Notre Dame Journal of Formal Logic
Volume XIV, Number 4, October 1973
NDJFAM

TRUTH, FALSEHOOD, AND CONTINGENCY IN FIRST-ORDER PREDICATE CALCULUS

CHARLES G. MORGAN

In this paper it is indicated how the results obtained in [1] may be extended to languages with the syntax of first-order predicate calculus. An additional important result is demonstrated to the effect that there can be no proof procedure for the set of logically contingent expressions. The proof of this latter result depends on the undecidability of the predicate calculus, and hence it does not apply to the sentential calculus. At this time the existence of a proof procedure for the logical contingencies of sentential calculus is an open question.

1. *Preliminaries.* Consider a formal language L with the following symbols:

Predicates: P, P_1, P_2, . . . (of varying degree)
Individual constants: a, a_1, a_2, . . .
Individual variables: x, x_1, x_2, . . .
Sentential connectives: &—"and," v—"or," −—"not"
Punctuation:), and (
Quantifiers: (x)—"for every x," ($\exists$x)—"for some x"

I will assume the standard definitions of "well-formed expression of L," and "atomic expression of L." The meta-symbols E, E_1, E_2, . . . will be used to refer to well-formed expressions of the language. I will presuppose the standard semantical theory of such languages, including the semantical definitions of "logically true" (LT), "logically false" (LF), "logically contingent" (LC), and "logically equivalent" (LE).

Let some axiomatic system PCT for L be given (the results in this paper apply to natural deduction systems as a special case). PCT will have axioms TA_1, TA_2, . . . , TA_n and a set of transformation rules (proof rules) TR_1, TR_2, . . . , TR_m. Let λ be a set of expressions of L, perhaps empty. If there is a proof of expression E from λ in the system PCT, I will write $\lambda \vdash_T E$. I will assume that PCT has the following properties:

Received April 5, 1971

Theorem A: *If E is any expression of L and $\vdash_T E$, then E is LT.*
Theorem B: *PCT is consistent in the sense that there is no expression E of L such that both $\vdash_T E$ and $\vdash_T -E$.*
Theorem C: *The axioms $TA_1, \ldots, TA_n$ are independent. That is, it is not possible to prove any one of the axioms from all of the others.*
Theorem D: *The system PCT is complete in the sense that if E is any expression of L that is LT, then $\vdash_T E$.*

The derivation of an analogous system PCF for LF expressions in L will closely parallel the developments in [1]. We will need the following two functions in the course of the derivation. Let F be defined as follows:

$F(E) = E$, if E is any atomic expression
$F(-E) = -F(E)$, for any expression E
$F(E_1 \vee E_2) = F(E_1) \,\&\, F(E_2)$, for any expressions E_1 and E_2
$F(E_1 \,\&\, E_2) = F(E_1) \vee F(E_2)$, for any expressions E_1 and E_2
$F((x)E) = (\exists x)F(E)$, for any expression E
$F((\exists x)E) = (x)F(E)$, for any expression E

Consider an expression of L, for example '$(x)(P_1x \vee -P_2x)$.' The F transform of this expression is '$(\exists x)(P_1x \,\&\, -P_2x)$.' Let T be defined as follows:

$T(E) = -E$, if E is any atomic expression
$T(-E) = -T(E)$, for any expression E
$T(E_1 \vee E_2) = T(E_1) \vee T(E_2)$, for any expressions E_1 and E_2
$T(E_1 \,\&\, E_2) = T(E_1) \,\&\, T(E_2)$, for any expressions E_1 and E_2
$T((x)E) = (x)T(E)$, for any expression E
$T((\exists x)E) = (\exists x)T(E)$, for any expression E

Once again consider the same example expression used above. The T transform of this expression is '$(x)(-P_1x \vee --P_2x)$.' The function T simply replaces an atomic expression by its negation.

