Notre Dame Journal of Formal Logic
Volume 31, Number 2, Spring 1990

Post's Functional Completeness Theorem

FRANCIS JEFFRY PELLETIER and NORMAN M. MARTIN*

Abstract The paper provides a new proof, in a style accessible to modern logicians and teachers of elementary logic, of Post's Functional Completeness Theorem. Post's Theorem states the necessary and sufficient conditions for an arbitrary set of (2-valued) truth functional connectives to be expressively complete, that is, to be able to express every (2-valued) truth function or truth table. The theorem is stated in terms of five properties that an arbitrary connective may have, and claims that a set of connectives is expressively complete iff for each of the five properties there is a connective that lacks that property.

Everyone knows the technique whereby, given an arbitrary (2-valued) truth table, one can construct a conjunctive (or disjunctive) normal form formula (using only connectives from $\{\vee,\wedge,\sim\}$) which has exactly that truth table. This proves that the set of connectives $\{\vee,\wedge,\sim\}$ is *functionally complete*: any (2-valued) truth table can be constructed from them. Everyone also knows the definitions of $\wedge$ in terms of $\{\vee,\sim\}$ and of $\vee$ in terms of $\{\wedge,\sim\}$. This shows that $\{\wedge,\sim\}$ and $\{\vee,\sim\}$ are also functionally complete sets of connectives. Everyone also knows that the sheffer stroke functions, $\uparrow$ and $\downarrow$, are each functionally complete. Most everyone knows that $\{\rightarrow,\mathbf{F}\}$ is functionally complete and that $\{\rightarrow,\underline{\vee}\}$ is functionally complete (**F** is the constant-false truth function, $\underline{\vee}$ is "exclusive or"). Some people, having worked through Church ([1], p. 131f.), even know that $\{[\ ],\mathbf{T},\mathbf{F}\}$ is functionally complete ([ ] is the ternary connective of "conditional disjunction": $[p,q,r]$ means "if q, then p else r"). However, what is not generally known is why these things are so. What is it about these particular sets of connectives that makes them functionally complete while (say) $\{\leftrightarrow,\sim\}$ is not functionally complete?

*Thanks are due to Errol Martin, Graham Priest, and an anonymous referee for discussions and help with some of the proofs.

Received November 14, 1988; revised January 3, 1989

The answer to this is due to Post [6], according to most authorities (see, e.g., Church [1], p. 131, note 206, and Mukhopadhyay [5], p. 13). However, the proof, if one there be in this work, is given in a style not easily recognized as such by modern logicians. Part of the reason for this is the baroque notation used—a confusing adaptation of the already confusing notation of Jevons [3]. Another part of the reason has to do with the discursive nature of this work, and the fact that Post seems to be simultaneously pursuing several different topics. As remarked above, few logicians know the result despite the fact that it would be an invaluable aid in teaching elementary formal logic and is quite interesting in its own right. The result is reported and proved in various works in computer switching theory (e.g., Mukhopadhyay [5], p. 13f., repeated in Klir [4], pp. 54–61, and in Friedman and Menon [2], pp. 108–112), but the results have not found their way into the general logic literature. In any case, the proofs in these works generally proceed by consideration of "wiring circuits" and depend on properties of such diagrams. Post's original method is algebraic in nature, describing classes of formulas which can be produced by a given set of generators, and the relations that hold among such classes. We here wish to present a proof of the theorem couched in terminology familiar to modern logicians (and teachers of elementary logic) and which proceeds in a simple manner that all such persons can understand. Some of the names given to various classes of connectives (and the systems of logic generated therefrom) have become accepted (in the switching theory literature), but others have died out. For historical interest we mention Post's usage below. This paper then is a description and proof of Post's Functional Completeness Theorem written with teachers of elementary logic in mind.

Post's theorem is in terms of properties of truth functions, each one of which is such that the proposed set of connectives must contain a connective that lacks that property, if the set is to be functionally complete. We start with a description of these properties. But first, as an aid to one of the later descriptions, we define the notion of a *dummy position* in a truth function. Intuitively it is a position of the truth function which never makes a difference in evaluating a formula. For example, suppose we make up the truth function $(p \cdot q)$ thus:

p	q	$p \cdot q$
T	T	F
T	F	T
F	T	F
F	F	T

Note that this function "really" is just the negation of q—the value of p in no way ever makes a difference, so the first position is "dummy". More formally, if f is a truth function of n variables and

$$f(x_1, \ldots, x_{i-1}, \mathrm{F}, x_{i+1}, \ldots x_n) = f(x_1, \ldots, x_{i-1}, \mathrm{T}, x_{i+1}, \ldots x_n)$$

for *all* the possible values of the other variables, then the ith position is a dummy position for f. That is, it never matters what the value of the ith position is.

