GAME THEORETICAL SEMANTICS AS THE BASIS OF A GENERAL LOGIC

(DRAFT)

March 15 2010

Jaakko Hintikka and John Symons

1. Quantifiers as dependence indicators

The usual basic logic, variously known as first-order logic, predicate calculus or quantification theory is essentially the logic of the two quantifiers, viz. the existential quantifier $(\exists x)$ and the universal quantifier $(\forall y)$, plus truth functional propositional connectives. It is a much richer logic than this simple description might first suggest. For instance, much, though not all, of actual mathematical reasoning can be construed as such first-order reasoning. Moreover, it is the logic used in set theory which is often thought of as the lingua franca of all mathematics. The source of this richness and subtlety of the semantics of this logic is not immediately obvious. Often it is thought that the semantics of quantifiers is exhausted by their ranging over a class of values sometimes called the universe of discourse. Frege expressed this idea by construing quantifiers as higher order predicates expressing the nonemptiness or the exceptionlessness of lower-order predicates. However, this is a restrictive misconception, since quantifiers have another semantical function. By their formal dependence on each other quantifiers can express the actual material dependence of their variables on each other. This is in fact the only direct method of expressing dependencies between variables on the first-order level.

This component of the semantics of quantifiers is the source of the expressive strength of first-order logic. Indeed, for practically all theoretical purposes, the pattern of dependence which obtains between different quantifiers (and propositional connectives) *is* the logical structure of their formulas, as will be spelled out later in this work

In conventional formulations of first-order logic, the formal dependencies of quantifiers are expressed by the nesting of their scopes. The usual way of expressing this formal (syntactical) scope of a quantifier is by a pair of parentheses (brackets) following it and encompassing its scope. As will be seen, this notion of scope is not without its problems. One problem is the usual notation. This notation is incapable of codifying the totality of dependence structures that should be expressed by means of first-order logic. The reason is that scope structure in the received logic of quantification is that of a labeled tree or a finite number of such trees. As such, we can express only patterns of dependence relations that have a tree structure. There is no reason to expect that all dependence relations behave in this way. This means that the conventional first-order logic is not the general logic of quantifiers.

This deficiency is partly corrected in what is known as independence friendly (IF) first-order logic.[1] There, the syntactic structure of formulas of conventional logic is left intact except for allowing an

[1] For an introduction to IF first-order logic see Hintikka (1996) and Sandu and Sevenster (forthcoming)

existential quantifier $(\exists y)$ to be independent of a universal quantifier $(\forall x)$ within whose formal scope it occurs. This is expressed by writing it as $(\exists y / \forall x)$.

But IF logic uses only a part of the available freedom from the fetters of the received first-order notation. In order to formulate an even more general logic, it is advisable to start not from syntactic structure of first-order sentences but from their semantic dependence structure. For this purpose the usual semantics is not very appropriate. It proceeds compositionally from inside out and is therefore incapable of capturing in a simple and natural way dependence structure.[2]

2. *Game theoretical semantics as a codification of dependence*

Instead what is known as game-theoretical semantics (GTS) is much better suited to the task.[3] There each occurrence of a logical constant prompts a move in a game. The dependence or independence of a quantifier (or other logical constant) on another quantifier or constant is expressed by the dependence (or independence, respectively) of the moves that they prompt of each other. Dependence and independence is here understood as informational independence in the sense of game theory. What this means is that the player making a "dependent" move knows what the move is that it "depends" on. For instance, consider the semantical game associated with the sentence

(2.1) $(\forall x)((\exists y)L(x,y) \,\&\, (\exists z)\mathrm{H}(\mathrm{z},\mathrm{x}))$

In this game, the falsifier chooses a value for the variable x, without knowing of the choices of the values of y and of z by the verifier. The former of these is made with knowledge of the value of x but not that of z, the latter with knowledge of the value of x but not that of y. In brief, dependence patterns are turned into patterns of information flow in GTS.