Theorem 1: *Let E be any expression of L. Then E is LT if and only if $T(E)$ is LT. Similarly, E is LF if and only if $T(E)$ is LF. Also, E is LC if and only if $T(E)$ is LC.*

Proof: Let E be an expression of L and let I be an arbitrary interpretation. Consider an arbitrary predicate P_i in E, say of degree n. Let p_{it} be the set of n-tuples of which P_i is true under I, and let p_{if} be the set of n-tuples of which P_i is false under I. Consider an interpretation I' of $T(E)$ as follows: the domain of I' is the same as that of I; any given predicate P_i in $T(E)$ is to be true of the n-tuples in the set p_{if} and false of the n-tuples in the set p_{it}, where P_i is true of those in p_{it} and false of those in p_{if} under I. Clearly $T(E)$ takes the same truth value under I' as E takes under I. Further, every interpretation I has associated with it an interpretation I'. Thus E is true under some I if and only if $T(E)$ is true under I'. Similarly, E is false under some I if and only if $T(E)$ is false under I'. Hence E is LT (LF or LC) if and only if $T(E)$ is LT (LF or LC).

Theorem 2: *For any expression E, $F(E)$ is LE to $-T(E)$.*

Proof: The proof is by induction on the number n of connectives and quantifiers in E. Suppose $n = 0$; then E is an atomic expression. Thus $F(E) = E$, and $T(E) = -E$; clearly E is LE to $--E$. Now, suppose the theorem holds for all n less than some p. We must show that the theorem holds for $n = p$.

Case 1: $E = E_1 \vee E_2$. Then $F(E) = F(E_1)$ & $F(E_2)$. By the induction hypothesis, $F(E_1)$ is LE to $-T(E_1)$, and $F(E_2)$ is LE to $-T(E_2)$. Hence $F(E)$ is LE to $-T(E_1)$ & $-T(E_2)$. But, $-T(E_1)$ & $-T(E_2) LE -(T(E_1) \vee T(E_2)) = -T(E_1 \vee E_2)$. Thus, $F(E)$ is LE to $-T(E)$.

Case 2: $E = E_1$ & E_2. This is similar to Case 1, exchanging '&' and 'v' throughout.

Case 3: $E = -E_1$. Then $F(E) = -F(E_1)$. By the induction hypothesis, $F(E_1)$ is LE to $-T(E_1)$. Hence $F(E)$ is LE to $--T(E_1)$. But $--T(E_1) = -T(-E_1)$. Hence $F(E)$ is LE to $-T(E)$.

Case 4: $E = (x)E_1$. Then $F(E) = (\exists x)F(E_1)$. By the induction hypothesis, $F(E_1)$ is LE to $-T(E_1)$. Further, neither F nor T adds, removes, nor exchanges variables. Hence $F(E)$ is LE to $(\exists x) -T(E_1)$. But $(\exists x) -T(E_1)$ is LE to $-(x)T(E_1)$. Further, $-(x)T(E_1) = -T((x)E_1)$. Hence $F(E)$ is LE to $-T(E)$.

Case 5: $E = (\exists x)E_1$. This is similar to Case 4, exchanging '$(\exists x)$' and '(x)' throughout.

We see that the theorem holds for $n = p$, and thus by induction it holds for all n.

Theorem 3: *For any expression E, E is LT if and only if $F(E)$ is LF; E is LF if and only if $F(E)$ is LT; E is LC if and only if $F(E)$ is LC.*

Proof: By Theorem 1, E is LT (LF or LC) if and only if $T(E)$ is LT (LF or LC). By Theorem 2, $F(E)$ is LE to $-T(E)$. But for any expression E_1, we have the following: (i) E_1 is LT if and only if $-E_1$ is LF, (ii) E_1 is LF if and only if $-E_1$ is LT, and (iii) E_1 is LC if and only if $-E_1$ is LC. Thus E is LT (LF or LC) if and only if $-T(E)$ is LF (LT or LC) if and only if $F(E)$ is LF (LT or LC).

