots, x_k)$ one such y. As remarked in the footnote on page 12, this is justified by the axiom of choice.

Following this idea, we rewrite a first order sentence in negation normal form into a Σ^1_1-sentence as follows: if $\exists y[\psi]$ is a subformula of a sentence φ, that occurs within the scope of the universal quantifiers $\forall x_1, \ldots, \forall x_k$, we choose a new ($k$-ary) function symbol f, delete the existential quantifier $\exists y$, replace all occurrences of y in ψ by $f(x_1, \ldots, x_k)$, and add the second order quantification $\exists f$ as first quantifier to φ. By iteration, we can eliminate all the original first order $\exists$-quantifiers, and obtain a Σ^1_1-formula Φ of the following form:

$$\exists f_1 \ldots \exists f_l \overline{\varphi},$$

where $\overline{\varphi}$ is a first order formula without existential quantification, and in which the function symbols $f_1, \ldots, f_l$ occur with the appropriate arguments.[22]

This procedure can be applied to IF-first order sentences as well. An existential quantification of the form $(\exists y/x_i)$ is excluded from the scope of $\forall x_i$, and hence x_i will not be an argument of the Skolem function we introduce. For example:

$$\forall x_1 \forall x_2 \exists y[\psi(x_1, x_2, y)] \overset{\text{Skolem}}{\dashrightarrow} \exists f \forall x_1 \forall x_2 [\psi(x_1, x_2, f(x_1, x_2))]$$

and:

$$\forall x_1 \forall x_2 (\exists y/x_2)[\psi(x_1, x_2, y)] \overset{\text{Skolem}}{\dashrightarrow} \exists f \forall x_1 \forall x_2 [\psi(x_1, x_2, f(x_1))].$$

In IF-logic the scope of universal quantifiers over disjunctions is also relevant. Hence, the procedure has to be generalized to handle the disjunctions.

If in a sentence φ an $\vee$-connective occurs under the scope of $\forall x_1, \ldots, \forall x_k$, then a choice for the left- or right hand side is made (by the Verifier) depending on the chosen values (by the Falsifier) for $x_1, \ldots, x_k$.

We will translate this decision with a new k-ary function symbol f_i, having $x_1, \ldots, x_k$ as arguments and, let's say, $\{0, 1\}$ as range. Then "$f_i(x_1, \ldots, x_k) =$

[22] It is fairly possible that an existential quantifier occurs outside the scope of any universal quantifier. In this case, the corresponding Skolem-function will have no arguments and hence rather be a *constant*. For notational convenience, we will allow function symbols to denote constants as 0-ary functions.

Another remark to be made is that the order of elimination of the first order existential quantifications in φ, determines the order of the corresponding second order existential quantifications in Φ. A different order of these second order quantifications results in an equivalent second order formula. To make the procedure deterministic, we could introduce a convention, e.g. to eliminate the first order existential quantifiers from left to right.

0" will indicate a choice for the left-hand side, and "$f_i(x_1, \ldots, x_k) = 1$" a choice for the right-hand side.[23]
As with the existential quantifiers, the use of the slash notation for $\vee$ in IF-logic will not change the translation procedure. Variables occurring under the slash will not occur as arguments of the (generalized) Skolem-function. The following example of a simple (IF-)sentence should indicate how the Skolem-procedure for disjunction works:

$$\forall x_1 \forall x_2 [\psi_1(x_1, x_2)(\vee/x_1)\psi_2(x_1, x_2)]$$

becomes

$$\exists f \forall x_1 \forall x_2 [(f(x_2) = 0 \to \psi_1(x_1, x_2)) \wedge (f(x_2) = 1 \to \psi_2(x_1, x_2))].$$

With the generalized Skolemization procedure, we can work through an IF-first order sentence φ until all existential quantifications and all disjunctions are translated in terms of Skolem-functions. The result will be a Σ^1_1-sentence Φ with one function symbol for every existential quantifier or disjunction of the original formula, i.e. with one function symbol for each move of the Verifier. The arguments of those functions reflect the moves of the Falsifier on which he can base his choice.
This observation shows how Skolem functions correspond with Hintikka's notion of strategy[24]: Φ can be read as the truth condition for the original first order formula. This implies that in IF-logic the second order translation *is* equivalent to the first order original. In other words: the axiom of choice is incorporated in IF-logic.[25]

4.2 Σ^1_1 translated to IF-first order

We have just seen how we can treat IF-first order logic as part of Σ^1_1. In this section we demonstrate Hintikka's procedure to translate a Σ^1_1-sentence into IF-first order logic. This leads to the claim that IF-first order logic has precisely the expressive power of Σ^1_1.

The translation procedure can be found in [Hin96], p. 62-63 (with an easily reparable mistake in (3.44): the double arrow should be a single one). An example has already been given in section 3.3, with formula (2).

[23] Note that for a successful application of this method, we need (different) interpretations for the constants 0 and 1 in a model. In other words, the model under consideration should contain at least 2 elements. Hintikka leaves this implicit, but secures these interpretations by assuming that the models contain elementary arithmetic.

[24] Cf. section 2.3.

[25] Cf. footnote on page 12 and [Hin96], p. 40.

Any Σ^1_1-sentence Φ can, according to Hintikka, be written in the following form:

$$\exists f_1 \exists f_2 \ldots \exists f_l \forall x_1 \forall x_2 \ldots \forall x_n [\psi] \tag{3}$$

where ψ is a quantifier-free ordinary first order formula, in which function symbols $f_1, \ldots, f_l$ and variable names $x_1, \ldots, x_n$ occur, and such that the following conditions are satisfied:

- the function symbols do not occur nested;
- each function symbol occurs with only one sequence of arguments. (For example: if f_i is a unary function symbol and $f_i(x_j)$ occurs in φ, then $f_i(x_k)$ with $k \neq j$ does not occur; cf. section 3.3.)

We can bring Φ to this form by applying the necessary number of the following transformations:

1. In the case that Φ contains an existential quantification over predicates, we can rewrite this into a quantification over functions by substituting the predicate by its characteristic function. For example: $\exists P \forall x[P(x)]$ becomes $\exists f \forall x[f(x) = 1]$. (Here again, for successful application of this method, models are assumed to contain equality and interpretations for 0 and 1.)

2. If in Φ two function symbols occur nested, as for example in:

   $$\exists f_1 \exists f_2 \forall x[\varphi(f_2(f_1(x)))],$$

   we can bypass this by introducing an extra individual variable:

   $$\exists f_1 \exists f_2 \forall x_1 \forall x_2[x_2 = f_1(x_1) \rightarrow \varphi(f_2(x_2))].$$

   (For this, models are assumed to contain equality.)

3. If in Φ one function symbol occurs with different sequences of variables, for example:

   $$\exists f_1 \forall x_1 \forall x_2[\varphi(f_1(x_1), f_1(x_2))],$$

   we can bypass this by introducing an extra function symbol:[26]

   $$\exists f_1 \exists f_2 \forall x_1 \forall x_2[(x_1 = x_2 \rightarrow f_1(x_1) = f_2(x_2)) \wedge \varphi(f_1(x_1), f_2(x_2))].$$

   (For this again, models are assumed to contain equality.)