The naturalness and power of GTS is shown by the fact that it yields simple and natural truth conditions for quantified sentences. A natural truth condition for a quantificational sentence is the existence of the "witness individuals" that show its truth. For a sentence of the form $(\exists x)$ F[x], a witness individual b is an individual that satisfies F[b].

However, such witness individuals typically depend on other individuals. For instance, the witness individuals for a sentence of the form $(\forall x)$ $(\exists y)$ F[x,y] are the individuals b such that for each given a satisfies F [a, b]. Here it is a function of a. Thus the existence of witness individuals includes the existence of such dependence functions. For instance, for $(\forall x)$ $(\exists y)$ F[x,y] such a witness function f is one that satisfies for each individual a the formula F[a, $f(a)$]. These functions are known as Skolem functions. The natural truth condition for a sentence S is therefore the existence of its Skolem functions (including constant functions like the b just mentioned). The Skolem functions of S express all the different modes of dependence involved in S. They are the main ingredients of the logical form of first order sentences.

---

[2] Here, it is assumed that all the formulas considered are in the negation normal form. In other words, the only connectives are &, ∨, and ~, and all negation signs ~ are prefixed to (negated or unnegated) atomic formulas or identities.

[3] For an introduction to GTS see Hintikka and Sandu 1996

The semantical game $G(S)$ connected with S is sometimes described as a verification game. This terminology is suggestive, but not quite accurate. Rather $G(S)$ is an attempt to locate a part of the universe of discourse (model) in which S is true, against the attempts of an opponent who is trying to defeat the attempt by introducing individuals from the universe of discourse that have to be incorporated in the model. This part of the model need not be a proper part. Therefore it can be the entire world or model in which our sentences are evaluated for their truth. It is not unnatural to call the players of the two person zero-sum game "the verifier" and "the falsifier" or in brief $\underline{V}$ and $\underline{F}$ respectively.

From the game idea, the precise rules of $G(S)$ are easily gathered. They can be formulated as follows:

A game with S_o is played on a model (world) M with the domain D (M)in which all its nonlogical notions have been interpreted. There are two players, $\underline{V}$ (the verifier) and $\underline{F}$ (the falsifier). At each move, the players are considering a sentence S, beginning with S_o.

The rules governing the moves are as follows

(R. v) If S is (F_1 v F_2), the verifier chooses F_1 or F_2. The game is continued with it

(R. &) If S is (F_1 & F_2), likewise, except that the choice of F_1 or F_2 is made by the falsifier.

(R. E) If S is $(\exists x)$ F[x], the verifier chooses a member b of $D(M)$. The game is continued with respect to F[b].

(R. ~) If S is ~F, the two players exchange their roles, as defined by these rules. The game is continued with respect to F

(R. =) The substitutivity of identicals

In a finite number of steps the players end up with atomic sentence A. Since its nonlogical constants have been interpreted, A is true, false or neither. Accordingly the outcome of the play of the game is the verifier's win, the falsifier's win, or a tie.

The game can be taken to be a zero-sum game.

These rules can be motivated in terms of the interpretation of the game as described earlier. The best strategy for the verifier is to try to choose always a true sentence and for the falsifier to choose a false one.

3. *Game theoretical semantics as a study of dependence patterns*

The subtlest feature of GTS is the definition of truth. Michael Dummett once wrote that truth is like winning in a game. (1978) We disagree. In GTS, truth is defined as the existence of a winning strategy for the verifier $\underline{V}$. The truth of S is not like winning a verification game with S. It is being in a position of winning $G(S)$ no matter what one's opponent does in this game.

This way of defining truth is remarkable in several ways. For one thing it makes no reference to the particular language which sentence S is, that is to quantifiers or propositional connectives. It can therefore be extended to the semantics of any notion whose meaning can be characterized game theoretically.

