Dependencies between quantifiers vs. dependencies between variables

Gabriel Sandu

January 7, 2019

One of the revolutionary aspects of modern logic consists in considering valid inferences that involve multiple quantification. In this case one needs to consider quantifiers that appear in the scope of other quantifiers. In this paper I consider two kinds of dependencies: scopal dependencies between quantifiers and material dependecies beween (the values of) variables. Some focus will be put on the discussion of mutual dependencies of both kinds.

1 The other function of quantifiers

One of the revolutionary aspects of modern logic consists in considering valid inferences that involve multiple quantification. In this case one needs to consider quantifiers that appear in the scope of other quantifiers. Within the sequence of quantifiers in a formula is linearly ordered, one indicates the scopal dependency of a quantifier on other quantifiers in a syntactic way by writing the former after the latter. The formal, scopal dependencies between quantifiers are indications of material dependencies between the values of the quantified variables in an underlying universe of discourse. The way these material dependencies are specified depends on the semantic representation. Each such representation has to solve the challenge that comes from the need to combine a semantic mechanism which corresponds to the "ranging over" semantic job of a quantifier and thereby considers one quantifier at a time, with a distinct mechanism that "glues" the successive steps together.

In the Frege-Tarski tradition this challenge is solved in a relatively straightforward way. For any formula in which the quantifiers are linearly ordered one can consider only one singly quantified formula at a time and still account for the dependency of a quantifer on the previous one in the sequence. To illustrate, consider the scheme

1. $Q_1xQ_2yR(x,y)$

where Q_1x and Q_2y are any two (generalized) quantifiers. The interpretation of this sentence is specified in the following steps: (1) is true if and only if there are are q_1 $a's$ for each one of which there are q_2 b's such that $R(a,b)$. Here q_1

and q_2 are the semantic conditions associated with the generalized quantifers Q_1, and Q_2, respectively. (It is not easy to give an English paraphrase; 10) Here the material dependence of the values of the variables, the "gluing together" is very weak being achieved by a relative expression synonymous with "for each one of which." One of Sher's examples is

2. Three frightened elephants were chased by a dozen hunters

represented by

3. $3x12y\,(E(x) \wedge H(x) \wedge C(x,y))$

where $3x$ is the generalized quantifier interpreted in a universe of discourse M by the set of all subsets of M with three elements, and the quantifier $12y$ is interpreted analogously. Thus (3) says that there are three frightened elephants, for each one of which there are 12 hunters such that every hunter chases it.

In the case in which Q_1, and Q_2 are the standard quantifiers $\forall$ and $\exists$, respectively, as in the sentence

4. $\forall x \exists y R(x,y)$

its truth-conditions state the existence of appropriate sets which are introduced sequentially: there exists a set X consisting of the whole universe such that for each of its elements a, there is a non-empty set Y such that a stands in the R-relation with each element of Y. The two sets can actually be composed into one binary relation S, making the truth-condition equivalent to: there is a set S such that for each a there is at least a b such that $R(a,b)$. Under some weak set-theoretical assumptions, the last condition can be shown to be equivalent to the more familiar: for each a there is at least one b such that $R(a,b)$. As we see, the material dependency between the values of the variables 'x' and 'y' is rather weak, in the sense that any value of x does not constrain in any way the corresponing value of y except "externally" through the relation R (we assume that...). In other words, the material dependencies of the values of the variables take the form of a tree were each arrow starting from the root points to an individual which represents a possible value of x, which in turn is connected by arrows to all the individuals (leaves) with which it stands in the R-relation.

The dependencies we are interested in in this paper are arbitrary dependencies between standard first-order quantifiers; in particular we are interested in the scopal dependencies of an existential quantifier on a sequence of other standard quantifiers. When these dependencies can be linearly ordered, as in (4), the semantic interpretation may follow the same pattern as that given for (4). An increase in the number of quantifiers may lead, however, to scopal dependency patterns which cannot be linearized. One of these patterns, discovered long time ago by Henkin (4), involves four quantifiers:

- For every x and x', there exists a y depending only on x and a y' depending only on x' such that $Q(x,x',y,y')$ is true (here $Q(x,x',y,y')$ is a quantifier free formula).