[26] Cf. the example of section 3.3.

Once we have transformed Φ into a Σ^1_1-formula of the form (3), we can translate it into an IF-first order formula by repeating the following procedure for $i = 1, \ldots, l$: let $x_{i_1}, \ldots, x_{i_m}$ be the variables from $x_1, \ldots, x_n$ that do not occur in the sequence of arguments of f_i. Remove the second order quantification $\exists f_i$, and insert the IF-first order quantification $(\exists y_i / x_{i_1}, \ldots, x_{i_m})$ directly left from the quantifier-free part of the formula, ψ.[27] In ψ, we replace every occurrence of f_i (with its sequence of arguments) by y_i.

Thus, Hintikka claims: every Σ^1_1-sentence can be translated into an IF-first order sentence. We end this section with some observations regarding this claim.

Tacitly, prenex normal form for Σ^1_1-sentences is presupposed (and hence, negation is prefixed to quantifier-free subformulas only). Moreover, an arbitrary Σ^1_1-sentence could contain existential quantifications over individual variables: no attention is paid to this in Hintikka's procedure. We assume that he intends the procedure to leave these quantifications unchanged.

We observe that it is no coincidence that the translation from a Σ^1_1-sentence to IF-first order logic works particularly well if the sentence is of the form (3): this is exactly the kind of Σ^1_1-sentence that emerges from the Skolemization-procedure in section 4.1.

Now the question arises whether the composition of the Skolemization procedure and the Σ^1_1-to-IF translation procedure, applied on an IF-formula φ, results in φ itself, or else at least in an IF-formula equivalent to φ.

We easily see that even for simple (IF-)first order sentences with only one existential quantification (where in both directions the translation consists of only one step), the composition is *not* the identity. Consider

$$\exists y \forall x [x = y] :$$

Skolemization translates it into $\exists f \forall x [x = f]$ (where f is a Skolem-constant rather than a Skolem-function), which Σ^1_1-formula is translated by the procedure of this section to the IF-first order formula

$$\forall x \exists y / x [x = y].$$

In section 4.5, we will show that these two formulas can only be called equivalent in a weak sense.

Remarks and further investigation:

- We should carefully investigate the semantical consequences of all manipulations with formulas, IF-first order as well as classical second order, in the last two sections: in what sense and under which conditions is an emerging sentence equivalent to its original? What semantics for Σ^1_1 do we assume?

[27] In Hintikka's approach, the order in a block of moves for the same player is irrelevant (see also section 5.3). We could have chosen any location in-between the block of universal quantifiers and the quantifier-free part of the formula. By the choice made here, the variables y_i occur in the same order as the function variables f_i in the original Σ^1_1-formula.

- Suppose we make the procedures of the last two sections deterministic by prescribing some order in which to apply the steps. On which class of IF-first order formulas is the composition of both procedures the identity?

4.3 The law of the excluded middle

A crucial property of GTS and IF-logic, is the failure of the law of the excluded middle: it is not the case that, in all models M and for all IF-sentences φ, $\varphi \vee {\sim}\varphi$ is `true` in M.

Before we demonstrate this, note that the following is an immediate consequence of the definitions in GTS: for every sentence φ and every suitable model M

$$\begin{gathered}\varphi \text{ is } \mathtt{false} \text{ in } M \\ \Updownarrow \\ \mathsf{P}_0 \text{ has a winning strategy in } G_M(\varphi) = G_M(\varphi, \emptyset, 1) \\ \Updownarrow \\ \mathsf{P}_1 \text{ has a winning strategy in } G_M(\varphi, \emptyset, 0) \\ \Updownarrow \\ \mathsf{P}_1 \text{ has a winning strategy in } G_M({\sim}\varphi, \emptyset, 1) = G_M({\sim}\varphi) \\ \Updownarrow \\ {\sim}\varphi \text{ is } \mathtt{true} \text{ in } M.\end{gathered}$$

To analyze the law of the excluded middle in GTS, consider the game $G_M(\varphi \vee {\sim}\varphi)$ for some IF-sentence φ. The Verifier (P_1) has a winning strategy for this game if and only if he has a winning strategy for either $G_M(\varphi)$ or for $G_M({\sim}\varphi)$: he can use the first move to choose this subgame of $G_M(\varphi \vee {\sim}\varphi)$. Hence, for every IF-sentence φ and every suitable model M:

$$\begin{gathered}\varphi \vee {\sim}\varphi \text{ is } \mathtt{true} \text{ in } M \\ \Updownarrow \\ \varphi \text{ is } \mathtt{true} \text{ in } M \text{ or } {\sim}\varphi \text{ is } \mathtt{true} \text{ in } M \\ \Updownarrow \\ \varphi \text{ is either } \mathtt{true} \text{ in } M \text{ or } \mathtt{false} \text{ in } M.\end{gathered}$$

Now recall the simple IF-sentence ψ in section 3.2:

$$\forall x \exists y/x[{\sim}(x = y)].$$

We showed that this sentence is not `true` in any model M. But this does not imply that ψ is therefore `false` in every model M: the existence of a winning strategy for the Falsifier in $G_M(\psi)$ can be expressed by the (classical) formula[28]

$$\exists k \forall y[k = y],$$

[28] We will study falsity conditions in more detail in section 5.3.

which only holds in one-element models.
Hence, in models M containing more than one-element: ψ is neither **true** nor **false** in M, and as a consequence, $\psi \vee \sim\psi$ is not **true** in M.[29]

If φ is an IF-sentence that is neither **true** nor **false** in a given model M, we call φ **undecided in** M.

4.4 Contradictory and game theoretical negation

The contradictory negation of an IF-sentence φ is an IF-formula $\overline{\varphi}$ such that for every model M: $\overline{\varphi}$ is **true** in M if and only if φ is not **true** in M.

Ernst Zermelo (1913) proved that every finite depth, strictly competitive[30] two-player game of perfect information is determined: either P_0 or P_1 has a winning strategy. In particular, semantical games for classical first order formulas are determined.

From this fact, it follows that the principle of the excluded middle *does* hold in GTS on classical first order formulas φ: for every suitable model M, the game $G_M(\varphi \vee \sim\varphi)$ is determined, so the Verifier has a winning strategy for either $G_M(\varphi)$ or for $G_M(\sim\varphi)$. But the latter is the case if and only if the Falsifier has a winning strategy for $G_M(\varphi)$. So, in the case that φ is a classical first order sentence: φ is **false** in M if and only if it is not **true** in M. In other words: for classical first order formulas game theoretical and contradictory negation coincide.

This argument does not hold for the non-classical IF-first order formulas: these give rise to finite, two-person zero-sum games of *imperfect* information. So in general, in IF-logic with GTS the contradictory negation 'φ is not **true** in M' does *not* coincide with the game theoretical negation 'φ is **false** in M'.

