Notre Dame Journal of Formal Logic
Volume 21, Number 4, October 1980

Predicate-Functors and the Limits of Decidability in Logic

ARIS NOAH

1 In this paper,* we use predicate-functor logic to probe the limits of decidability. Quine's assertion that polyadic logic remains decidable as long as there are no permutations or recurrences of predicate places is given precise formulation and tested through consideration of two well-defined subsystems of Quine's full system of predicate-functor logic.

Quine's rationale for developing predicate-functor logic (see [3] and [4]) is that predicate-functors provide a tool for a discriminating analysis of various syntactical functions carried out by the bound variables of quantification. Distinct functions are apportioned to distinct functors, whose combined use can express exactly what can be expressed in the quantifier-variable notation of first-order quantification theory.

Quine's analysis has led him to the conclusion that

> the essential services of the variable are the permutation of predicate places and the linking of predicate places by identity. The permutation job is discharged in our predicate-functor logic by the functors 'p_i', and the linking job by the self functor 'S'. . . . The existential force of quantification, at any rate, is no essential or distinctive service of the variable; it is carried as well by the cropping functor '$\daleth$' and, for that matter, by the Boolean '$\neq \Lambda$'. ([3], p. 304).

---

*I would like to give full credit to Fred Sommers of Brandeis University for setting me on the track which led to the results presented in this paper. Sommers has developed a very interesting algebraic system of logic. Part of his algebra is very closely analogous to what is here called "restricted predicate-functor logic", and it was the study of that system that led me to the isolation and exploration of the restricted system of predicate-functor logic. I would also like to thank W. V. Quine for reading a draft of this paper and suggesting a number of helpful revisions.

Received September 5, 1979; revised December 10, 1979

Quine thinks that his analysis can illuminate the limits of decidability in logic. He asserts that decision procedures cease to be generally available at the precise point where the variable is called upon to perform its two essential functions of permutation and identification of predicate places (i.e., precisely at the point where translation of quantificational schemata into predicate-functor language would essentially involve the functors 'p_i' and 'S') (see [4], p. 282).

Quine's thesis would imply that polyadic logic remains decidable, like monadic logic, as long as there are no permutations or recurrences of predicate places. As evidence for this, Quine presents a set of polyadic schemata which despite their complexity do not involve such permutations or recurrences, and shows that they are decidable. The set *FS* of "fluted" schemata, is defined as follows:

> Every predicate letter has the same variable 'x' as its first argument, though this repeated letter may in its different occurrences be bound by different occurrences of '(x)' or '$(\exists x)$'. Every predicate letter has one and the same letter 'y' as its second argument, if any; and so on. And, a final requirement, each occurrence of a 'y' quantifier stands in the scope of some 'x' quantifier; each occurrence of a 'z' quantifier stands in the scope of some 'y' quantifier; and so on. ([4], p. 282).

The decision procedure for fluted schemata is based on a method worked out by Herbrand for monadic predicate logic and extended by Quine to all fluted schemata, as described in [1]. In what follows, we will show that *FS* corresponds to a system of predicate-functor logic employing only three primitive functors.

2 The *restricted system of predicate-functor logic* (*RPF*), contains only the following three primitive functors: the Boolean complement functor '$-$', the (existential) cropping functor '$\exists$' (we use this symbol to underscore its correspondence to the existential quantifier), and a special restricted form of Quine's Boolean intersection functor, which we shall call *homogeneous conjunction functor* ('$\cdot$'). Whereas Quine's Boolean intersection functor compounds predicates of varying degrees n, m, our homogeneous conjunction functor is defined only for predicates of the same degree.

Translation back and forth from quantification theory to restricted predicate-functor logic is carried out on the basis of the following equivalences:

(1) $(F^n \cdot G^n)x_1 \ldots x_n \equiv F^n x_1 \ldots x_n \cdot G^n x_1 \ldots x_n$
(2) $(-F^n)x_1 \ldots x_n \equiv -F^n x_1 \ldots x_n$
(3) $(\exists F^n)x_1 \ldots x_{n-1} \equiv (\exists x_n)F^n x_1 \ldots x_{n-1}x_n$

(where '$\exists F^n$' represents a predicate of degree $n - 1$). For convenience, we can further define a universal cropping functor corresponding to universal quantification, as follows:

(4) $(\forall F^n)x_1 \ldots x_{n-1} \equiv (\forall x_n)F^n x_1 \ldots x_{n-1}x_n$.