It is also a more abstract notion than might first seem to be the case. Strictly speaking in the GTS explained above, to assert *S* is to make a purely existential statements, that is to claim that there exists a winning strategy for *V*. Of course, a statement by an actual speaker normally has a greater force for it creates the expectation that the speaker knows what the winning strategies are like. But this is not a defect of our definition which in fact opens the possibility for interesting further distinctions. Usually, when a speaker asserts something, say *S* the presumption is that she or he knows (perhaps in a rather weak sense) the truth of *S*. (This is illustrated by the epistemic forms of Moore's paradox.) But knowing the truth of *S* is normally taken to mean, not knowing that there exists a winning strategy for *V* in *G*(*S*), but actually knowing at least one such strategy.

This suggests an interesting new analysis of knowledge. It also explains pragmatically why there is a stronger force to actual assertions than the abstract meaning just defined.

There are interesting relations between the key notions of this study and traditional concepts of logical theory. For example, it is not very clear what has been meant or what should be meant by "logical form". Whatever sense earlier discussions might have intended, by "logical form" the most important systematic significance that the notion has is the dependence structure examined here.

It is important to realize that GTS is not a theory of certain games labeled "semantical". Rather, these games offer an analogue to patterns of variable dependence by reference to which these patterns can be studied. One must therefore be careful not to attribute to the dependence structures everything that can be said of semantical games. For instance, the game theoretical unrestricted notion of information set does not always make interpretational sense as applied to dependence patterns. In terms of the game analogy, of course a player knows in which node of the game tree (in extensive form) she or he is in. But the player perhaps does not know how she or he got there, in other words, what happened at the earlier moves in the same play of the game.

In any case, certain possible dependence patterns cannot be analyzed game theoretically along the lines typically followed by logicians. Not only is it thinkable that there might be correlated variables so strongly mutually dependent that they cannot each be represented separately as functions of some third variable. It is known from quantum theory that noncommuting variables behave in this way. Such mutual dependence obviously cannot be represented analogically by the semantic games defined above. Even the "general" logical theory outlined here therefore has its limitations. However, they can be overcome by admitting mixed strategies into semantical games. This line of thought is not pursued here any further however. **(**See here Hintikka 2008 and Sandu and Sevenster 2008.)

One apparent difference between patterns of actual dependence relations and patterns of informational dependence in semantical games is that actual material dependence is seemingly transitive whereas informational dependence in games need not be. This difference is merely apparent, however. For the relations of dependence that hold between two particular moves are only relations of partial dependence . What a move in a semantical game depends on is a combination of all the different earlier moves that the player in question is aware of. It is only such total dependence that is transitive. The "dependence" relation" in the sense of GTS indicates a player's awareness of earlier moves. It is not always transitive, although its ancestral is.

4. Expressing dependence patterns

The crucial question now is what dependence (information flow) pattern the moves in a semantical game can exhibit. Very few restrictions are in fact needed. A player's move can, of course, be independent of an earlier move or of one's opponent. In first order IF logic it is usually developed only independencies of existential quantifiers from universal quantifiers further outside are admitted. The implicit rationale of this restriction is examined later in this essay. This restriction is by no means unavoidable since in game theory the so-called players can be teams of agents who cannot pool their information, a "player's" move can be independent of the same "player's " earlier moves. Hence the only restriction is that the moves have to take place in linear time. This can be taken to rule out, among other requirements, mutual dependence.

This irreducibility of dependence structure to a linear order poses the problem as to how the dependence structure can be represented syntactically. Each move is prompted by a quantifier or a propositional connective. The dependence structure of the moves connected with them can be imbedded in a linear structure because the ancestror of the dependence relation is transitive. The order of moves in this linear order may be thought of as their temporal order.