Henkin represented the four quantifiers prefix in a branching form:

$$\begin{pmatrix} \forall x & \exists y \\ \forall x' & \exists y' \end{pmatrix} Q(x, x', y, y')$$

to emphasize that $\exists y$ is only in the scope of $\forall x$ and $\exists y'$ is only in the scope of $\forall x'$. Finding a semantic interpretation for the branching prefix is not trivial. **(author?)** (1) proposed a general scheme (for monotone quantifiers) which respects the spirit of the interpretation given by (4), except that now, to account for the partial ordering of scopes, the relevant sets are not introduced sequentially but right from the beginning. When these sets are combined into corresponding relations, the result may be expressed in second-order logic by the sentence

$$\exists R \exists S \left(\forall x \exists y R(x, y) \wedge \forall x' \exists y' S(x', y') \wedge \forall x \forall x' \forall y \forall u \left(R(x, y) \wedge S(x', u) \rightarrow Q(x, x', y, y')\right)\right).$$

This is not the semantic interpretation chosen by Henkin for the branching quantifier. Henkin's interpretation is based on a stronger dependency between the values of the variables, i.e., functional dependence, which goes back to Skolem. Perhaps the best way to introduce it is with respect to our earlier example (4). In this case the scopal dependency of $\exists y$ on $\forall x$ induces a functional correlation between the values of 'y' on the values of 'x': for each a, which is a value of 'x' there is exactly one b, which is a value of 'y' such that $R(a, b)$. This correlation is assumed to be given by an unspecified function f so that the truth-conditions of (4) may be now expressed by the second-order sentence:

5. $\exists f \forall x \left(y = f(x) \wedge R(x, y)\right)$

or equivalently

6. $\exists f \forall x R(x, f(x))$.

The function f is called a *Skolem function*.

Generalizing the Skolem functions approach to branching quantifiers yields Henkin's initial interpretation:

$$\begin{pmatrix} \forall x & \exists y \\ \forall x' & \exists y' \end{pmatrix} Q(x, x', y, y') \Leftrightarrow \exists f \exists g \forall x \forall x' Q(x, x', f(x), g(x'))$$

2 Game-theoretical semantics (GTS)

Henkin's interpretation of branching quantifiers based on the generalization of the Skolem functions approach motivated Hintikka's game-theoretical interpretation of first-order quantifiers and connectives. A (semantic) game is associated with any first-order sentence and underlying model which interprets the non-logical constants of the sentence. The game is played by two players, the Verifier (whose moves corresponds to existential quantifiers and disjunctions) and the

Falsifier (universal quantifiers and conjunctions). As an example we consider the game associated with the sentence φ

$$\forall x(\exists y L(x,y) \wedge \exists z H(x,z))$$

and a model M. The Falsifier chooses an individual from the universe of M, say a, to be the value of 'x' after which he has the choice between the left and the right conjunct. If the former, the Verifier chooses an individual, say b, to be the value of 'y' and the play stops. The Verifier wins the play if $L^M(a,b)$; otherwise the Falsifier wins. If the right conjunct is chosen, the Verifier chooses an individual, say c, to be the value of 'z' and this play stops here, with similar conditions for winning and loosing.

Now as we see from the example, a quantifier or a connective being in the scope of another quantifier amounts in the game-theoretical setting to the *informational dependence* of the move prompted by the former on the move prompted by the latter. It is codified by the notion of *information set* associated with a given move, an epistemic notion which indicates which other moves the player making that move is aware of. Thus the the scopal dependencies of quantifiers (and connectives) map directly into the knowledge of the players codified by information sets.

On the other side, the truth (falsity) of a sentence S in a model M is defined as the existence of a winning strategy for the Verifier (Falsifier) in the appropriate semantic game. When the strategies are defined determistically (functionally), a winning strategy for the Verifier (when it exists) decomposes into Skolem functions. They express the material dependencies of the appropriate quantified variables. Referring to Skolem functions, say f, whenever $b = f(a)$ Hintikka thought of $b$as a *witness individual* which depends ontologically on the individual a. Thus the game-theoretical framework has two levels, each with its own notion of dependency. The epistemic level of information sets which map isomorphically the scopal dependencies of quantifiers and connectives. The ontological level of strategic functions which create a network of material dependencies between the individuals (values of the quantified variables) of the underlying universe.