Theorem 4: *For any expression E, $F(F(E)) = E$.*

Proof: The proof is by induction on the number n of connectives and quantifiers in E. Suppose $n = 0$; then E is atomic. But then $F(F(E)) = F(E) = E$. Hence the theorem holds for $n = 0$. Suppose the theorem holds for all n less than some p; we must show that it holds for $n = p$.

Case 1: $E = -E_1$. Then $F(F(E)) = F(F(-E_1)) = F(-F(E_1)) = -F(F(E_1))$. But by the induction hypothesis, $F(F(E_1)) = E_1$. Hence $F(F(E)) = -E_1 = E$.

Case 2: $E = E_1 \vee E_2$. Then $F(F(E)) = F(F(E_1 \vee E_2)) = F(F(E_1)$ & $F(E_2)) = F(F(E_1)) \vee F(F(E_2))$. By induction hypothesis, $F(F(E_1)) = E_1$ and $F(F(E_2)) = E_2$. Thus $F(F(E)) = E_1 \vee E_2 = E$.

Case 3: $E = E_1$ & E_2. This is similar to Case 2, exchanging '&' and 'v' throughout.

Case 4: $E = (x)E_1$. Then $F(F(E)) = F(F((x)E_1)) = F((\exists x)F(E_1)) = (x)F(F(E_1))$. By induction hypothesis, $F(F(E_1)) = E_1$. Hence, $F(F(E)) = (x)E_1 = E$.

Case 5: $E = (\exists x)E_1$. This is similar to Case 4, exchanging '(x)' and '$(\exists x)$' throughout.

We see that the theorem holds for $n = p$, and thus by induction it holds for all n.

2. *The System PCF*. The system *PCF* is an axiomatic system for the logical falsehoods of L. It is derived from the system *PCT* by taking the F transform of the system *PCT* in a way that will now be explained. The axioms of *PCF* are just $F(TA_1)$, $F(TA_2)$, . . . , $F(TA_n)$, and I will denote them by FA_1, . . . , FA_n. In order to obtain the transformation rules of *PCF*, denoted by FR_1, . . . , FR_m, simply take the F transform of every expression mentioned in the rules for *PCT*, being concerned only with changes brought about in the connectives and quantifiers, and ignore 'F' when it is applied to expression letters. For example, a typical rule of conditional proof for *PCT* could be stated as follows:

At any stage of a proof, say line r, any premise E_1 may be introduced, providing that:

(a) the premise is discharged at or before the last line of the proof (say at line n) by writing a line of the form '$-E_1 \vee E_2$' (where E_2 is the n-1st line of the proof) and appealing to lines r and n - 1; and

(b) if the proof proceeds past the line on which the premise was discharged (line n) then no line past the nth may appeal to any of the lines r through n - 1 (where r is the line at which the premise was introduced).

The F transform of this rule would read exactly the same except that in condition (a), the expression '$-E_1 \vee E_2$' would be replaced by '$-E_1 \& E_2$.' A rule of substitution that does not explicitly mention any connectives or quantifiers would remain unchanged. Let λ be a set of expressions of L, perhaps empty. If there is a proof of expression E from λ in the system *PCF*, I will write $\lambda \vdash_f E$. Now, let S_1, S_2, . . . be meta-meta-expressions. The same connectives and quantifiers will be used below for the object language, meta-language, and meta-meta-language expressions, but this should occasion no confusion.

Lemma 1: *An object language expression E_1 has the form specified by a meta-language expression S_1 if and only if $F(E_1)$ has the form specified by the F transform of S_1, where the F transform of S_1 is obtained by taking $F(S_1)$, being concerned only with changes brought about in the connectives and quantifiers, and ignoring 'F' when applied to an expression letter.*

Proof: The proof is by an easy induction on the number n of connectives and quantifiers in S_1. Suppose $n = 0$; then S_1 is simply an expression letter. Thus the F transform of S_1 is just an expression letter. But any syntactically well-formed object language expression has the form specified by a single expression letter of the meta-language. Hence the theorem holds for $n = 0$. Now, suppose the theorem is true for all n less than some p. We must show that the theorem holds for p. We will have five cases as in the proof of Theorem 4. In each case, the theorem follows immediately from the induction hypothesis and the definition of F. Thus the theorem holds for all n.

