Notre Dame Journal of Formal Logic
Volume 28, Number 3, July 1987

A First-Order Logic With No Logical Constants

CHARLES B. DANIELS*

1 The language LQ The language *LQ* consists of a denumerable set $\{x, y, z, x_1, y_1, \ldots\}$ of (individual) variables, for each $n \geq 1$ a set of n-place predicates, and a denumerable set of atomic wffs. *LQ* has no punctuation marks. The set of wffs of *LQ* is the smallest set S such that

(1) if A is an atomic wff, $A \in S$
(2) if $\langle x_1, \ldots, x_n \rangle$ is a sequence of variables and F is an n-place predicate, $Fx_1 \ldots x_n \in S$, and
(3) if A, $B \in S$ and x is a variable, $xAB \in S$.[1,2]

An occurrence of a variable x in A is free if it is not in a subwff xBC of A. A variable x occurs free in A if there is a free occurrence of x in A. A variable x is free for a variable y in a wff A if no free occurrence of x in A is in a subwff yBC in A. We write Ax/y for the wff that results when the variable y is substituted for the variable x at all free occurrences of x in A.

Finally, throughout this paper, where A is a wff in which a variable x does not occur free, we use the notation $A(x)^*$ and $A(x)^{**}$ in the following way: $A(x)^*$ stands for xAA when $A(x)^{**}$ stands for A, and $A(x)^{**}$ stands for xAA when $A(x)^*$ stands for A.

We now provide natural deduction rules for *LQ*. In almost all natural deduction systems we have the following three rules:

*I wish to thank Bas van Fraassen, Kit Fine, and an anonymous reader for helpful suggestions.

Received November 12, 1986

Stutter

$$\begin{array}{|l} A \\ \\ A \end{array}$$

Importation

$$\begin{array}{|ll} A & \\ & \begin{array}{|l} [y] \\ A \end{array} \end{array}$$

Hypothesis

$$\begin{array}{|l} \underline{A} \\ \end{array}$$

where in Importation y is not free in A. To these we add the following six:

Switch

$$\begin{array}{|l} xAB \\ xBA \end{array}$$

Expansion

$$\begin{array}{|l} A(x)^{*} \\ xA(x)^{**}B \end{array}$$

Combination

$$\begin{array}{|l} A \\ B \\ yxABxAB \end{array}$$

Reductio

$$\begin{array}{|ll} & \begin{array}{|l} \underline{A(x)^{**}} \\ B(y)^{*} \\ B(y)^{**} \end{array} \\ A(x)^{*} & \end{array}$$

Generalization

$$\begin{array}{|ll} & \begin{array}{|l} [y] \\ xAB \end{array} \\ yAB & \end{array}$$

Instantiation

$$\begin{array}{|l} yAB \\ xAy/zBy/z \end{array}$$

where: (1) in Expansion x is not free in B; (2) in Combination neither x nor y is free in A and B; (3) in Generalization x is free in neither A nor B and the subproof has no hypothesis; and (4) in Instantiation y is free for z in A and B and x is free in neither Ay/z nor By/z.

In classical logic, *LQC*, with $\sim$, $\vee$, and $\forall$ we have the following natural deduction rules:

$\sim$-Out

$$\begin{array}{|ll} & \begin{array}{|l} \underline{\sim A} \\ B \\ \sim B \end{array} \\ A & \end{array}$$

$\sim$-In

$$\begin{array}{|ll} & \begin{array}{|l} \underline{A} \\ B \\ \sim B \end{array} \\ \sim A & \end{array}$$

$\vee$-Out

$$\begin{array}{|ll} A \vee B & \\ & \begin{array}{|l} \underline{A} \\ C \end{array} \\ & \begin{array}{|l} \underline{B} \\ C \end{array} \\ C & \end{array}$$

∨-In	∀-Out	∀-In
$\begin{array}{\|l} \\ A \\ \\ A \vee B \end{array}$ (or $B \vee A$)	$\begin{array}{\|l} \\ (\forall y)A \\ \\ Ay/x \end{array}$	$\begin{array}{\|l} \begin{array}{\|l} [y] \\ \\ A \end{array} \\ (\forall y)A \end{array}$

where in ∀-Out y is free for x in A and in ∀-In the subproof has no hypothesis.