Contradictory negation is not expressible in IF-first order logic, that is: in terms of game theoretical negation and the other logical symbols of the language.[31] One could, as Hintikka proposes ([Hin96], p. 147), extend IF-first order logic with the symbol '$\neg$' to indicate contradictory negation, and to add the following rule to the semantics: for every IF-sentence φ and suitable model M

($\neg$) $\neg\varphi$ is **true** in M exactly if φ is not **true** in M; otherwise $\neg\varphi$ is **false** in M.

[29] Similarly, $\psi \vee \sim\psi$ is not **false** in M either.

[30] I.e. for $k \in \{0, 1\}$: P_k wins iff P_{1-k} loses, and P_k either wins or loses (no tie).

[31] See [Hin96], p. 133 for Hintikka's argument, using the separation theorem.

Note that this negation is only defined in front of an entire sentence, and not as a game rule (like the game theoretical negation '$\sim$'). The resulting logic is called **Extended Independence Friendly first order logic.**

Remarks and further investigation:

- The contradictory negation expresses the non-existence of a winning strategy for the Verifier, hence can be expressed as a second order formula of the form

  $$\neg\exists f_1 \ldots \exists f_k \forall x_1 \ldots \forall x_m[\psi(f_1, \ldots, x_m)],$$

  with ψ quantifier-free. Since Σ^1_1 is not closed under negation, this is not necessarily a Σ^1_1-formula: it is a Π^1_1-formula. If the expressive power of IF-first order logic coincides with the expressive power of Σ^1_1, what would the expressive power of Extended IF-first order logic be?

4.5 Equivalence

As in classical first order logic, Hintikka defines two IF-sentences φ_1 and φ_2 to be equivalent if and only if they are `true` in the same models, i.e. for every model M (in which both φ_1 and φ_2 can be interpreted): φ_1 is `true` in M iff φ_2 is `true` in M. We denote this by: $\varphi_1 \equiv_{\mathtt{t}} \varphi_2$.
But other than in classical first order logic, as a consequence of the failure of the law of the excluded middle, being `true` in the same models does not automatically imply being `false` in the same models.[32]

For example[33], the IF-first order sentence

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

is in Hintikka's view logically equivalent to the ordinary first order sentence

$$\exists y \forall x[x = y].$$

Both formulas are `true` in one-element models only. In models with at least two elements however, the first is `undecided` while the latter is simply `false`.

We use the notation $\varphi_1 \equiv_{\mathtt{f}} \varphi_2$ to express that for every model M: φ_1 is `false` in M iff φ_2 is `false` in M. Furthermore, we define:

$$\varphi_1 \equiv \varphi_2 \overset{d}{\Leftrightarrow} \varphi_1 \equiv_{\mathtt{t}} \varphi_2 \text{ and } \varphi_1 \equiv_{\mathtt{f}} \varphi_2.$$

Now we can conclude:

$$\forall x(\exists y/x)[x = y] \equiv_{\mathtt{t}} \exists y \forall x[x = y]$$

[32] Cf. [Hin96], p. 65.

[33] See [Hin96] p. 51, and the last paragraph of section 4.2.

but:

$$\forall x(\exists y/x)[x = y] \not\equiv_{\mathfrak{f}} \exists y \forall x[x = y],$$

so:

$$\forall x(\exists y/x)[x = y] \not\equiv \exists y \forall x[x = y].$$

This shows that the notion '$\equiv$' of logical equivalence is stronger than the notion '$\equiv_{\mathfrak{t}}$' that Hintikka uses. The etymology of equivalence ('having equal values') seems to plead in favor of '$\equiv$', because this notion of equivalence takes both the truth- and the falsity-values of formulas into account.

Note that the weak notions of equivalence have the following property: for all IF-sentences φ_1, φ_2 it follows from the definitions (see also page 21) that

$$\varphi_1 \equiv_{\mathfrak{t}} \varphi_2 \Leftrightarrow {\sim}\varphi_1 \equiv_{\mathfrak{f}} {\sim}\varphi_2$$

and

$$\varphi_1 \equiv_{\mathfrak{f}} \varphi_2 \Leftrightarrow {\sim}\varphi_1 \equiv_{\mathfrak{t}} {\sim}\varphi_2.$$

Remarks and further investigation:

- As announced in the last paragraph of section 4.2, this section demonstrates that the composition of the Skolemization and Σ^1_1-IF-translation procedures of sections 4.1 and 4.2 does not preserve strong equivalence. We should be aware of this fact when relying on these procedures to prove properties of IF-logic.

4.6 Compositionality

One of the claims Hintikka makes in [Hin96], is that IF-logic does not admit of compositional semantics. Note that GTS is indeed not defined in a compositional way: it treats a sentence from outside in, whereas one would expect an inductive, 'inside-out' procedure for compositional semantics. To provide an argument for his claim, Hintikka introduces an alternative notation for IF-logic. Instead of writing

$$\forall x(\exists y/x)[\varphi(x, y)]$$

one could write

$$(\forall x//y)\exists y[\varphi(x, y)]$$

with the same interpretation: the choice of a value y is not allowed to depend on the (chosen) value for x. In the new notation the independency is announced in advance, and the formation rules become context independent.[34]

In compositional semantics for IF logic, we would have to give one meaning to the subformula $\exists y[\varphi(x,y)]$ independent of the context in which it appears. But, in the new notation, we want to give different interpretations to this subformula as part of $\forall x \exists y[\varphi(x,y)]$ and as part of $(\forall x//y)\exists y[\varphi(x,y)]$ (or $\exists x \exists y[\varphi(x,y)]$: in the case of more existential quantifiers information independence is assumed by convention! See section 5.3).

This is not a strict impossibility proof, as Hintikka points out himself ([Hin96], p. 112). Actually, Wilfrid Hodges has developed compositional semantics for IF-logic, under the name of 'trump semantics'. The current status of this discussion can be found in [Hod01] and [SH01].

5 Negation normal form, implication and Skolemization

5.1 De Morgan's laws in IF-logic

Hintikka works with IF-formulas in negation normal form, and claims that arbitrary IF-formulas can be rewritten in negation normal form using the traditional rules, i.e. De Morgan's laws.[35]
In section 2.1 we saw how these laws hold in GTS for classical first order formulas: if a classical first order formula φ is transformed into a formula φ' by application of De Morgan's laws, $G_M(\varphi)$ and $G_M(\varphi')$ are essentially the same.[36] A winning strategy in $G_M(\varphi)$ is at the same time a winning strategy in $G_M(\varphi')$, and vice versa, for both players. From this, it follows that if φ is a classical first order formula, φ and its negation normal form are equivalent in GTS in the strong sense ('$\equiv$') of section 4.5.

[34] Hintikka claims that the double slash notation has some advantages over the single slash notation that has become the standard. It would, for instance, allow us to define a distribution law like:

$$(\forall x//\vee)[P(x) \vee Q(x)] \Rightarrow \forall x[P(x)] \vee \forall x[Q(x)].$$

For Hintikka these two formulas are equivalent, which justifies this law. In the strong sense of section 4.5, they are not: in $M := \langle\{0,1\}, P \mapsto \text{"even"}, Q \mapsto \text{"odd"}\rangle$, the first one is `undecided`, while the latter is `false` (see also the table on page 31).