3 Game-theoretical semantics as a basis for general logic

The question asked in **(author?)** (6) was: How can one extend scopal quantifier dependencies in order to express more material dependencies between the values of the quantified variables? Given the game-theoretical setting in the background, this question may be rephrased as: What are all the possible patterns of information flow (quantifier dependencies) compatible with the game rules for quantifiers and connectives? Recalling that quantifier dependency maps into dependencies between moves in the relevant semantic game, two minimal conditions came quite naturally:

- A player's move can be informationally dependent only of an earlier move in the game
- The moves have to take place in linear time (linear time playability condition)

The two conditions are seen to be easily satisfied by the semantic games associated with first-order sentences where a move is informationally dependent on all the earlier moves in the game. Given that we wanted to extend the dependency patterns between quantifiers beyond the first-order ones, this is a condition we did not want any longer to assume, so we gave it up and with it the transitivity of the dependency relation. This together with the fact that the logical constants occurring φ are not in the scope of those occurring in ψ in the conjunct $\varphi \wedge \psi$, it was natural to think that the dependency relation governing scopes is an antisymmetric, partial, intransitive, and discrete ordering. The problem to be solved was to find a way to faithfully map this partial ordering onto a linear order which could be thought of as the temporal order of the moves in a semantical game. In this way the linear time playability condition would have been ensured. Notice that this condition is not fulfilled by Henkin's bramching quantifiers prefix.

The way we chose to implement this condition in **(author?)** (6) was to indicate separately the dependence (and independence) between moves. That is, we assumed that all the quantifiers and connectives depend *ceteris paribus* on all quantifiers and connectives before them in the ordering. The exceptions are indicated by the slash as in

$$\forall x \forall x' (\exists y / \{x'\}) (\exists y' / \{x, y\}) Q(x, x', y, y')$$

which is our representation of the branching quantifier. Here $(\exists y / \{x'\})$ indicates that $\exists y$ is not in the scope (is independent) of $\forall x'$ and thus, given the assumption, it is only in the scope of $\forall x$. Similarly $(\exists y' / \{x, y\})$ indicates that $\exists y$ is independent of $\forall x$ and $\exists y$ and thus it is only in the scope of $\forall x'$. Game-theoretically the first condition means that the Verifier does not know the value chosen by the Falsifier for 'x'; and the second condition means that the Verifier does not know the value chosen by the Falsifier for 'x'neither the one chosen by herself for 'y'. There are some subtleties here concerning the Verifier "forgetting" her own earlier moves but they will not be my concern in this paper. Notice, however, the time linearity of the 4 moves in the semantic game.

There would have been another way to implement the linear time playability condition, namely to assume that *ceteris paribus* all quantifiers and connectives are scopally independent of each other. The *ceteris paribus* condition is now represented by

$$Q_2 y \parallel Q_1 x$$

which indicates the scopal dependence of $Q_2 y$ on $Q_1 x$ which occurs at its right, and likewise for connectives. The two representations are equivalent. We chose the first one, and we called the result IF first-order logic.

4 Dependency of quantifiers: Independence-Friendly Logic

The IF sentence φ_{inf}

7. $\forall x \exists y (\exists z / \{x\})(x = z \wedge c \neq y)$

defines (Dedekind) infinity, a property which cannot be expressed in ordinary first-order logic. The (scopal) dependency relation between quantifiers is anti-symmetric and intransitive. The latter follows from the fact that $\exists y$ depends on $\forall x$ and $\exists z$ depends on $\exists y$ but $\exists z$ does not depend on $\forall x$ (in other words, in her move correspoding to $\exists y$ the Verifier knows the value chosen for x by the Falsifier, and in the move corresponding to $\exists z$ show knows her own earlier move but show she does not know the choice made by the Falsifier). Instead, the material dependency between (the values of) variables expressed by the Skolem functions $y = f(x)$ and $z = g(y)$ is transitive: one can define a new function which expresses the dependecy of the values of 'z' on the values of 'x': $h(z) = g(f(x))$.