Formulas in *LQ* can be defined in *LQC* and vice versa. In *LQ* we have the following definitions:

'$\sim A$' $=_{df}$ 'xAA', where x is not free in A,
'$A \vee B$' $=_{df}$ '$yxAAxBB$', where neither x nor y is free in A or B,
'$(\forall x)A$' $=_{df}$ '$xyAAyAA$', where y is not free in A and x is.

Likewise in *LQC*, we have the definitions:

'xAB' $=_{df}$ '$\sim A \vee \sim B$', where x is free in neither A nor B,
'xAB' $=_{df}$ '$(\forall x)(\sim A \vee \sim B)$', where x is free in either A or B.

These are not proper definitions, to be sure; they are definition *schemes*. Suppose we find '$\sim A$' in a wff in a deduction. Which member of the denumerably infinite set $\{xAA$: where A is a wff and x is a variable not free in $A\}$ are we to take '$\sim A$' to be standing for? The answer is, of course, any member—*provided* that we take it to be standing for the same member throughout a deduction.

Not *all* wffs of *LQC* can be defined in *LQ*. Wffs with vacuous quantifiers like '$(\forall x)(\forall y)(\forall x)A$', or '$(\forall x)A$' where x is not free in A, are not defined in *LQ*. But these formulas can be dropped from *LQC* without loss anyway. Henceforth *LQC* will be considered not to have them.

The question arises of whether the rules of *LQ* can be validated in *LQC*, and vice versa. The answer is Yes. The proof is tedious. I shall just offer two examples from it.

First it must be shown that all the rules of *LQC* can be validated in *LQ*. Here is a proof of *LQC*'s rule ∨-Out in *LQ*. The varialbes x and z are chosen so as not to occur free in A or B, and y so as not to occur free in C.

$$
\begin{array}{r|l}
1. & zxAAxBB \\
2. & \quad \begin{array}{|l} \underline{yCC} \end{array} \\
3. & \quad \begin{array}{|l} \quad \begin{array}{|l} \underline{A} \end{array} \end{array} \\
4. & \quad \begin{array}{|l} \quad \begin{array}{|l} C \end{array} \end{array} \\
5. & \quad \begin{array}{|l} \quad \begin{array}{|l} yCC \end{array} \end{array} \\
6. & \quad \begin{array}{|l} xAA \end{array} \\
7. & \quad \begin{array}{|l} \quad \begin{array}{|l} \underline{B} \end{array} \end{array} \\
8. & \quad \begin{array}{|l} \quad \begin{array}{|l} C \end{array} \end{array} \\
9. & \quad \begin{array}{|l} \quad \begin{array}{|l} yCC \end{array} \end{array} \\
10. & \quad \begin{array}{|l} xBB \end{array} \\
11. & \quad \begin{array}{|l} xzxAAxBBzxAAxBB \end{array} \\
12. & \quad \begin{array}{|l} zxAAxBB \end{array} \\
13. & C
\end{array}
$$

That C follows from A and from B is given in $\vee$-Out, which explains lines 4 and 8. Line 11 follows by Combination. Reductio is used in lines 6, 10, and 13.

Second it must be shown that the rules of LQ can be validated in LQC. As an example, let me prove LQ's Combination rule in LQC:

1.	A			
2.	B			
3.		$\sim A \vee \sim B$		
4.			$\sim A$	
5.				$\sim A \vee \sim B$
6.				A
7.				$\sim A$
8.			$\sim(\sim A \vee \sim B)$	
9.				
10.			$\sim B$	
11.				$\sim A \vee \sim B$
12.				B
13.				$\sim B$
14.			$\sim(\sim A \vee \sim B)$	
15.		$\sim(\sim A \vee \sim B)$		
16.		$\sim A \vee \sim B$		
17.	$\sim(\sim A \vee \sim B)$			
18.	$\sim(\sim A \vee \sim B) \vee \sim(\sim A \vee \sim B)$			

Thus one can claim that for each wff A in LQ for which there is a categorical proof, there is a categorical proof of any translation $T(A)$ of A in LQC, and vice versa. So if A is a theorem of LQ, $T(A)$ is a theorem of LQC, and vice versa.