This linearization involves indicating separately the dependence and independencies between different moves. There are two ways of doing so. In the first, it is assumed that all quantifiers and connectives depend ceteris partibus on quantifiers and connectives before them in the ordering. Exceptions are indicated by the slash / as is usually done in IF logic. Thus the independence of (Q_2y) of (Q_1x) is expressed by writing it (Q_2y/Q_1x), and likewise for quantifiers.

The other method is to assume that ceteris paribus all quantifiers are independent of each other. Then the dependence of (Q_2y) of (Q_1x) which occurs to the left of it is expressed by writing it as $(Q_2y//Q_1x)$, and likewise for connectives.

The two methods are equivalent. In this paper, the former is used unless otherwise indicated.

Each conjunct and each disjunct defines a subgame. The games with different conjuncts (or different disjuncts) are parallel rather than played one after another.

There is nevertheless no reason to exclude interaction between the (sub) games connected with different parallel disjuncts (or parallel conjuncts). But since there must be mutual dependencies between different conjuncts or disjuncts, their left to right order can reflect the direction of dependencies.

If there are dependencies in the moves prompted by the ingredients of one disjunct (or conjunct) say D_2 on the moves connected with another disjunct say D_1 they cannot be indicated by parentheses as they are usually employed. In this usage, a disjunct is marked off from other disjunct by parentheses (brackets) around it. At the same time, dependencies of different operators are expressed by means of the nesting of their scopes. If we try to do both here, the scope of the brackets demarcating disjuncts and the scopes of the quantifiers (or propositiona connectives) in another disjunct will overlap and hence cannot be nested.

However, such situations do not present any difficulties for our independence (slash) notation. There we do not need brackets to express dependence, and brackets are used only syntactically to distinguish different disjuncts and conjuncts from each other.

In the received notation for first-order logic, the brackets associated with a quantifier have another function. They are supposed to indicate the limits of the segment of a formula in which the quantified variable is bound to that particular quantifier. This means assuming, obviously tacitly – that these two tasks can be performed by the same pair of brackets. This assumption is nevertheless unacceptable as a general principle. There is no reason to believe that the limits of binding are somehow correlated with the dependence structure. It is in fact easy to see that this is not always the case in natural language. Hence it is simply an unnecessary restriction to require such tandem operation in a logical notation.

The syntactical manifestation of the two tasks are what Chomsky apparently was trying to capture in his government and binding theory. If this was his aim, his theory was bound to be incomplete. For sometimes the dependence (government) structure depends on the lexical items in question. An example is offered by the difference in the logical behavior of the two universal quantifiers *any* and *every*.

In the rules for the moves in semantical games the origin and the past history of the variables and their substitution-values does not make any definitional difference. This means that binding can be expressed simply by the identity of the variable in question. That is what will be assumed in the rest of this paper. This procedure obviously presupposes that a different variable is used in all different (occurrences of) quantifiers. It is not so far from what happens in natural languages where variable binding is not by parentheses but by syntactical relations between an anaphor and its head.

It is to be noted that brackets have another different function which will be used here. It is to separate conjuncts from each other and similarly for disjuncts.

The resulting essentially game theory- free notation is closer to natural language that the conventional formalism of first-order logic. There are no parentheses in natural language whose semantical function would be the same as in conventional first-order logic. Yet many subtle logical forms (dependence structures) are expressed in natural language. How is this possible? We will return to this problem separately.

This suffices to explain how different dependence patterns, in other words, different logical forms can be expressed syntactically. It can be pointed out that parentheses are needed only to separate conjuncts and disjuncts from each other. They are not needed to indicate the binding scopes of quantifiers.

5. Dependence patterns as logical forms

The nature of the dependence structure as the genuine logical form is manifested in the fact that important logical operations, such as negation and conditionalization, can be traced back to operations on the dependence structures of the sentences in question.