Now returning to our earlier question "What are all the possible patterns of information flow (quantifier dependencies) compatible with the game rules for quantifiers and connectives?", we shall rephrase it, following **(author?)** (8) as:

- What patterns of scopal dependencies between quantifiers and correspondingly, what patterns of material dependecies between variables does the linear time playability condition exclude?

Hintikka & Symons' answer is: mutually dependent variables. The challenge is now to understand what such variables are. In **(author?)** (5), Hintikka talks about "strongly correlated variables which are mutually dependent so that they cannot be represented separably as functions of some third variable (non-commuting variables in quantum theory)". One of the examples Hintikka considers is the following IF-sentence

8. $\forall t \forall x \forall y \, (\exists z / \, \{x\}) \, (\exists u / \, \{y\}) \, [x = z \wedge y = u \wedge S(t, x, y)] \; .$

which expresses the scopal dependency of $\exists z$ on $\forall t$ and $\forall y$ and that of $\exists u$ on $\forall t$ and $\forall x$. Here the quantifier $\forall t$ ranges over moments of time. Making explicit the dependencies of the variables induced by the quantifier dependencies yield the following second-order sentence which expresses the truth-conditions of (8):

9. $\exists f \exists g \forall t \forall x \forall y \, [x = f(t, y) \wedge y = g(t, x) \wedge S(t, x, y)]$

(9) shows that $x$is a function of time and of the variable y whereas y is a function of time and of the other variable x.

Unfortunately Hintikka's example can be shown to be flawed: (9) can be true only in a model (set) with one element. To see this, suppose, for a contradiction, that (9) is true in a model which has two distinct elements, $a$and b. It is easy to see that in this case there are no functions f and g as described in (9). For let

$t = a$, $x = a$ and $y = a$. Then we must have $a = f(a, a)$. If on the other side, we let $t = a$, $x = b$ and $y = a$, we should have $b = f(a, a)$, which is impossible.

The problem here is that the scheme

$$\forall t \forall x \forall y \, (\exists z / \{x\}) \, ... \, [...x = z...]$$

can be true only in a model with one element. And by analogy the same holds of the scheme

$$\forall t \forall y \, (\exists u / \{y\}) \, ... \, [...y = u...]$$

If we ignore the time variable t, (9) becomes

10. $\exists f \exists g \forall x \forall y \, [x = f(y) \wedge y = g(x) \wedge S(x, y)]$

But (10) may be shown to lead to the same problem as before, that is, it can be true only in models with one element (for $x = a$ and $y = b$ we get $a = f(b)$ and $b = g(a)$; from $x = b$ and $y = b$ we get $b = f(b)$ which is impossible). As in the earlier example, the problem is with the scheme

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

which can be true only in a singleton set.

Perhaps considerations of this sort determined Hintikka to abandon later on (8) the claim that mutual dependencies of two variables, can be expressed in IF logic and to conclude that mutual dependency illustrates a pattern of variable dependency which cannot be analyzed game-theoretically "along the lines typically followed by logicians". In his paper with Symons Hintikka endorsed explicitly the linear time playability condition.

If I am allowed to speculate, I think that Hintikka wanted to get a mutual correlation between the values of two variables, 'x' and 'y' so that the values of' 'y' depend on those of 'x' in one way, and conversely, the values of 'x' depend on those of 'y' in (possibly) another way. The problem now springs from the fact that in order to get the first correlation, Hintikka needed something like

(i) $\exists y$ depends only on $\forall x$

and in order to obtain the second correlation he needed

(ii) $\forall x$ depends only on $\exists y$.

But taken jointly (i) and (ii) violate the linear time playability condition, which may have led Hintikka and Symons to the conclusion of their paper. Independently of Hintikka and Symons' motivation, I think that the question of the logical representation of mutually dependent variables is of independent interest. Hintikka tried to answer this question in the framework of IF logic, that is, through the scopal dependency conditions among quantifiers. It might help to approach the same question from a different but related angle.