(Obviously, '$\forall F^n$' is equivalent to '$- \exists - F^n$'.) We can also define functors corresponding to sentential alternation and material implication, as follows:

(5) $(F^n \vee G^n)x_1 \ldots x_n \equiv F^n x_1 \ldots x_n \vee G^n x_1 \ldots x_n$

(6) $(F^n \rightarrow G^n)x_1 \ldots x_n \equiv F^n x_1 \ldots x_n \rightarrow G^n x_1 \ldots x_n$.

(Generally, to each truth-functional connective corresponds a predicate-functor. In fact, the connective can be seen as a special case of the more generic predicate-functor, used to join only predicates of null degree, i.e., closed sentences.) The following equivalence concerns the identity predicate[1]:

(7) $Ixy \,.\!\equiv\!.\, x = y$.

3 We will show now that any closed quantificational schema that is "fluted" in Quine's sense is translatable into the language of *RPF*. The method of translation is based on the equivalences of Section 2.

First, we drive all quantifiers as far in as possible. We thus get a schema such as the following:

$$(\exists x)\{-Ax \cdot (Bx \rightarrow (\exists y)[Rxy \;.\!\vee\!.\; (Sxy \cdot -Txy) \;.\!\vee\!.\; -(\exists z)(Fxyz \vee -Gxyz)])\}.$$

(Truth-functions of closed component schemata such as the above are translated by translating each closed component schema according to the method outlined below.)

We begin with the innermost quantifier. Its scope must be a truth-functional combination of open sentence schemata all the predicate letters of which represent predicates of the same degree (in this case, they are both three-place predicates), followed by the same variables (x, y, z) in identical sequence (xyz). Thus, the entire scope can be "homogeneously compounded" into a single complex (three-place) predicate schema, using equivalences (1), (2), (5), and (6) (in this example, (2) and (5)):

$$Fxyz \vee -Gxyz \equiv (F \vee -G)xyz.$$

We now use equivalence (3) (or (4)) to "crop" this predicate down to a two-place predicate, as follows:

$$(\exists z)(F \vee -G)xyz \equiv (\exists(F \vee -G))xy.$$

Furthermore:

$$-(\exists z)(F \vee -G)xyz \equiv (-\exists(F \vee -G))xy.$$

We now consider the scope of the next innermost quantifier, in this case '$(\exists y)$'. It again consists of a truth-functional combination of open sentence schemata (including the complex two-place predicate schema into which the innermost quantifier and its scope have been transformed), which can be "homogeneously compounded" into a single complex predicate schema, thus:

$$\{[Rxy \;.\!\vee\!.\; (Sxy \cdot -Txy)] \vee (-\exists(F \vee -G))xy\} \equiv \{[R \vee (S \cdot -T)] \vee (-\exists(F \vee -G))\}xy.$$

We proceed as follows:

$$(\exists y)\{[R \vee (S \cdot -T)] \vee (-\exists(F \vee -G))\}xy \equiv (\exists\{[R \vee (S \cdot -T)] \vee (-\exists(F \vee -G))\})x.$$

Now we can compound this last predicate in x with the other predicates in x, and then, again, crop it:

$$(\exists x)(-Ax\cdot(Bx \rightarrow (\exists\{[R \vee (S\cdot -T)] \vee (-\exists(F \vee -G))\})x))$$
$$\equiv \exists(-A^1\cdot\{B^1 \rightarrow \exists[(R^2 \vee (S^2\cdot -T^2)) \vee (-\exists(F^3 \vee -G^3))]\}).$$

The result is a schema of *RPF* (the superscripts make explicit the degree of the various predicate letters).