2 Semantics A model for LQ is a triple $\langle U, At, I\rangle$, where U is a nonempty set, At is a subset of the set of atomic sentences of LQ, and I is a function such that:

(I1) for all variables x, $I(x) \in U$, and
(I2) for all $n \geq 1$ and all n-place predicates F, $I(F) \subseteq U^n$.

Let $x, x_1, \ldots, x_n$ be variables, F be an n-place predicate, p be an atomic wff, A and B be wffs, M $(= \langle U, At, I\rangle)$ be a model, and Mx $(= \langle U, At, I_x\rangle)$ be a model where for all predicates and variables α other than x, $I(\alpha) = I_x(\alpha)$.

(MAt) $M \Vdash p$, if $p \in At$, otherwise $M \nVdash p$
(MPr) $M \Vdash Fx_1 \ldots x_n$ if $\langle I(x_1), \ldots, I(x_n)\rangle \in I(F)$, otherwise $M \nVdash Fx_1 \ldots x_n$
(MSe) $M \Vdash xAB$ if for all Mx $Mx \nVdash A$ or $Mx \nVdash B$, otherwise $M \nVdash xAB$.

$\Vdash A$ if $M \Vdash A$ for all models M.

The semantics of LQC is that of LQ save that LQC has different evaluation rules for complex wffs from the one rule used in LQ. These are:

($M\sim$) $M \Vdash \sim A$ if $M \nVdash A$, otherwise $M \nVdash \sim A$
($M\vee$) $M \Vdash A \vee B$ if $M \Vdash A$ or $M \Vdash B$, otherwise $M \nVdash A \vee B$
($M\forall$) $M \Vdash (\forall x)A$ if for all Mx $Mx \Vdash A$, otherwise $M \nVdash (\forall x)A$.

Lemma *Where the variable x does not occur free in A, in both LQ and LQC $Mx \Vdash A$ iff $(Mx)^* \Vdash A$.*

This can easily be proved by induction on the complexity of A.

As a consequence of this lemma, we have, for instance, that where x does not occur free in A and in B: (1) $Mx \Vdash A$ iff for all $(Mx)^*$ $(Mx)^* \Vdash A$; (2) $(Mx \nVdash A$ or $Mx \nVdash B)$ iff (for all $(Mx)^*[(Mx)^* \nVdash A$ or $(Mx)^* \nVdash B])$, and so on.

Theorem *Where $T(A)$ is a translation of A:* (I) *for all wffs A of LQC and models M, $M \Vdash A$ in LQC iff $M \Vdash T(A)$ in LQ; and* (II) *for all wffs A of LQ and models M, $M \Vdash A$ in LQ iff $M \Vdash T(A)$ in LQC.*

Proof: By induction on the complexity of A.

Base Cases. (1) A is atomic. $T(A) = A$, and the case is immediate; (2) A is of the form $Fx_1 \ldots x_m$. $T(A) = A$, and the case is immediate.

Inductive Hypothesis (IH): The theorem holds for all wffs of complexity less than n. Suppose that A is of complexity n, then:

(Ia) A is $\sim B$ and $T(A)$ is xBB, where x is not free in B, $M \Vdash xBB$ in *LQ* iff for all Mx, $Mx \nVdash B$ or $Mx \nVdash B$ in *LQ* iff for all Mx, $Mx \nVdash B$ in *LQ* iff, by the lemma, $M \nVdash B$ in *LQ* iff, by IH, $M \nVdash B$ in *LQC* iff, by $(M\sim)$, $M \Vdash \sim B$ in *LQC*.

(Ib) A is $B \vee C$ and $T(A)$ is $yxBBxCC$, where neither x nor y is free in B and C. $M \Vdash yxBBxCC$ in *LQ* iff for all My, $My \nVdash xBB$ or $My \nVdash xCC$ in *LQ* iff, by the lemma, $M \nVdash xBB$ or $M \nVdash xCC$ in *LQ* iff for some My, $(My \Vdash B$ and $My \Vdash B)$ or for some My, $(My \Vdash C$ and $My \Vdash C)$ in *LQ* iff for some My, $My \Vdash B$ or for some My, $My \Vdash C$ in *LQ* iff, by the lemma, $M \Vdash B$ or $M \Vdash C$ in *LQ* iff, by IH, $M \Vdash B$ or $M \Vdash C$ in *LQC* iff, by $(M\vee)$, $M \Vdash B \vee C$ in *LQC*.