Theorem of Proof Correspondence: *Let $E_1, \ldots, E_p$ be a series of expressions of L, and let $F(E_1), \ldots, F(E_p)$ be a series of expressions of L obtained from the first by taking the F transform of each expression in that series. Then $E_1, \ldots, E_p$ constitutes a proof of E_p in PCT if and only if $F(E_1), \ldots, F(E_p)$ constitutes a proof of $F(E_p)$ in PCF.*

Proof: Consider an arbitrary step E_i in the series. There are two cases:

Case 1: E_i is an axiom. But E_i is an axiom of *PCT* if and only if $F(E_i)$ is an axiom of *PCF*.

Case 2: E_i follows according to rule TR_j from $E_{i1}, \ldots, E_{it}$. Since the function F changes no variables, constants, or predicates, restrictions on variables, constants, and predicates are not changed by taking the F transform of the rules. For example, if we require for the application of rule TR_j that the variable x should not have been introduced at an earlier stage by rule TR_k, then FR_j will require for its application that the variable x should not have been introduced at an earlier stage by the rule FR_k. Thus E_i satisfies such restrictions with regard to TR_j if and only if $F(E_i)$ satisfies such restrictions with regard to FR_j. Now, by Lemma 1, E_i, $E_{i1}, \ldots, E_{it}$ have the syntactical forms specified by TR_j if and only if $F(E_i), F(E_{i1}), \ldots, F(E_{it})$ have the syntactical forms specified by FR_j. Thus, E_i follows according to rule TR_j if and only if $F(E_i)$ follows according to FR_j.

Thus each step in the E_i series is justified if and only if each step in the $F(E_i)$ series is justified. Thus the E_i series constitutes a proof of E_p in *PCT* if and only if the $F(E_i)$ series constitutes a proof of $F(E_p)$ in *PCF*.

Corollary 1: *$\vdash_T E$ if and only if $\vdash_F F(E)$.*
Corollary 2: *Let λ be a set of expressions, and let $F(\lambda)$ be the set of expressions whose members are the F transforms of the members of λ. Then $\lambda \vdash_T E$ if and only if $F(\lambda) \vdash_F F(E)$.*

Proof: Note that step E_i in the *PCT* proof is a member of λ if and only if $F(E_i)$ is a member of $F(\lambda)$. The proof is then the same as the proof for the Theorem of Proof Correspondence.

It is now possible to prove that the system *PCF* has properties similar to those of *PCT*. The proofs that *PCF* has these properties will all have a certain similarity in that they depend on the fact that *PCT* has the corresponding properties.

Theorem A′: *If E is any expression of L and $\vdash_F E$, then E is LF.*

Proof: Suppose for some E, $\vdash_F E$ and E is not *LF*. Then by Theorem 4, $\vdash_F F(F(E))$. By Corollary 1, $\vdash_T F(E)$. By Theorem 3, $F(E)$ is not *LT*. But this contradicts Theorem A. Hence E must be *LF*.

Theorem B′: *PCF is consistent in the sense that there is no expression E of L such that $\vdash_F E$ and $\vdash_F -E$.*

Proof: Suppose that for some E, $\vdash_F E$ and $\vdash_F -E$. Then $\vdash_F F(F(E))$ and $\vdash_F F(F(-E))$. This latter means that $\vdash_F F(-F(E))$. Thus, $\vdash_T F(E)$ and $\vdash_T -F(E)$. But this contradicts Theorem B. Hence there is no such E.

Theorem C′: *The axioms $FA_1, \ldots, FA_n$ are independent. That is, it is not possible to prove any of the axioms from all of the others.*

Proof: Suppose it were possible to prove FA_i from the other axioms. Then by taking the F transform of the proof we would have a proof of TA_i from the other axioms in PCT. But this would contradict Theorem C. Hence it is not possible to prove FA_i from the other axioms.