A disadvantage of the double slash notation is that it causes ambiguities when there are several $\vee$-connectives falling under the scope of $\forall x$.

[35] See section 3.2.

[36] The same moves are made by the same players, in the same order. As remarked in section 2.1, it would be useful to formalize the notion of being 'the same game'.

Does the same hold for arbitrary IF-formulas? Our answer is 'yes', but only provided we extend the slash notation given in section 3.2 in such way that also $\forall$ and $\wedge$ may be 'slashed'.
To illustrate this, consider the following formulas:

$$\begin{array}{rcl} \varphi & : & \forall x(\exists y/x)[x=y]; \\ \sim\varphi & : & \sim\forall x(\exists y/x)[x=y]; \\ \psi_1 & : & \exists x\forall y[\sim(x=y)]; \\ \psi_2 & : & \exists x(\forall y/x)[\sim(x=y)]. \end{array}$$

Following Hintikka in his claim that the traditional rules apply to IF-formulas, and given his definition of IF-formulas (only applying the slash to $\exists$ and $\vee$), ψ_1 is the only candidate for the IF-negation normal form (NNF) of the IF-formula $\sim\varphi$. Indeed, $\sim\varphi$ and ψ_1 are equivalent, but only in the weak sense '$\equiv_t$'.[37]
Now, how do we rewrite $\sim\sim\varphi$ in NNF? One way is, of course, to let the two initial negations cancel out[38]: φ is in NNF. On the other hand, we should be able to push "the negation signs as deep into the formulas as they can go": see the quote from [Hin98] on page 5 of this paper. Pushing the innermost negation sign 'into' φ, we get $\sim\psi_1$ out of $\sim(\sim\varphi)$ by the previous paragraph. Pushing the remaining negation sign into ψ_1, we get the formula $\varphi' := \forall x\exists y[x=y]$ as NNF of $\sim\sim\varphi$.
But now we have both φ and φ' as NNF's for $\sim\sim\varphi$, and we can easily see: $\varphi \not\equiv_t \varphi'$. What we observe here, is the failure of the **substitution property** for weakly equivalent subformulas: in $\sim\sim\varphi$, we replaced the subformula $\sim\varphi$ by the $\equiv_t$-equivalent formula ψ_1, but the resulting formula is no longer $\equiv_t$-equivalent to $\sim\sim\varphi$.[39] In fact, $\sim\psi_1$ is $\equiv_f$-equivalent to $\sim\sim\varphi$. We can summarize the situation in the following diagram:

$$\begin{array}{ccccc} \varphi & \equiv & \sim\sim\varphi = \sim(\sim\varphi) & \equiv_f & \sim\psi_1 \\ = & & & & \equiv \\ \forall x(\exists y/x)[x=y] & & \not\equiv_t & & \forall x\exists y[x=y]. \end{array}$$

Making IF-logic 'symmetric', in the sense that both $\exists$- and $\forall$-moves, and both $\vee$- and $\wedge$-moves, can be informationally independent of moves of the opponent, enables us to work with strong equivalence. The formula ψ_2 is a

[37] To see this, note that for every suitable model M, P_1 does not have a winning strategy in either $G_M(\sim\varphi)$ or $G_M(\psi_1)$; on the other hand, P_0 has a winning strategy in $G_M(\psi_1)$ and not in $G_M(\phi)$.

[38] See section 2.1.

[39] We use the remark made at the end of section 4.5. By the fact that $\sim\varphi \not\equiv_f \psi_1$, we also know that $\sim\sim\varphi \not\equiv_t \sim\psi_1$.

strongly equivalent negation normal form of $\sim\varphi$. We leave it to the reader to verify the equivalences in the following diagram:

$$\begin{array}{ccccc} \varphi & \equiv & \sim\sim\varphi = \sim(\sim\varphi) & \equiv & \sim\psi_2 \\ = & & & & \equiv \\ \forall x(\exists y/x)[x = y] & & = & & \forall x(\exists y/x)[x = y]. \end{array}$$

In the symmetric definition of IF-logic, we can formulate De Morgan-like laws, exactly reflecting the changes of roles of the players, including their informational restrictions:

$$\begin{array}{rcl} \sim(\forall x/\overline{y})\varphi & \equiv & (\exists x/\overline{y}) \sim\varphi \\ \sim(\exists x/\overline{y})\varphi & \equiv & (\forall x/\overline{y}) \sim\varphi \\ \sim(\psi_1(\wedge/\overline{y})\psi_2) & \equiv & \sim\psi_1(\vee/\overline{y}) \sim\psi_2 \\ \sim(\psi_1(\vee/\overline{y})\psi_2) & \equiv & \sim\psi_1(\wedge/\overline{y}) \sim\psi_2 \end{array}$$

where $\overline{y}$ denotes a -possibly empty- sequence of variables associated with moves of the opponent.

5.2 Implication

In GTS, no rule is provided for implication ('$\rightarrow$'). We are tempted to define it in the usual manner, i.e. for all IF-formulas φ and ψ

$$\varphi \rightarrow \psi \overset{d}{=} \sim\varphi \vee \psi. \tag{4}$$

What does this mean in Game Theoretical Semantics? Given a model M, the first move in the game $G_M(\varphi \rightarrow \psi)$ is the choice by the Verifier to continue by one of the games $G_M(\sim\varphi)$ or $G_M(\psi)$. The strategy-function prescribing this choice, combined with strategies for $G_M(\sim\varphi)$ and $G_M(\psi)$, make a strategy for the entire game $G_M(\varphi \rightarrow \psi)$.

A winning strategy for the Verifier in either $G_M(\sim\varphi)$ or $G_M(\psi)$, combined with the appropriate choice in the first move, is a winning strategy for the Verifier in the game $G_M(\varphi \rightarrow \psi)$. Hence, $\varphi \rightarrow \psi$ is `true` in M if either $\sim\varphi$ or ψ is `true` in M.[40] If the Verifier has no winning strategy for either $G_M(\sim\varphi)$ or $G_M(\psi)$, the optimal choice is to choose, if possible, a game for which the Falsifier has no winning strategy either: $\varphi \rightarrow \psi$ is `undecided` in M if neither $\sim\varphi$ nor ψ is `true` in M, and at least one of both is not `false` in M either.

Following this line of thought, we draw a truth table, using '1' for `true`, '0' for `false`, and '?' for `undecided`:[41]

[40] Or, in other words: $\varphi \rightarrow \psi$ is `true` in M if either φ is `false` in M or ψ is `true` in M.

[41] This truth value of an IF-sentence indicates the index of the player with a winning strategy, or a question mark in the case neither player has one.