4 Let us now show that every schema of *RPF* is translatable into a fluted schema in *FS*. Take the following schema of *RPF*:

$$\exists\{A^1 \rightarrow [(B^1\cdot\exists(R^2 \vee S^2)) \text{ .v. } \exists(R^2 \rightarrow (\exists F^3 \vee \exists G^3))]\}.$$

(Again, truth-functions of such "closed" predicate-functor schemata can be translated by translating each component "closed" schema according to the method outlined below.) This time we start with the outermost cropping functor and, applying equivalence (3) of Section 2, we get:

$$(\exists x)\{A^1 \rightarrow [(B^1\cdot\exists(R^2 \vee S^2)) \text{ .v. } \exists(R^2 \rightarrow (\exists F^3 \vee \exists G^3))]\}x.$$

Applying (6) to the complex predicate in x, we get:

$$(\exists x)\{A^1x \rightarrow [(B^1\cdot\exists(R^2 \vee S^2)) \text{ .v. } \exists(R^2 \rightarrow (\exists F^3 \vee \exists G^3))]x\}.$$

Applying (5) to the consequent, we get:

$$(\exists x)\{A^1x \rightarrow [(B^1\cdot\exists(R^2 \vee S^2))x \text{ .v. } [\exists(R^2 \rightarrow (\exists F^3 \vee \exists G^3))]x]\}.$$

We proceed to restore x to every predicate letter, by applying (1) to the first component of the alternation:

$$(B^1\cdot\exists(R^2 \vee S^2))x \equiv B^1x\cdot(\exists(R^2 \vee S^2))x.$$

But: $(\exists(R^2 \vee S^2)x \equiv (\exists y)(R^2 \vee S^2)xy$ (by (3))
and: $(R^2 \vee S^2)xy \equiv R^2xy \vee S^2xy$ (by (5)).

Thus, the whole first component of the alternation becomes:

$$B^1x\cdot(\exists y)(R^2xy \vee S^2xy).$$

Applying (3) to the second component of the alternation we get:

$$(\exists(R^2 \rightarrow (\exists F^3 \vee \exists G^3)))x \equiv (\exists y)(R^2 \rightarrow (\exists F^3 \vee \exists G^3))xy.$$

But: $(R^2 \rightarrow (\exists F^3 \vee \exists G^3))xy \equiv R^2xy \rightarrow (\exists F^3 \vee \exists G^3)xy$ (by (6)).

By (5),

$$(\exists F^3 \vee \exists G^3)xy \equiv (\exists F^3)xy \vee (\exists G^3)xy.$$

So far, the second component of the alternation has been translated into:

$$(\exists y)[R^2xy \rightarrow ((\exists F^3)xy \vee (\exists G^3)xy)].$$

Using (3), we get:

$$(\exists F^3)xy \equiv (\exists z)F^3xyz$$
$$(\exists G^3)xy \equiv (\exists z)G^3xyz.$$

Therefore, the second component finally becomes:

$$(\exists y)[R^2xy \rightarrow ((\exists z)F^3xyz \text{ .v. } (\exists z)G^3xyz)].$$

Thus, the original schema has been translated into:

$$(\exists x)\{Ax \rightarrow [(Bx \cdot (\exists y)(Rxy \vee Sxy)) \\ .\vee.\ (\exists y)(Rxy \rightarrow ((\exists z)Fxyz\ .\vee.\ (\exists z)Gxyz))]\},$$

a fluted schema.

5 Predicate-functor analysis of *FS* has revealed that despite the apparent complexity of its schemata, it employs only three primitive functors. They are the same functors used by Quine in his predicate-functor equivalent of monadic logic, which he calls "Boolean algebra of predicates" ([2], pp. 96-110 and [4], p. 280). From the point of view of predicate-functor logic, the syntactical complexity of a system is not a matter of its extralogical list of predicates but a matter of the set of primitive functors it employs. Therefore, *RPF* can be seen as the widest "Boolean" system, including as special cases monadic logic (*RPF* with monadic predicates), propositional logic (*RPF* with null-degree predicates, where the functor '$\exists$' becomes immaterial), and the Boolean elaboration of polyadic logic.

It would be very satisfying at this point to conclude, with Quine, that the limits of decidability coincide with the limits of the Boolean part of logic. Unfortunately there is an intermediate step that separates *RPF* from a clearly undecidable system of predicate-functor logic employing Quine's trouble-making functors 'p_i' and 'S'. This intermediate step, which we can call *minimally extended system of predicate-functor logic* (for short, *MEPF*), employs the Boolean complement functor, the existential cropping functor, and Quine's nonhomogeneous conjunction functor. There are no permutations or recurrences of predicate places in its quantificational equivalent. Yet we do *not* know whether *MEPF* is decidable or not.