Indeed, even new (or perhaps not so completely new) logical operations can be defined by reference to the dependence structures. As was explained earlier, given a semantical game *G* (*S*) connected with a sentence *S* with a certain dependence structure, *S* is true iff there is a winning strategy for the verifier $\underline{V}$, that is, a strategy which leads to a win for V̲ no matter which falsification F̲ chooses. *S* is false If there is a winning strategy for the falsifier. If neither one is the case that is if *G*(*S*) is not determined, *S* has an indefinite truth value.

Now, there is a closely related sentence which is true iff there is a winning strategy for V̲ in the model (world structure) in question. Is will be called the affirmation A(S) of S. It can be considered as a truth condition of *S*. It is instructive to see what the operation is that takes us from *S* to A(S). It might seem that in order to express A(S) we would have to quantify somehow over the falsifier's strategies. For we want to say that a winning strategy leads to a win for the verifier against any strategy of the falsifier. Such quantification can be avoided, however, by letting the falsifier F̲ make moves without any restrictions that is, with full information of earlier moves. This can be expressed by removing all independence indicator slashes from $(\forall x)$ and &. That this is what is needed here can be seen from earlier explanations. The temporal order of moves can be thought of as being expressed by the left-to-right order. Quantifiers and connectives depend on all quantifiers on the left, except when excluded from the information flow by the slash. Hence the removal of slashes secures perfect information.

This throws some light on the nature of IF logic. This logic is characterized by the fact that all independences are independences of existential quantifiers of universal ones. Such independences are not affected in the transition form S to A(S). Hence IF sentences are the sentences *S* for which *A*(*S*) is true iff *S* is. This suggests that we might consider the original IF logic as a special case of the general logic considered here, viz. as the logic of affirmations A(S). There seems to be some ambiguity in the earlier discussion.

It has sometimes been said that to assert a sentence S with game theoretic semantics is to claim that there exists a winning strategy in G(S) for the verifier. This is true to the extent that the asserter is not claiming to know any such strategy. But in another sense it is somewhat inaccurate. The assertion is true iff such a strategy does exist. But it is false, not always when no such strategy exists but only when the falsifier has a winning strategy.

In any case, we can see why the original IF logic was felt to be so natural. Even though general first-order sentences do not reduce to IF first order level the conditions of their truth are like IF inferences in that they do not contain any dependences of $(\forall y)$ on $(\exists x)$.

Among the other logical operations that can be characterized by reference to dependence structures is negation or rather, different kinds of negation. Now, the failure of tertium non datur implies that we have to distinguish the strong (dual) negation ~ from the contradictory negation ¬. The former can be defined for any sentence S as corresponding to the game G (S) with the roles of two players exchanged. The falsity of S then expressed by ~S, which means the existence o a winning strategy for the falsifier F̲.

The contradictory negation ¬S of S means that there does not exist a winning strategy for V̲ in G (S). Since semantical games are not determinate in general, this does not imply that a falsifier has a winning strategy. This explanation only makes sense for a sentence initial ¬. The meaning of ¬ in an arbitrary position of a sentence will be explained later.

The different sequences of negations can be reduced to six viz the empty sequence, ~, ¬, ¬~, ¬¬, and ¬¬~, Game theoretically the first three say respectively that F has a winning strategy, that *V* does not have one, and that *F* does not have one, The sentence ¬¬S is true iff S is true, false otherwise. ¬¬~S is true iff S is false, and false otherwise.

It is of some linguistic interest to note that while the doubling of the strong negation cancels it, a doubled contradictory negation (which presumably is the most common meaning of negation in ordinary language) does not reduce to the original unnegated sentence.

6.Dependence patterns of logical notions

It can now be seen what changes in the dependence pattern (logical form) characterize the negations ¬S and ~S respectively. We can likewise characterize in this way negation combinations like ¬~S, ¬¬S, and ¬¬~S. Once the dependence patterns involved in them are determined, the syntactic forms of expressions for such propositions can be formed in the way explained above. The transition from S to the affirmation A(S) explained provides a model for the analysis (and synthesis) of the relevant dependence structures (For negation, cf., here Hintikka 2006)

Consider first ~S. It is true iff there exists a winning strategy for the falsifier, false if there is a similar strategy for the verifier and indeterminate otherwise. This can be expressed simply by reversing the roles of *V* and *F* in G(S). And this can be done by interchanging ∃ and ∀ as well as v and &, plus adding ~ to the negation prefix of each atomic formula or identity.