φ	ψ	$\sim\varphi$	$\varphi \rightarrow \psi$
1	1	0	1
1	?	0	?
1	0	0	0
?	1	?	1
?	?	?	?
?	0	?	?
0	1	1	1
0	?	1	1
0	0	1	1

Under this definition for implication, the IF-first order subformula

$$\exists y \forall x[x = y] \rightarrow \forall x(\exists y/x)[x = y]$$

is **logically** true in the following sense: the Verifier has a winning strategy in every model.[42] On the other hand, the converse implication

$$\forall x(\exists y/x)[x = y] \rightarrow \exists y \forall x[x = y]$$

is *not* logically true: it is true in one-element models, but undecided in all other models. This seems to support our conclusion of section 4.5: the formulas $\exists y \forall x[x = y]$ and $\forall x(\exists y/x)[x = y]$ should not be called equivalent.

A serious objection to this definition of implication however, is the fact that '$\varphi \rightarrow \varphi$' is not logically true for all IF-formulas φ: $\sim\varphi \vee \varphi$ is undecided in every model in which φ is undecided. Hence, under the definition of '$\rightarrow$' as given in formula (4), the following property does *not* hold: for all IF-formulas φ, ψ and every suitable model M, $\varphi \rightarrow \psi$ is true in M if 'φ is true in M' implies that 'ψ is true in M'.[43]

Aiming for a definition of implication that does have this property, we look for an IF-formula with the following truth condition: 'the existence of a winning strategy for the Verifier in $G_M(\varphi)$ implies the existence of a winning strategy for the Verifier in $G_M(\psi)$'. This can be written out as a formula $\Phi \rightarrow \Psi$, with $\Phi, \Psi \in \Sigma_1^1$, which is generally not a Σ_1^1-formula. Hence, in general, we cannot hope for a translation of this implication into an IF-first order formula.

We can only define such notion of implication as a 'truth functional conditional', i.e. on the level of the strategies and not on the level of the game rules (see [Hin96], p. 138).

[42] To see this, note that in one-element models ϕ is false, and in all other models ψ is true.

[43] Note that the converse of this property, Modus Ponens, does hold: if $\varphi \rightarrow \psi$ and φ are both true in a given model M, then ψ is true in M as well.

5.3 Skolemization as concept of strategy

Hintikka often appeals to our intuitions about games and strategies. That these are not always precise enough to avoid ambiguities, can be concluded from the following example, due to Wilfrid Hodges ([Hod97], p. 548).
As we have seen in section 4.5, the IF-formula

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

is `undecided` in all models containing more than one element. Now consider:

$$\forall x\exists z(\exists y/x)[x = y].$$

According to the definitions we have given so far, the Verifier could use the following strategy in a semantical game for this formula: choose z equal to x, and then y equal to z.[44] This strategy is winning in every model, so the addition of the empty quantification $\exists z$ would change the truth value of the first formula from `undecided` to `true` in all models containing more than one element.
One might object by arguing that, if y depends on z, and z depends on x, then y (indirectly) depends on x, and that this shouldn't be allowed by the the 'slashed-out' x under $\exists y$. This objection is reasonable, but in the bare definitions of IF-logic, there is nothing to prevent it.
According to Hintikka, some extra specification is needed to prevent that "otherwise "forbidden" dependencies of existential quantifiers could be created through the mediation of intervening existential quantifiers" ([Hin96], p. 63).
Hintikka's suggested specification is quite counterintuitive: he introduces the provision that "moves connected with existential quantifiers are always independent of earlier moves with existential quantifiers". But in that case: how can a classical first order sentence like $\exists x\exists y[x = y]$ still be `true`?

The answer to this last question lies in the interpretation of information (in)dependency and strategies strictly in terms of Skolem functions and their arguments. This is present in Hintikka's more or less formal argumentation, but also seems to determine what he considers intuitively clear and what not. Consistently applying this interpretation, the (confusing) extra specification is not needed.

We illustrate this by working through some very simple examples, following what we consider to be Hintikka's line of thought. As proposed in section 5.1, we also allow the slash notation for $\forall$ and $\wedge$ to bring the game theoretical

[44] This phenomenon of using other parts of the formula to reconstruct information in a situation of imperfect information, has been called 'signaling'.

negations of the formulas into negation normal form.

Recall that the truth condition of a sentence φ (section 2.3) emerges as a Σ_1^1-formula from the process of Skolemization, as described in section 4.1. It will be of the following appearance:

$$\exists f_1 \ldots \exists f_k \exists c_1 \ldots \exists c_n \forall x_1 \ldots \forall x_m [\varphi'(f_1, \ldots, x_m)],$$

where the existential quantifications correspond to all the (possible) moves of the Verifier (0-ary Skolem-'functions' are from now on represented as Skolem-*constants* c_i), and the universal quantifications to all the (possible) moves of his opponent. The part between the square brackets, φ', is a quantifier-free (classical) first order formula.
This truth condition expresses the existence of a winning strategy for the Verifier in the following sense: for each move that can occur during the game, a choice based on the available (quantifier-)moves of the opponent is prescribed by the choice-functions and constants. These functions and constants are to be chosen before the start of the game, i.e. their choice cannot depend on moves of the opponent. A strategy cannot be adapted during the game.
Although Hintikka's specification seems to suggest that a player should make his moves independently of his own previous moves, a player is allowed to choose the functions or constants equal to, or correlated with (in other words: dependent of), other functions and constants in his strategy.

We can determine the falsity condition exactly in the same way as the truth condition, because the falsity condition of a sentence φ is the truth condition of its game theoretical negation $\sim\varphi$ (see page 21). Hence, a procedure to find the falsity condition of φ is to subsequently rewrite $\sim\varphi$ into negation normal form (using the rules from section 5.1) and apply the Skolemization procedure of section 4.1.

We have worked these procedures out for some very simple IF-first order formulas. First, some formulas containing no connectives:

	φ	$\sim\varphi$	truth condition for φ	falsity condition for φ
1	$\forall x \exists y[R(x,y)]$	$\exists x \forall y[\sim R(x,y)]$	$\exists f \forall x[R(x,f(x))]$	$\exists k \forall y[\neg R(k,y)]$
2	$\forall x \exists y/x[R(x,y)]$	$\exists x \forall y/x[\sim R(x,y)]$	$\exists c \forall x[R(x,c)]$	$\exists k \forall y[\neg R(k,y)]$
3	$\exists y \forall x[R(x,y)]$	$\forall y \exists x[\sim R(x,y)]$	$\exists c \forall x[R(x,c)]$	$\exists g \forall y[\neg R(g(y),y)]$
4	$\exists y \forall x/y[R(x,y)]$	$\forall y \exists x/y[\sim R(x,y)]$	$\exists c \forall x[R(x,c)]$	$\exists k \forall y[\neg R(k,y)]$
5	$\exists x \exists y[R(x,y)]$	$\forall x \forall y[\sim R(x,y)]$	$\exists c_1 \exists c_2[R(c_1,c_2)]$	$\forall x \forall y[\neg R(x,y)]$
6	$\forall x \forall y[R(x,y)]$	$\exists x \exists y[\sim R(x,y)]$	$\forall x \forall y[R(x,y)]$	$\exists k_1 \exists k_2[\neg R(k_1,k_2)]$
7	$\forall x \exists z \exists y/x[R(x,y)]$	$\exists x \forall z \forall y/x[\sim R(x,y)]$	$\exists f \exists c \forall x[R(x,c)]$	$\exists k \forall z \forall y[\neg R(k,y)]$

The table shows that formulas φ on lines (2), (3) and (4) share the same truth condition (and are hence equivalent in Hintikka's sense: '$\equiv_t$'). The formulas on lines (1), (2) and (4) share the same falsity condition (and are

hence 'falsity-equivalent': '$\equiv_{\mathfrak{f}}$'). Hence, formulas (2) and (4) are equivalent in the strong sense '$\equiv$'.[45]
Hodges's example from the beginning of this section arises from the formula φ on line (7) when interpreting the symbol R by standard equality. The empty quantifications $\exists f$ in the truth condition, and $\forall z$ in the falsity condition are irrelevant in these (classically interpreted) Σ^1_1-formulas, and can be left out. We then see that this formula is equivalent ('$\equiv$') to the formulas φ on lines (2) and (4).