(Ic) A is $(\forall x)B$ and $T(A)$ is $xyBByBB$, where y is not free in B. $M \Vdash xyBByBB$ in *LQ* iff for all Mx, $Mx \nVdash yBB$ or $Mx \nVdash yBB$ in *LQ* iff for all Mx, $Mx \nVdash yBB$ in *LQ* iff for all Mx there is some Mxy such that $Mxy \Vdash B$ and $Mxy \Vdash B$ in *LQ* iff for all Mx there is some Mxy such that $Mxy \Vdash B$ in *LQ* iff, by the lemma, for all Mx, $Mx \Vdash B$ in *LQ* iff, by IH, for all Mx, $Mx \Vdash B$ in *LQC* iff, by $(M\forall)$, $M \Vdash (\forall x)B$ in *LQC*.

(IIa) A is xBC, x is free in neither B nor C, and $T(A)$ is $\sim B \vee \sim C$. $M \Vdash \sim B \vee \sim C$ in *LQC* iff $M \Vdash \sim B$ or $M \Vdash \sim C$ in *LQC* iff $M \nVdash B$ or $M \nVdash C$ in *LQC* iff, by IH, $M \nVdash B$ or $M \nVdash C$ in *LQ* iff, by the lemma, for all Mx, $Mx \nVdash B$ or $Mx \nVdash C$ in *LQ* iff $M \Vdash xBC$ *in LQ*.

(IIb) A is xBC, x is free in B or C, and $T(A)$ is $(\forall x)(\sim B \vee \sim C)$. $M \Vdash (\forall x)(\sim B \vee \sim C)$ in *LQC* iff for all Mx, $Mx \Vdash \sim B \vee \sim C$ in *LQC* iff for all Mx, $Mx \Vdash \sim B$ or $Mx \Vdash \sim C$ in *LQC* iff for all Mx, $Mx \nVdash B$ or $Mx \nVdash C$ in *LQC* iff, by IH, for all Mx, $Mx \nVdash B$ or $Mx \nVdash C$ in *LQ* iff, by (MSe), $M \Vdash xBC$.

This concludes the proof of the theorem.

Thus *LQ* is sound and complete with respect to its semantics iff *LQC* is sound and complete with respect to its semantics.

Problem: What is the shortest length for a single axiom for the language of *LQ* that together with the rules:

(R1) if $\vdash A$ and $\vdash yAxBC$, then $\vdash C$ (where y is not free in A, B, or C and x in B or C),
(R2) if $\vdash xAB$, then $\vdash yAB$ (where x is free in neither A nor B),

will give in translation all and only the theorems of a complete axiomatization of the language of *LQC*?

NOTES

1. It might be alleged that the title of this paper is false advertising in that the variable x in xAB counts as a logical constant, or at least that a structural feature of xAB counts as a logical constant. In the sense in which the typical description of an artificial language starts by listing the primitive signs out of which it is to be assembled—variables, *constants*, and punctuation marks—the only constants in the language just described are predicate and sentence constants. There are no other constants. In the sense that an axiomatics is given which allows complex wffs of the form xAB to be transformed, or an interpretation is intended for this language which arrives at a valuation of the complex wff xAB in terms of valuations of the wffs A and B of which it is composed, the language does contain something that puts wffs together into a complex wff and requires the complex to be manipulated or interpreted. This something may itself be thought of as a logical constant despite the fact that a particular language contains no sign to represent it. The title, then, is at most *half* misleading.

2. Using the variable as a dyadic sentence connective also works for second-order logic, provided that there are no sentence variables. Sentence variables spoil things, since if '*p*' is a sentence variable, '*pAB*' may turn out to be ambiguous. If A is '*ppp*' and B '*p*', we have '*ppppp*'; but this can be decomposed so that A is '*p*' and B is '*ppp*'. A reader has also pointed out that Quine uses a similar device for predicate-functor logic in [1].

REFERENCE

[1] Quine, W. V. O., "Predicate functors revisited," *The Journal of Symbolic Logic*, vol. 46 (1981), pp. 549–652.

20/3987 Gordon Head Road
Victoria, British Columbia
Canada V8N 3X5