The stronger statement that ~S is positively true (S positively false) is expressed by A(~S).

Consider next ¬ S. What it says is that there does not exist a winning strategy for V in G(S). In other words, for each strategy of the verifier there exists a strategy of the falsifier that defeats it. That is leads to a win for F. This might seem to involve quantification over strategies and hence go beyond what can be expressed by speaking of dependence structures alone . This is the same problem we encountered in defining A(S) and it can be solved in the same way. The effect of letting the verifier use any strategy whatsoever use any strategy can be reached by freeing the verifiers moves from all informational restrictions. Formally, this means omitting all the independence indicator slashes / from (∃x) and v.

After that, we must exchange the roles of the players in the same way as in forming ~S, in other words, exchange ∀ and ∃ as well as v and &. Also, the contradictory negation ¬ must be added to each negation prefix of an atomic formula or identity.

It can be seen that if (¬S) so characterized is not true, then it is false, never indeterminate.

If a formula of the form ¬F occurs as a subforumla of a longer formula an additional explanation is needed. Interpretationally ¬F says that there is no winning strategy for the verifier in the game G(F). But speaking of such a game makes sense only if what happens in it is independent of the context. Hence in a context forming ¬F means making all the quantifiers and connectives independent of the quantifiers and connectives outside ¬F within whose syntactical scope they occur.

Interpretationally speaking, ~S is true iff F can always win even if V knows which strategy F is using. In contrast, ¬S is true iff F can always win if F knows which strategy V is using.

In this way the semantics of the two negations becomes completely transparent.

As a consequence, we can see what kind of dependence patterns there is for ¬¬S. Interpretationally this means making both players aware of all the previous moves. Notationally, it amounts to removing all slashes / . Systematically, it means a logic with tertium non datur. Thus we obtain the received first order logic as a special case of the general logic explained here as the logic of sentences of the form ¬¬S.

The following example illustrates these explanations. Let L (x,y) say "x loves y" and let S be the sentence

(6.1) $(\forall x)(\forall z)(\exists y/\forall z)\ (\exists u/\forall x)\ ((x\neq z) \supset (L(y,x)\ \&\ L(u,z)\ \&\ (\sim L(y,z)\ \&\sim L(u,x)))$

This says that each person has a unique lover.

Then ~S is according to the rules explained above

(6.2) $(\exists x)(\exists z)(\forall y/\exists z)\ (\forall u/\exists x)\ ((x\neq z)\ \&\ (\sim L(y,x) \vee \sim L(u,z) \vee L(y,z) \vee L(u,x)\ )$

This can also be written

(6.3) $(\exists x)(\exists z)(\forall y/\exists z)\ (\forall u/\exists x)((x\neq z)\ \&\ ((L(y,x)\ \&\ L(u,z)) \supset (L(y,z) \vee L(u,x)))$

Reflecting on (6.3) one sees that it can only be true if there is a couple who have all their lovers in common.