In answer to the question we asked on page 29: the formula $\exists x \exists y[x = y]$ (a special case of line 5) is **true** in every model, because the Verifier is allowed to choose the same value for both constants c_1 and c_2.

In the following table the same has been done for some simple IF-formulas containing the connectives $\vee$ and $\wedge$:

φ	$\sim\varphi$	truth condition for φ	falsity condition for φ
$A \vee B$	$\sim A \wedge \sim B$	$\exists c[(c = 0 \rightarrow A)$ $\wedge(c \neq 0 \rightarrow B)]$	$\neg A \wedge \neg B$
$\forall x[P(x) \vee Q(x)]$	$\exists x[\sim P(x) \wedge \sim Q(x)]$	$\exists f \forall x[(f(x) = 0 \rightarrow P(x))$ $\wedge(f(x) \neq 0 \rightarrow Q(x))]$	$\exists k[\neg P(k) \wedge \neg Q(k)]$
$\forall x[P(x)(\vee/x)Q(x)]$	$\exists x[\sim P(x)(\wedge/x) \sim Q(x)]$	$\exists c \forall x[(c = 0 \rightarrow P(x))$ $\wedge(c \neq 0 \rightarrow Q(x))]$	$\exists k[\neg P(k) \wedge \neg Q(k)]$
$\forall x[P(x)] \vee \forall x[Q(x)]$	$\exists x[\sim P(x)] \wedge \exists x[\sim Q(x)]$	$\exists c \forall x_1 \forall x_2[(c = 0 \rightarrow P(x_1))$ $\wedge(c \neq 0 \rightarrow Q(x_2))]$	$\exists g \forall y[(y = 0 \rightarrow \neg P(g(y)))$ $\wedge(y \neq 0 \rightarrow \neg Q(g(y)))]$
$\exists x[P(x) \vee Q(x)]$	$\forall x[\sim P(x) \wedge \sim Q(x)]$	$\exists c_1 \exists c_2[c_2 = 0 \rightarrow P(c_1))$ $\wedge(c_2 \neq 0 \rightarrow Q(c_1))]$	$\forall x[\neg P(x) \wedge \neg Q(x)]$
$A \wedge B$	$\sim A \vee \sim B$	$A \wedge B$	$\exists k[(k = 0 \rightarrow \neg A) \wedge$ $(k \neq 0 \rightarrow \neg B)]$
$\exists x[P(x) \wedge Q(x)]$	$\forall x[\sim P(x) \vee \sim Q(x)]$	$\exists c[P(c) \wedge Q(c)]$	$\exists g \forall x[(g(x) = 0 \rightarrow \neg P(x))$ $\wedge(g(x) \neq 0 \rightarrow \neg Q(x))]$
$\exists x[P(x)(\wedge/x)Q(x)]$	$\forall x[\sim P(x) \vee \sim Q(x)]$	$\exists c[P(c) \wedge Q(c)]$	$\exists k \forall x[(k = 0 \rightarrow \neg P(x))$ $\wedge(k \neq 0 \rightarrow \neg Q(x))]$
$\exists x[P(x)] \wedge \exists x[Q(x)]$	$\forall x[\sim P(x)] \vee \forall x[\sim Q(x)]$	$\exists f \forall y[(y = 0 \rightarrow P(f(y)))$ $\wedge(y \neq 0 \rightarrow Q(f(y))]$	$\exists k \forall x_1 \forall x_2[(k = 0 \rightarrow [\neg P(x_1)])$ $\wedge(k \neq 0 \rightarrow [\neg Q(x_2)])]$
$\forall x[P(x) \wedge Q(x)]$	$\exists x[\sim P(x) \vee \sim Q(x)]$	$\forall x[P(x) \wedge Q(x)]$	$\exists k_1 \exists k_2[k_2 = 0 \rightarrow \neg P(k_1))$ $\wedge(k_2 \neq 0 \rightarrow \neg Q(k_1))]$

The examples and (sketched) proofs in [Hin96], demonstrate that the Skolemization procedure as applied in the tables above, and not intuition, defines how Hintikka interprets the notions of strategy and truth for IF-formulas. It seems therefore necessary to investigate this procedure closely, and to formally prove that the emerging Σ^1_1-formulas have those properties that we would expect from truth- and falsity-conditions.

6. Concluding remarks and further investigation

Hintikka's project seems to be inspired by his earlier work in natural language semantics (see, for example [HS97]). It is not directly clear how a successful approach in this area can be a guarantee for success in the foundations of mathematics. I agree that logic can be used for different purposes,

[45] Cf. section 4.5.

and both the descriptive and the deductive function have their merits. But I don't think the one should be preferred over the other in an absolute sense: it will depend on the nature of the field of application which function of logic serves best. In my opinion, the Foundations of Mathematics are best served by a logic with great *deductive* power.[46] This seems problematic in IF-logic, because it lacks a notion of implication being able to express logical consequence (section 5.2).

Despite its differences with classical first order logic, Hintikka claims some classical properties for IF-logic (a.o. compactness, the separation theorem, and the downward Löwenheim-Skolem theorem). The arguments given in support of these claims ([Hin96], pp. 59-61) use the translation procedures from IF-first order logic to Σ_1^1 and back, as described here in sections 4.1 and 4.2. The same goes for his claim that a truth predicate for IF-logic can be defined at first order level.
It is, however, left to our intuitions to trust that these translation procedures are 'sound'. In what sense do the original formula and its translation correspond? Does it matter which semantics is used for Σ_1^1? Does the fact that the composition of the procedures in sections 4.1 and 4.2 preserves only weak equivalence,[47] have consequences for the classical properties mentioned above?
Once we have ascertained that we can trust the arguments based on the translation procedures of IF-logic to Σ_1^1 and back, haven't we then concluded that IF-first order logic is nothing more or less than Σ_1^1 (and hence not the revolutionary new logic it was promised to be)?