Theorem D′: *The system PCF is complete in the sense that if E is any expression of L that is LF, then $\vdash_F E$.*

Proof: Let E be any expression of L that is LF. Then $F(E)$ is LT. But then $\vdash_T F(E)$. Hence $\vdash_F F(F(E))$. But then $\vdash_F E$.

3. *The Impossibility of a System PCC.* It is a well known fact due to Church that there is no decision procedure for first-order predicate calculus. That is, there is no effective procedure such that given any arbitrary expression E, the procedure will tell us in a finite number of steps whether or not E is LT. Using this result, it is possible to prove that there can be no system PCC such that (a) all and only logically contingent expressions are provable in PCC, and (b) given an arbitrary series of expressions of L, there is an effective procedure for determining whether or not the series constitutes a valid proof in PCC. The plan of attack is to show that if there were such a system, then it would be possible to given an effective decision procedure for L. I will assume throughout that systems PCT and PCF are given satisfying conditions (a) and (b) for LT and LF expressions, respectively.

Theorem 5: *There is no system PCC such that* (a) *all and only the logically contingent expressions of L are provable in PCC, and* (b) *given an arbitrary series of expressions of L—$E_1, E_2, \ldots, E_p$—there is an effective procedure for determining whether or not the series constitutes a valid proof in PCC of E_p.*

Proof: Suppose there were such a system. As before, if an expression E is provable in PCC, I will write $\vdash_C E$. Let some Gödel numbering be given for expressions and sequences of expressions in L. Then given any two arbitrary positive integers n_1 and n_2, there is an effective procedure for deciding whether:

(i) n_1 is the number of an expression of L

(ii) n_2 is the number of a sequence of expressions of L, the last member of which, say E, is the expression numbered n_1

(iii) The sequence of expressions of which n_2 is the number is either: (iii.a) a valid proof in PCT of E, (iii.b) a valid proof in PCF of E, or (iii.c) a valid proof in PCC of E.

Let an arbitrary expression E be given, and determine its Gödel number, n_1. Now, no matter what positive integer n_2 we choose, the answer to (i) will be 'yes.' Clearly either $\vdash_T E$, $\vdash_F E$, or $\vdash_C E$. Beginning with 1 and stepping through the positive integers in order, we will eventually find an n_2 such that the answer to (ii) is 'yes' and the answer to one of (iii.a)-(iii.c) is 'yes.' If the answer to (iii.a) is 'yes,' then E is LT; if the answer to (iii.b) is 'yes,' then E is LF; if the answer to (iii.c) is 'yes,' then E is LC. Thus there is an effective decision procedure for L. But this contradicts Church's results. Hence there can be no such system PCC.

There is no way to restrict the proof of the above theorem to sentential calculus, since there is a decision procedure for languages with that syntactical structure.

4. *Further Comments.* There are several problems for research that remain open. I will list only a few here:

1. I still have no proof or disproof of the existence of a system SCC for logical contingencies.

2. The semantical theory of LF type languages has yet to be examined, although it appears to be exactly parallel to the traditional theory.

3. It is possible using the techniques outlined above to derive systems in which various combinations of LT, LF, and LC expressions are theorems. The various properties (especially semantical properties) of such systems should be interesting.

4. Another interesting area of research involves the use of a meta-language for LF expressions to describe other formal languages; at least until now, meta-languages have always been of the LT variety.

5. In a slightly different vein, LF languages have some nice applications to problems in philosophy of science and artificial intelligence, especially in the realm of scientific discovery (see [2]).

REFERENCES

[1] Morgan, Charles G., "Sentential calculus for logical falsehoods," *Notre Dame Journal of Formal Logic*, vol. XIV (1973), pp. 347-353.

[2] Morgan, Charles G., "Some Tentative Steps Toward the Algorithmic Generation of Hypotheses," presented to the Canadian Philosophical Association, 1971.

University of Alberta
Edmonton, Alberta, Canada