5 Dependency of variables: Dependence Logic (11)

As already pointed out, the formal dependency between quantifiers induces a material, functional dependence between the values of the corresponding variables which is encoded in the semantics using (generalized) Skolem functions. For an illustration, we recall the equivalence between (8) and (9). But one can try to represent the functional dependency between the values of variables directly in the syntax compltely disentangled from the dependencies of the quantifiers which bind them. To this effect, **(author?)** (11) extends the syntax of first-order logic with *dependence atoms*

$$= (x_1, ..., x_n; y)$$

which have the intended interpretation: the value of y is (functionally) determined by the values of $x_1, ..., x_n$. The semantic unit of evaluation of a dependence atom is a team X, that is, a set of partial assignments sharing a common domain of variables with values in the universe of an underlying model. The semantic clause for a dependence atom is then expressed by

(i) $M \models_X = (x_1, ..., x_n; y)$ if and only if for all $s, s' \in X$, if s, s' agree on $x_1, ..., x_n$, then they also agree on y.

For an example, let $X = \{s_1, s_2, s_3\}$ be a team where the three assignments share the same domain $\{x, y, z\}$ as shown below:

	x	y	z
s_1	1	1	0
s_2	2	1	1
s_3	1	1	1

It is easy to check that $M \models_X = (x; y)$ but $M \not\models_X = (x, y; z)$. The clauses for complex formulae generalize the semantic clauses for first-order logic. We give here only the clauses for quantifiers:

(ii) $M \models_X \exists x\psi$ if and only if there is a function $f : X \to M$ such that $M \models_{X[x,f]} \psi$, where $X[x, f]$ is the team formed by extending each assignment s in X with $(x, f(s))$.

(iii) $M \models_X \forall x\psi$ iff $M \models_{X[x,M]} \psi$ where $X[x, M]$ is the team formed from X by extending each assignment s in X with (x, a) for each a in M.

Notice that this interpretation induces, as game-theoretical semantics, a functional dependency between the values of the existentiallly quantified variable x and the values of the other variables bound by the quantifiers in the scope of which $\exists x$ occurs.

Finally we define:

(iv) A sentence φ (formula with no free variables) is true in the model M if $M \models_{\{\emptyset\}} A$ where $\emptyset$ is the empty assignment.

It should not be too difficult to see, based on the definitions, that the truth of $\forall x \exists y R(x, y)$ is equivalent to the truth of the second-order sentence $\exists f \forall x R(x, f(x))$. But now the functional dependency between the values of 'y' and those of 'x' may be explicitly asserted in the object language:

$$\forall x \exists y R(x, y) \Leftrightarrow \forall x \exists y (= (x; y) \wedge R(x, y)) \Leftrightarrow \exists f \forall x R(x, f(x)).$$

The dependence atom $= (x; y)$ may be read off from the scopal dependency of the two quantifiers. Given the semantic interpretation of the quantifiers as described by the two clauses above, this atom is redundant and it may be omitted. The interesting cases (i.e. those which lead to an increase expressive power) are the ones in which the dependencies between variables differ from those induced by the scopal dependencies of quantifiers. For an example consider the sentence

11. $\forall x \exists y \exists z (= (x; y) \wedge = (y; z) \wedge x = z \wedge c \neq y)$

which is equivalent with

12. $\forall x \exists y \exists z (= (x; y) \wedge = (x, y; z) \wedge = (y; z) \wedge x = z \wedge c \neq y)$

and with the second-order sentence

13. $\exists f \exists g \forall x \left(x = g(f(x)) \wedge c \neq f(x)\right).$

The truth of (13) asserts the existence of an injective function f whose range is not the whole universe. In other words, (13) is true in a model M if and only if the universe of M is (Dedekind) infinite. Notice that the conjunct $= (x, y; z)$ is the one induced by the semantic interpretation of the quantifiers (and their scopal dependencies) and is thus redundant. But its redundancy follows also from $= (y; z)$. As a general principle, if $z$depends on y, then z depends on any larger sequence of variables which contains y.