According to what has been said, ¬*S* is

(6.4) $(\exists x)(\exists z)(\forall y)(\forall u)((x\neq z)\ \&\ \neg L(y,x) \vee \neg L(u,z) \vee\ L(y,z) \vee L(u,x)))$

This is equivalent to

(6.5) $(\exists x)(\exists z)(\forall y)(\forall u)(x \neq z\ \&\ (L(y,x)(L(y,x)\ \&\ L(u,z)) \supset (L(y,z) \vee L(u,x))))$
$(y,z)(\vee/\exists x,\exists z)L(u,z)(\supset\exists x,\exists z)L(u,x)))$

What this says is that there is a couple such that of any pair of their respective lovers one or the other is also a lover of the other member of the couple. It is interesting to see here how the independence indicators in (6.3) make it stronger than (6.5)

Furthermore, ¬¬S will be

(6.6) $(\forall x)(\forall z)(\exists y)(\exists u)((x\neq z) \supset (L(y,x)\ \&\ (L(u.z))\ \&\ \neg L(y,z) \vee \neg\ L(u,x))$

This says that each member of any couple has a lover that the other one does not have. Notice how (6.6) is weaker than (6.1).

In sum, what has been found shows that the truth conditions of a sentence S depend only on the dependencies and independences of the verifier moves on the falsifier moves in G(S). However, the reverse (independencies matter for the conditions of the falsity of S, that is of the truth of ~S depend only on the dependencies of the verifiers moves in G(¬S), hence only on the falsifiers moves in G (S)

7. *Reduction to first order level*

We have thus outlined a system of logic, not in the sense of a deductive system, but in the more general sense of a semantically interpreted formalism (language). It will be called the general logic or general first-order logic. It was seen to contain the conventional first-order logic as a part. It also contains the usual IF first-order logic and also what has been called extended IF first-order logic. It is essentially the same as what has been called fully-extended IF logic. However, this term would be misleading because it hides the generality and simplicity of general logic. This logic is not one of the ever so many "alternative logics." It is not an alternative to the conventional Frege Russell logic of quantifiers. It is a liberated and generalized form of the classical logic. As was pointed out its leading ideas are rooted in the very meaning of the quantifiers.

As the general first-order logic is formulated here, it can be developed further, in more than one direction. In one of them the problem of mutual dependence can be overcome. This prompts more sweeping new conceptualization than can be explained here.

The possibility of general logic has major consequences in more than one direction. First, this logic makes second order logic and thereby for all practical purposes the entire higher-order logic dispensable for traditional foundational purposes. The job of second-order logic is done on the first order level in our general logic.

This can be shown by showing how second-order formulas can be translated in the language of general logic. The basic idea of this translation is seen in the following example from the equivalence of the following two sentences;

(7.1) $(\exists f)(\forall x)\, F[x, f(x)]$

(7.2) $(\forall x_1)\, (\forall x_2)\, (\exists y_1/\forall x_2)\, (\exists y_2/\forall x_1)\, (((x_1=x_2) \supset (y_1=y_2))\, \&\, F[x_1,y_1]\, \&\, F[x_2,y_2$

Here F[x,y] can be any general first-order formula with x and y as its only free individuals. The equivalence of (7.1) and (7.2) can be seen by expressing (7.2) in its Skolem function form:

(7.3) $(\exists f_1)\, (\exists f_2)\, (\forall x_1)\, (\forall x_2)\, ((x_1=x_2) \supset (f_1(x_1) \neq f_2(x_2)))\, \&\, F\, [x_1, f_1(x_1)]\, \&\, F\, [x_2, f_2(x_2)])$

The first conjunct in (7.3) shows that $f_1 = f_2$. Hence (7.3) reduces to (7.1).

This reduction works for any number of initial existential function quantifiers $(\exists f_1)$, $(\exists f_2)$… However it is restricted by the fact that f must have one and the same argument or arguments in the entire initial second order sentence. This limitation can be overcome by introducing a new existential function quantifier for each different kind of occurrence of a function with different arguments. The relations of these quantified function variables to each other and to the old ones can easily be spelled out by means of general first-order formulas.

For instance we can replace a sentence of the form

(7.4) $(\exists f)(\forall x)(\forall y)\ F[x,y\ f(x), f(y)]$

by the sentence

(7.5) $(\exists f)\ (\exists g)\ (\forall x)(\forall y)\ ((x=y) \supset (f(x) =g(y))\ \&\ F[x, y, f(x), g(y)]$

Here f and g each has everywhere the same respective arguments.