Now let us define five classes of possible truth functions.

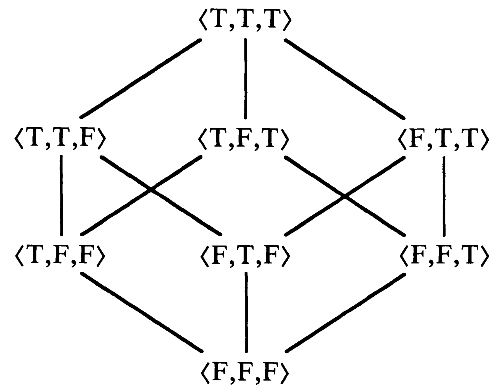
Type 1: Functions closed under T. [Post: "β-functions". The set generated by β-functions is his C_2.] For an arbitrary truth function f, f is closed under T iff $f(T,T,\ldots T) = T$. A function is not closed under T therefore if it has an F in the first row of its truth table (in the standard ordering of rows).

Type 2: Functions closed under F. [Post: "γ-functions". The set generated by γ-functions is his C_3.] For an arbitrary truth function f, f is closed under F iff $f(F,F,\ldots F) = F$. A function is *not* closed under F therefore if it has a T in the last row of its truth table (in the standard ordering of rows).

Type 3: Counting functions. [Post: "Alternating functions". The set of all functions generated by alternating functions is his L_1.] A counting function is one in which every nondummy position always makes a difference. That is, given any row of a truth table, if you ignore values of dummy positions, a change in the value of one argument (holding all others constant) will create a change in the value of the function. So: each position of such a function either never makes a difference ("dummy position") or else it *always* makes a difference. This means that countability can be easily tested in the following way. First, delete dummy positions. Then a function is counting if one of the following two situations occurs: (a) in every row in which the value of the function is T, there are an even number of T's assigned to the arguments of the function, and in every row in which the function is F, there are an odd number of T's assigned to the arguments of the function; or (b) in every row in which the value of the function is T, there are an odd number of T's assigned to the arguments of the function, and in every row in which the function is F, there are an even number of T's assigned to the arguments of the function. That this is an adequate test can be seen by considering a simple example. We ignore any dummy positions. Now suppose $f(T,T,T) = T$; then since a change in an argument must result in a change of the function value, $f(T,T,F) = F$, and applying this fact again we get $f(F,T,F) = T$, and so on. Here everywhere the value of the function is T there are an odd number of T's in the arguments and everywhere the function is F there are an even number of T's. Had the value of $f(T,T,T) = F$, the reverse would have been the case. We call the ones that return T exactly when an even number of arguments are T "even functions"; those that return T when exactly an odd number of arguments are T are called "odd functions". A function f is noncounting iff, after deleting dummy positions, there is at least one n-tuple where $f(x_1,\ldots,T,\ldots,x_n) = f(x_1,\ldots,F,\ldots x_n)$. Also note that since this ith position of the noncounting function is not a dummy position, there is also at least one sequence of truth values $\langle y_1,\ldots,y_n\rangle$ such that $f(y_1,\ldots,T,\ldots,y_n) \neq f(y_1,\ldots,F,\ldots,y_n)$. Note finally that the constant functions **T** and **F** are each counting functions, since they do not have any nondummy variables.

Type 4: Monotonic functions. [Post: "A:a functions". The set generated by A:a functions is his A_1.] A monotonic function is one in which the value of the function "follows" the values of the arguments. That is, if f is an n-adic monotonic function and $\langle x_1,\ldots,x_n\rangle$ and $\langle y_1,\ldots,y_n\rangle$ are sequences of truth values, then: if $\langle x_1,\ldots,x_n\rangle \leq \langle y_1,\ldots,y_n\rangle$ then

$f(x_1,\ldots,x_n) \le f(y_1,\ldots,y_n)$. To make sense of this we note first that we consider $F < T$, and so what $f(x_1,\ldots,x_n) \le f(y_1,\ldots,y_n)$ rules out is that $f(x_1,\ldots,x_n) = T$ while $f(y_1,\ldots,y_n) = F$. A sequence of truth values $\langle x_1,\ldots,x_n\rangle$ is $\le$ to another sequence $\langle y_1,\ldots,y_n\rangle$ just in case whenever $x_i = T$ then $y_i = T$. Pictorially we can represent this as a lattice with the sequence $\langle T,T,\ldots,T\rangle$ at the top and $\langle F,F,\ldots,F\rangle$ at the bottom. For example, the 3-tuple lattice is:



A sequence is $<$ another just in case the former is below the latter along lines in the lattice. A function f is monotonic iff applying it successively to tuples downwards along the lines never results in having a F followed by a T. Or: is nonmonotonic iff there is at least one place where $f(x_1,\ldots,x_n) = T$ while $f(y_1,\ldots,y_n) = F$ and yet $\langle y_1,\ldots,y_n\rangle$ is above $\langle x_1,\ldots,x_n\rangle$ in the lattice.