6 Mutual dependency between variables

The last example should make it clear that dependency between variables is transitive, i.e., if y depends on x and z depends on y then $z$depends on x, reflexive, $x$depends on x, but not symmetric. That is, there are cases (e.g. many-one correlations) in which y depends on x but x does not depend on y. The mutual dependency of two variables can be expressed in a straightforward way:

$$= (x; y) \wedge = (y; x).$$

It should be clear that any team X which verifies $= (x; y) \wedge = (y; x)$ establishes a one-to-one correlation between the individuals which are the values of 'x' and

those which are the values of 'y'. To see this, we observe that the functional correlation f which associates the values of x with the values of y cannot be such that it sends two distinct individuals, say a and b to one and the same individual, say c, because the truth of the dependence atom $= (y; x)$ forces a and b to be identical. Using this fact, it can be shown (2) that (Dedekind) infinity can be defined in Dependence Logic:

14. $\forall x \exists y (= (y; x) \wedge y \neq c)$

(recalling that the formula $= (x; y)$ is redundant.)

It should be obvious that all the scopal dependencies of quantifiers in IF logic can be expressed using dependence atoms. It turns out that the converse is also true, i.e. dependence atoms can be contextually eliminated using scopal dependencies of IF quantifiers. Thus an occurrence of

$$\left(= (\overrightarrow{X}; y) \wedge ...\right)$$

in a sentence may be replaced by

$$(\exists z / W) \, (z = y \wedge ...)$$

where W is the set of variables dominating z minus $\overrightarrow{X}$.

Here is an example. The Dependence logic sentence

$$\forall x \exists y (= (y; x) \wedge y \neq c)$$

has the same truth-conditions as the IF logic sentence

$$\forall x \exists y \, (\exists z / \{x\}) \, (z = x \wedge y \neq c)$$

which is our earlier sentence (7) whose truth-conditions are expressed by (13). We take note that the IF counterpart obeys the "linear time playability condition" in the underlying game. But we also observe that one can induce a mutual dependency between the values of 'x' and the values of 'y' in IF logic, by introducing the extra quantifier $\exists z$ which is not in the scope of $\forall x$ but only in the scope of $\exists y$. We recall in this context Hintikka's endeavour to express mutual dependency by the IF sentence (we ignore the time variable)

8. $\forall x \forall y \, (\exists z / \{x\}) \, (\exists u / \{y\}) \, [x = z \wedge y = u \wedge S(x, y)]$.

As we pointed out earlier, the problem lies with the pattern $\forall x \forall y \, (\exists z / \{x\}) \, (...x = z...)$. One way to describe what we have tried to achieve in the last two sections is that in order to obtain the correct pattern, one should start with $\forall x \exists y$ which gives one side of the mutual correlation, and then, to get the other side, one either adds a dependence atom $= (y; x)$, or, equivalently, an IF quantifier $(\exists z / \{x\})$ together with the conjunct $x = z$.[1]

[1] I am indebted to Fausto Barbero for the material of the last two sections.

To conclude this section, if we take variable dependency as a basic feature of our general logic, then we obtain a substantial increase in expressive power when we combine it with the linear dependency of first-order quantifiers. The increasing growth in expressive power over ordinary first-order logic is due, as our example has suggested, to the mismatch between the two kinds of dependencies, one induced by quantifiers and the other one displayed by variables. As a result of this, the distinction between first-order and second-order logic is blurred as we argued in **(author?)** (7), as Dependence Logic, as well as IF logic captures concepts that were thought to be expressed only by means of second-order logic.

7 Kit Fine: dependency between arbitrary objects

Finally let us shortly describe another framework which deals with the same problems we have deat with in this paper. It is the framework of arbitrary objects introduced in **(author?)** (3). In this case there are also two kinds of dependency, a (material) dependency at the level of individual objects which is sustained, not by a relation of dependency between quantifiers, as in IF logic, but by a relation of dependency between arbitrary objects. Arbitrary objects are introduced by quantifiers (and are named by constants in the object language) and the dependency relation between them is represented in the object language. Thus when b is an arbitrary object that depends only on the arbitrary object a, then the values assigned to b must be determined by the values assigned to a. Arbitrary objects are divided into independent and dependent ones. Perhaps at this stage an example might help. Consider the natural language discourse fragment:

15. Every farmer owns a donkey. He beats it; he feeds it rarely.

Here 'Every farmer' introduces an independent arbitrary object, and 'a donkey' introduces another arbitrary object which is dependent on the first. The first arbitrary object is associated with a set of individual farmer, and so is the second. Every pair (a, b) of such individual objects stand in the relation of ownership, $a$owns b.The anaphoric pronoun 'He' is a place holder for the name of the first arbitrary object, and 'it' is a placeholder for the name of the second arbitrary object. And so on.