We followed Hintikka in defining the semantics prior to the syntax of IF-logic. Only by the examples given in terms of Skolem-functions (like in section 3.1) we get to understand how the slash-notation *and* the notion of strategy should be interpreted for IF-logic.
The way in which Hintikka's proposal is set up, highly resembles the theory of Henkin-quantifiers, especially in its focus on skolemization. Skolemization requires a negation normal form, which, as we have seen in section 5.1, can only be obtained in general provided we extend the slash notation to $\forall$ and $\wedge$.
If IF-logic really is to serve as "a new and better basic logic" ([Hin96], p.ix), both syntax and semantics need to be defined in a more rigorous way. The notions of scope and of strategy should get formal descriptions. How do we think of the choice of a strategy by the players? Are all strategies available to them? And if we conceive them as functions, do they have to be in some sense constructive to be realistically usable as a decision method?

[46] Hintikka will consider this a dogma "ripe for rejection", [Hin96], p. viii.
[47] Cf. the end of section 4.2 and section 4.5.

Theo Janssen's critical approach, [Jan01], distinguishing information independence from imperfect information, shows that one should be careful about the interpretation of the aspect of information as well.

Game theory nicely incorporates the notion of 'information' into logic, and the idea of breaking open the traditional restrictions of quantifier-scopes is simple but eye-opening.
On the other hand, rereading the 'mission statement' at the start of [Hin96] (see the quotation at page 1 of this paper), one is inclined to say that the conceptual problems that arise in IF-logic still outweigh the conceptual problems of 'earlier views'.

References

[Ham88] Alan G. Hamilton. *Logic for Mathematicians.* Cambridge University Press, revised edition, 1988.

[Hen61] L. Henkin. Some remarks on infinitely long formulas. In *Infinitistic methods*, pages 167–183. Pergamon Press, 1961.

[Hin96] Jaakko Hintikka. *The Principles of Mathematics Revisited.* Cambridge University Press, 1996.

[Hin98] Jaakko Hintikka. Truth definitions, skolem functions and axiomatic set theory. *The bulletin of Symbolic Logic*, 4(3):303–337, September 1998.

[Hod97] W. Hodges. Compositional semantics for a language of imperfect information. *Bulletin of the IGPL*, 5 (4):539–563, 1997.

[Hod01] Wilfrid Hodges. Formal features of compositionality. *Journal of Logic, Language and Information*, 10:7–28, 2001.

[HS97] Jaakko Hintikka and Esa Saarinen. *Game Theoretical Semantics.* Reidel, 1997.

[Jan01] Theo M. V. Janssen. On the definition of independence in logic. In Gabriel Sandu and Marc Pauly, editors, *Workshop on Logic and Games.* European Summer School on Logic, Language and Information, August 2001.

[SH01] Gabriel Sandu and Jaakko Hintikka. Aspects of compositionality. *Journal of Logic, Language and Information*, 10:49–61, 2001.

[Ten98] Neil Tennant. Games some people would have all of us play. *Philosophia Mathematica*, 6:90–115, 1998.

Computer Science Reports **Department of Mathematics and Computer Science**
Technische Universiteit Eindhoven

If you want to receive reports, send an email to: m.m.j.l.philips@tue.nl (we cannot guarantee the availability of the requested reports)

In this series appeared:

97/02	J. Hooman and O. v. Roosmalen	A Programming-Language Extension for Distributed Real-Time Systems, p. 50.
97/03	J. Blanco and A. v. Deursen	Basic Conditional Process Algebra, p. 20.
97/04	J.C.M. Baeten and J.A. Bergstra	Discrete Time Process Algebra: Absolute Time, Relative Time and Parametric Time, p. 26.
97/05	J.C.M. Baeten and J.J. Vereijken	Discrete-Time Process Algebra with Empty Process, p. 51.
97/06	M. Franssen	Tools for the Construction of Correct Programs: an Overview, p. 33.
97/07	J.C.M. Baeten and J.A. Bergstra	Bounded Stacks, Bags and Queues, p. 15.
97/08	P. Hoogendijk and R.C. Backhouse	When do datatypes commute? p. 35.
97/09	Proceedings of the Second International Workshop on Communication Modeling, Veldhoven, The Netherlands, 9-10 June, 1997.	Communication Modeling- The Language/Action Perspective, p. 147.
97/10	P.C.N. v. Gorp, E.J. Luit, D.K. Hammer E.H.L. Aarts	Distributed real-time systems: a survey of applications and a general design model, p. 31.
97/11	A. Engels, S. Mauw and M.A. Reniers	A Hierarchy of Communication Models for Message Sequence Charts, p. 30.
97/12	D. Hauschildt, E. Verbeek and W. van der Aalst	WOFLAN: A Petri-net-based Workflow Analyzer, p. 30.
97/13	W.M.P. van der Aalst	Exploring the Process Dimension of Workflow Management, p. 56.
97/14	J.F. Groote, F. Monin and J. Springintveld	A computer checked algebraic verification of a distributed summation algorithm, p. 28
97/15	M. Franssen	λP-: A Pure Type System for First Order Loginc with Automated Theorem Proving, p.35.
97/16	W.M.P. van der Aalst	On the verification of Inter-organizational workflows, p. 23
97/17	M. Vaccari and R.C. Backhouse	Calculating a Round-Robin Scheduler, p. 23.
97/18	Werkgemeenschap Informatiewetenschap redactie: P.M.E. De Bra	Informatiewetenschap 1997 Wetenschappelijke bijdragen aan de Vijfde Interdisciplinaire Conferentie Informatiewetenschap, p. 60.
98/01	W. Van der Aalst	Formalization and Verification of Event-driven Process Chains, p. 26.
98/02	M. Voorhoeve	State / Event Net Equivalence, p. 25
98/03	J.C.M. Baeten and J.A. Bergstra	Deadlock Behaviour in Split and ST Bisimulation Semantics, p. 15.
98/04	R.C. Backhouse	Pair Algebras and Galois Connections, p. 14
98/05	D. Dams	Flat Fragments of CTL and CTL*: Separating the Expressive and Distinguishing Powers. P. 22.
98/06	G. v.d. Bergen, A. Kaldewaij V.J. Dielissen	Maintenance of the Union of Intervals on a Line Revisited, p. 10.
98/07	Proceedings of the workhop on Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM'98) June 22, 1998 Lisbon, Portugal	edited by W. v.d. Aalst, p. 209
98/08	Informal proceedings of the Workshop on User Interfaces for Theorem Provers. Eindhoven University of Technology ,13-15 July 1998	edited by R.C. Backhouse, p. 180