We can likewise take case of the nesting of functions. For instance:

(7.6) $(\exists f)(\forall x)\ F[x, f(f(x))]$

can be replaced by

$(\exists f)\ (\exists g)\ (\forall x)(\forall z)\ (f(x) =z) \supset F[x, g(z)])$

By these means, we can eliminate from a given second-order sentence S all second-order quantifiers proceeding from the inside out. This elimination is a generalization of the well known translation of the Σ_1^1 fragment of second order logic to IF first-order logic. Here it yields an important result. It shows that the foundational job of the entire second-order logic can be done by our general first-order logic. In effect, second-order logic becomes dispensable. Since second-order logic can do the job of set theory, by implication set theory becomes dispensable too. This has striking consequences. Since set-theoretical reasoning is usually taken to be all that is needed in mathematics, it turns out that our general logic is very nearly the entire logic of mathematics. All mathematical reasoning can be taken to be nominalistic, that it, reasoning about structures formed by particular objects. This realizes Hilbert's one time dream which is arguably more important philosophically than the usually misunderstood "Hilbert's program". Hilbert (1922) blamed all the problems in the foundations of mathematics on the use of higher-order notions. We have in effect eliminated the need of assuming such entities in the sense of using them as values of the variables of quantification.

It might even be suggested that general first-order logic is at bottom not very far from mathematicians' informal working logic. For in both, we are typically dealing with ordinary looking first-order quantification. The force of general logic comes out of the relations of independence. This notion of independence might seem to be the main new and unprecedented feature of our general logic. Yet the fact is, even though historians of logic and mathematics have not paid much attention to it, rife with uses of the notion of independence. Existential quantifiers are in the informal jargon of mathematicians expressed by locutions like "one can find" or "one can choose". But every once in a while, those locutions come with the extra claim "independently of…". In fact, the uniformity concepts (uniform continuity, uniform differentiability, etc.) are cases in point. Such notions cannot be expressed without using our general first-order logic or at least IF logic. Also, combinatorial theory offers plenty of examples of the use of the idea of independence.

The general first-order logic we have formulated has many interesting applications. We hope to explore some of them in subsequent papers. In its own right, there are two directions in which our general logic can be generalized further. One is to introduce a notion of probability into our logic and semantics. Work in this direction has already been done by Hintikka (2008) and by Sandu and Sevnster, (forthcoming (b)). The other involves extending it to deal with mutual dependence variables. This turns out to be a major task that requires the development not only of new logic but in a sense new mathematics.

References

Dummett, M. 1978 *Truth and OtherEnigmas.* Harvard University Press: Cambridge MA.

Hilbert, David, 1922, “Neubegründung der Mathematick, Erste Mittelung”, *Abhandlungen aus dem mathematischen Seminar der Hamburgischen Universitatät.* vol. 1, pp. 157-177. In English in W.B. Ewald editor, *From Kant to Hilbert* vol. 2, Clarendon Press, Oxford, 1996, pp. 115-134.

Hintikka, Jaakko, 1996, *The Principles of Mathematics Revisited*, Cambridge U.P., Cambridge.

Hintikka, Jaakko, 2008, “IF logic in a wider setting”, in C. Dégremont, L. Keiff and H. Rickert, editors, *Dialogues, Logics and Other Strange Things,* College Publications, London, pp 195-210.

Hintikka, Jaakko, and Gabriel Sandu, 1996, “Game-Theoretical Semantics”, in J. van Benthem and Alice ter Meulen, *Handbook of Logic and Language*, Elsevier, Amsterdam, pp. 362-10.

Sandu, Gabriel, and Merhije Sevenster, forthcoming, *Games and Logic: A Strategic Account of IF Logic*, Cambridge U.P., Cambridge.

Sandu, Gabriel, and Merhije Sevenster, 2008,