Fine formulates an objection against the use of Skolem functions to encode the material dependencies of individuals which are associated with arbitrary objects. According to it Skolem functions cannot handle multi-dependencies, which arise for instance in the case of 3 arbitrary objects, a, b, and c, such that c depends on b in a particular way, and b depends on a in another way. Keeping in mind that arbitrary objects are introduced by quantifiers, the independent ones by universal quantifiers, and the dependent ones by existential quantifiers, I take Fine's concern (if I correctly understood him) to be essentially about

how to sustain the dependencies encoded by two Skolem functions by the scopal dependencies of three quantifiers, one universal and two existential ones. This is the question we discussed in the previos sections. Fine is right in that there is no way to arrange the three quantifiers to get the dependencies encoded by the two Skolem functions. The solution offered in IF logic is liberate the patterns of dependencies between quantifiers and generalize the notion of Skolem function. In this particular example, we get

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

where $\exists z$ corresponds to Fine's arbitrary object *c*and is only in the scope of $\exists y$ (Fine's arbitrary object b) which depends only on $\forall x$ (Fine's arbitrary object a). We can also achieve the same result using Dependence logic as a framework, adding to the linear sequence of quantifiers $\forall x \exists y \exists z$ the dependence atoms $= (x; y)$ and $= (y; z)$. Notice that the dependencies between the three variables asserted by the two atoms match Fine's dependencies between the three arbitrary objects, a, b and c. It is true that in this case we cannot, for instance, represent (15) in such a way that 'He' is a place holder of "Every farmer" but the gain in ontological parsimony is considerable. I have tackled some of these issues in **(author?)** (9).

References

[1] J. Barwise, On branching quantifiers in English, Journal of Philosophical Logic 8 (1):47 - 80, 1979.

[2] J. Kontinen, A. Kuusisto, P. Lohmann and J. Virtema, Complexity of two-variable dependence logic and IF-logic, Information and Computation, 239, pp.237-253, 2014.

[3] K. Fine and N. Tennant, A defence of Arbitrary Objects, in Proceedings of the Aristotelian Society, Supplementary Volumes Vol. 57 (1983), pp. 55-77+79-89.

[4] L. Henkin, Some remarks on infinitely long formulas, in P. Bernays, editor, Infinitistic Methods: proceedings of the Symposium on Foundations of Mathematics, Warsaw, 2-9 September 1959, pages 167-183, Oxford 1961. Pergamon Press.

[5] J. Hintikka, Quantum Logic as a fragment of Independence-Friendly Logic, Journal of Philosophical Logic, 31, 197-209, 2002.

[6] J. Hintikka and G. Sandu, Informational independence as a semantical phenomenon, in J. E. Fenstad et al, editors, Logic, Methodology and Philosophy of Science, volume 8, 571-589. Elsevier, Amsterdam, 1989.

[7] J.Hintikka and G.Sandu, What is Logic?, in Dale Jacquette (ed.), Philosophy of Logic, North Holland, pp. 13-39, 2006.

[8] J. Hintikka and J. Symons, unpublished manuscript. Game-theoretical semantics as a basis of a general logic.

[9] G. Sandu, Functional anaphora and arbitrary objects, forthcoming in Mircea Dumitru (ed), Metaphysics, Meaning, and Modalities. Themes from Kit Fine, Oxford University Press.

[10] G. Sher, Ways of Branching Quantiers, Linguistics and Philosophy, 13, 393-422, 1990.

[11] J. Väänänen, Depenndence Logic. Cambridge University Press, UK, 2007.