98/09	K.M. van Hee and H.A. Reijers	An analytical method for assessing business processes, p. 29.
98/10	T. Basten and J. Hooman	Process Algebra in PVS
98/11	J. Zwanenburg	The Proof-assistemt Yarrow, p. 15
98/12	Ninth ACM Conference on Hypertext and Hypermedia Hypertext '98 Pittsburgh, USA, June 20-24, 1998 Proceedings of the second workshop on Adaptive Hypertext and Hypermedia.	Edited by P. Brusilovsky and P. De Bra, p. 95.
98/13	J.F. Groote, F. Monin and J. v.d. Pol	Checking verifications of protocols and distributed systems by computer. Extended version of a tutorial at CONCUR'98, p. 27.
99/01	V. Bos and J.J.T. Kleijn	Structured Operational Semantics of χ , p. 27
99/02	H.M.W. Verbeek, T. Basten and W.M.P. van der Aalst	Diagnosing Workflow Processes using Woflan, p. 44
99/03	R.C. Backhouse and P. Hoogendijk	Final Dialgebras: From Categories to Allegories, p. 26
99/04	S. Andova	Process Algebra with Interleaving Probabilistic Parallel Composition, p. 81
99/05	M. Franssen, R.C. Veltkamp and W. Wesselink	Efficient Evaluation of Triangular B-splines, p. 13
99/06	T. Basten and W. v.d. Aalst	Inheritance of Workflows: An Approach to tackling problems related to change, p. 66
99/07	P. Brusilovsky and P. De Bra	Second Workshop on Adaptive Systems and User Modeling on the World Wide Web, p. 119.
99/08	D. Bosnacki, S. Mauw, and T. Willemse	Proceedings of the first international syposium on Visual Formal Methods - VFM'99
99/09	J. v.d. Pol, J. Hooman and E. de Jong	Requirements Specification and Analysis of Command and Control Systems
99/10	T.A.C. Willemse	The Analysis of a Conveyor Belt System, a case study in Hybrid Systems and timed μCRL, p. 44.
99/11	J.C.M. Baeten and C.A. Middelburg	Process Algebra with Timing: Real Time and Discrete Time, p. 50.
99/12	S. Andova	Process Algebra with Probabilistic Choice, p. 38.
99/13	K.M. van Hee, R.A. van der Toorn, J. van der Woude and P.A.C. Verkoulen	A Framework for Component Based Software Architectures, p. 19
99/14	A. Engels and S. Mauw	Why men (and octopuses) cannot juggle a four ball cascade, p. 10
99/15	J.F. Groote, W.H. Hesselink, S. Mauw, R. Vermeulen	An algorithm for the asynchronous *Write-All* problem based on process collision*, p. 11.
99/16	G.J. Houben, P. Lemmens	A Software Architecture for Generating Hypermedia Applications for Ad-Hoc Database Output, p. 13.
99/17	T. Basten, W.M.P. v.d. Aalst	Inheritance of Behavior, p.83
99/18	J.C.M. Baeten and T. Basten	Partial-Order Process Algebra (and its Relation to Petri Nets), p. 79
99/19	J.C.M. Baeten and C.A. Middelburg	Real Time Process Algebra with Time-dependent Conditions, p.33.
99/20	Proceedings Conferentie Informatiewetenschap 1999 Centrum voor Wiskunde en Informatica 12 november 1999, p.98	edited by P. de Bra and L. Hardman
00/01	J.C.M. Baeten and J.A. Bergstra	Mode Transfer in process Algebra, p. 14
00/02	J.C.M. Baeten	Process Algebra with Explicit Termination, p. 17.
00/03	S. Mauw and M.A. Reniers	A process algebra for interworkings, p. 63.
00/04	R. Bloo, J. Hooman and E. de Jong	Semantical Aspects of an Architecture for Distributed Embedded Systems*, p. 47.
00/05	J.F. Groote and M.A. Reniers	Algebraic Process Verification, p. 65.

00/06	J.F. Groote and J. v. Wamel	The Parallel Composition of Uniform Processes wit Data, p. 19
00/07	C.A. Middelburg	Variable Binding Operators in Transition System Specifications, p. 27.
00/08	I.D. van den Ende	Grammars Compared: A study on determining a suitable grammar for parsing and generating natural language sentences in order to facilitate the translation of natural language and MSC use cases, p. 33.
00/09	R.R. Hoogerwoord	A Formal Development of Distributed Summation, p. 35
00/10	T. Willemse, J. Tretmans and A. Klomp	A Case Study in Formal Methods: Specification and Validation on the OM/RR Protocol, p. 14.
00/11	T. Basten and D. Bošnački	Enhancing Partial-Order Reduction via Process Clustering, p. 14
00/12	S. Mauw, M.A. Reniers and T.A.C. Willemse	Message Sequence Charts in the Software Engineering Process, p. 26
00/13	J.C.M. Baeten, M.A. Reniers	Termination in Timed Process Algebra, p. 36
00/14	M. Voorhoeve, S. Mauw	Impossible Futures and Determinism, p. 14
00/15	M. Oostdijk	An Interactive Viewer for Mathematical Content based on Type Theory, p. 24.
00/16	F. Kamareddine, R. Bloo, R. Nederpelt	Characterizing λ-terms with equal reduction behavior, p. 12
00/17	T. Borghuis, R. Nederpelt	Belief Revision with Explicit Justifications: an Exploration in Type Theory, p. 30.
00/18	T. Laan, R. Bloo, F. Kamareddine, R. Nederpelt	Parameters in Pure Type Systems, p. 41.
00/19	J. Baeten, H. van Beek, S. Mauw	Specifying Internet applications with *DiCons*, p. 9
00/20	Editors: P. v.d. Vet and P. de Bra	Proceedings: Conferentie Informatiewetenschap 2000, De Doelen, Utrecht, 5 april 2000, p. 98
01/01	H. Zantema and J. v.d. Pol	A Rewriting Approach to Binary Decision Diagrams, p. 27
01/02	T.A.C. Willemse	Interpretations of Automata, p. 41
01/03	G.I. Joigov	Systems for Open Terms: An Overview, p. 39
01/04	P.J.L. Cuijpers, M.A. Reniers, A.G. Engels	Beyond Zeno-behaviour, p. 15
01/05	D.K. Hammer, J. Hooman, M.A. Reniers, O. van Roosmalen, A. Sintotski	Design of the mine pump control system, p. 69
01/06	J.C.M. Baeten and E.P. de Vink	Axiomatizing GSOS with termination, p. 22
01/10	7e Nederlandse Testdag editors L.M.G. Feijs, N. Goga, S. Mauw, T.A.C. Willemse	
01/11	Editors: P. de Bra, P. Brusilovsky and A. Kobsa	Proceedings: Third Workshop on Adaptive Hypertext and Hypermedia Sonthofen, Germany, July 14, 2001 Århus, Denmark, August 15, 2001
01/12	J.C.M. Baeten, H.M.A. van Beek and S. Mauw	Operational semantics of DiCons, a formal language for developing Internet applications.
01/13	C.A. Middelburg	Operational Conservativity with Binding Terms.
01/14	Editors: J.C.M. Baeten, D. Bosnacki, J.P.M. Voeten and L. Holenderski	Proceedings: Third Dutch Model Checking Day Eindhoven, November 7, 2001
01/15	Pim Lemmens and Geert-Jan Houben	XML to XML through XML
01/16	Francien Dechesne	Investigating the basic notions of Hintikka's Independence Friendly Logic

TU/e technische universiteit eindhoven

P.O. Box 513
5600 MB Eindhoven
The Netherlands

/ department of mathematics and computing science