If and when we get a definite answer to this question, however, we will be in possession of a precise characterization of the limits of decidability in logic: they will coincide with the limits of either *RPF* or *MEPF*.

6 Failing a conclusive answer to this question, we will briefly point out that the Quine-Herbrand decision procedure is rendered inapplicable beyond the strict confines of *RPF*.

As *RPF* is essentially a polyadic elaboration of monadic logic, so is Quine's decision procedure for "fluted" polyadic schemata an elaboration of Herbrand's method for monadic schemata. Herbrand's method proceeds by putting any monadic quantificational formula into a normal form consisting of an alternation of independent "constituent quantifications". The latter "serve as its mutually independent *truth arguments*, that is arguments that it is a truth function of, just as the pure truth-functional formula has its sentence letters" ([1], p. 59). We can then construct the formula's "Herbrand truth table", which assigns a truth value to the formula for each assignment of truth values to all 2^n constituent quantifications in those n predicate letters, and thus test it for validity. The "Herbrand truth table" of a "fluted" polyadic formula also assigns a truth value to it for each assignment of truth values to its large number of independent constituent quantifications; it is only a matter of exhausting an always finite number of possibilities.

The key to this extended method is, of course, the mutual independence of the formula's constituent quantifications; otherwise they couldn't serve as the formula's independent truth arguments. It is just this crucial requirement that can no longer be safeguarded once we move beyond *RPF*.

The cropping functors, which "crop" predicate schemata into closed sentence schemata or further predicate schemata (of lower degree), are in *RPF* always attached to homogeneous predicate-schemata: this is the essence of the system, safeguarded by its very formation rules (homogeneous conjunction functor).

The effect of introducing the nonhomogeneous conjunction functor is to lose homogeneity. An example of a schema of this slightly enlarged system would be the following:

$$\exists\{A^1 \cdot \exists[A^1 \cdot R^2 \cdot \exists(A^1 \cdot R^2 \cdot F^3)]\}.$$

In *RPF* we could never run into a situation like the above: the homogeneity of the conjunction functor would ensure that A^1 could not possibly recur within the scope of the second cropping functor, and that R^2 could not possibly reappear within the scope of the third cropping functor. But the extension of Quine's decision procedure [1] to all fluted schemata crucially depends on this factor: the inner structure of the scope of a cropping functor *must* be assumed free of occurrences of any predicate letter appearing outside that scope, otherwise the mutual independence of the constituent quantifications cannot be safeguarded.[2] Therefore, Quine's decision procedure is rendered inapplicable beyond the confines of *RPF*.

NOTES

1. For translation of schemata involving constant singular terms, see [3], p. 305.

2. Quine's decision procedure outlined in [1] does not extend to all fluted schemata, but only to a special category of them, which Quine calls "homogeneous" (i.e., the variables must stand in a fixed order after the predicate letters, the quantifiers must be nested always in that order, *and, furthermore, all predicate letters must be of the same degree*). Quine also remarks: "A further proviso was that all the predicate letters have the same number of argument places; but this appears superfluous" ([4], p. 282). The reason that proviso turns out to be superfluous is precisely the nonrecurrence of predicate letters appearing outside the scopes of quantifiers within these very scopes. Given this, Quine's decision procedure can be extended to all fluted schemata. But without this, it is rendered inapplicable.

REFERENCES

[1] Quine, W. V. O., "On the limits of decision," pp. 57-62 in *Proceedings of the 14th International Congress of Philosophy (Vienna, 1968)*, vol. III, University of Vienna, 1969.

[2] Quine, W. V. O., *Methods of Logic*, 3rd Ed., Holt, Rinehart and Winston, New York, 1972.

[3] Quine, W. V. O., "Algebraic logic and predicate functors," pp. 283-307 in *The Ways of Paradox,* revised and enlarged edition, Harvard University Press, Cambridge, Massachusetts, 1976.

[4] Quine, W. V. O., "The variable," pp. 272-282 in *The Ways of Paradox,* revised and enlarged edition, Harvard University Press, Cambridge, Massachusetts, 1976.

Department of Philosophy
Brandeis University
Waltham, Massachusetts 02154