Type 5: Self-dual functions. [Post: "Self-dual functions". The set generated by self-dual functions is his D_3.] A truth function f is self-dual if its reading from top to bottom is the same as the complement of reading it from bottom to top (in the standard ordering of values for arguments). So, for example, if f yields the values FTFFTTFT (reading from top to bottom of its truth table) we can reverse the order to get TFTTFFTF and complement this to get FTFFTTFT. This is the same as what we started with, and so f is self-dual. The function whose truth table reads FFTFTTFT is *not* self-dual. That is, a self-dual function obeys the following condition: for every row of the truth table $\langle x_1,\ldots,x_n\rangle$, $f(x_1,\ldots,x_n) = -f(-x_1,\ldots,-x_n\rangle$, where $-x_i$ is the opposite truth value from x_i. A function is *not* self-dual just in case there is a row of the truth table $\langle y_1,\ldots,y_n\rangle$ such that $f(y_1,\ldots,y_n) = f(-y_1,\ldots,-y_n)$.

One can easily verify the following statements of what types the usual connectives are:

∧,∨: classes 1,2,4	⊻: classes 2,3
→: class 1	T_k: classes 1,3,4
∼: classes 3,5	F_k: classes 2,3,4
↔: classes 1,3	=: classes 1,2,3,4,5
[ ]: classes 1,2	

T_k is any k-place truth function whose value is always T; F_k is any k-place truth function whose value is always F: [ ] is conditional disjunction; and = is the (monadic) identity function ($=p$ has the value that p has). One might also note that $\uparrow$ and $\downarrow$ do not fall into any of the five classes.

Having defined these classes of truth functions, we are in a position to state:

Theorem (Post's Functional Completeness Theorem) *A set X of truth functions (of 2-valued logic) is functionally complete if and only if, for each of the five defined classes, there is a member of X which does not belong to that class.*

We will prove the theorem by proving it in each direction. First we will prove that if we are given five (2-valued) truth functions which do not fall into the defined categories, we can generate all truth functions. So we start with

$f_1(x_1, \ldots, x_i)$ —which is not T-preserving

$f_2(x_1, \ldots, x_j)$ —which is not F-preserving

$f_3(x_1, \ldots, x_k)$ —which is not monotonic

$f_4(x_1, \ldots, x_m)$ —which is not counting

$f_5(x_1, \ldots, x_n)$ —which is not self-dual.

These functions may be of the same or different adicity, and indeed may even be the same functions, if some function simultaneously is not T-preserving and not self-dual (say).

Lemma 1 *Given f_1, f_2, and f_3 we can define $\sim$.*

Proof: Let us use f_1^*, f_2^*, f_3^* to indicate the truth function just like f_1, f_2, f_3 except that all the arguments x_h are replaced by the same sentence letter.

First note that $f_1^*(p)$ and $f_2^*(p)$ have these partial truth tables, since f_1 and f_2 are not T-preserving and not F-preserving, respectively

p	$f_1^*(p)$	$f_2^*(p)$
T	F	?
F	?	T

If either $f_1^*(\text{F}) = \text{T}$ or $f_2^*(\text{T}) = \text{F}$, then we have negation immediately. So suppose the contrary. In this case $f_1^*(p)$ is the constant **F** and $f_2^*(p)$ is the constant **T**.

Now, since f_3 is nonmonotonic, there must be at least one pair of k-tuples (that is, rows of the function's truth table) $\langle x_1, \ldots, x_k \rangle$ and $\langle y_1, \ldots, y_k \rangle$ where all corresponding values of x_i and y_i agree (are both T or both F) except for one, call it x_j and y_j, and in that case $x_j = \text{T}$ and $y_j = \text{F}$ but $f_3(x_1, \ldots, x_j, \ldots, x_k) = \text{F}$ while $f_3(y_1, \ldots, y_j, \ldots y_k) = \text{T}$. Using these two rows of the original f_3's truth table (there may be more than just the one pair of rows, in which case we just pick one of the pairs), we will construct a new function f_3' using a new variable p. Recall that by assumption we have the constant functions **T** and **F** at our disposal. If in this pair of rows x_i (and hence y_i) both take the value F, then replace that occurrence of x_i by **F**, and if they both take the value T replace it by **T**. In the only place where they differ, replace it by the new variable p. So our new function looks like $f_3'(z_1,\ldots,z_j,\ldots,z_k)$ where all the z's except z_j are either **T** or **F** depending on whether in the rows we were looking at that position was T or F in both. Since there is really only one variable in f_3', let us call this function $f_3^*(p)$. What is the value of f_3^*(T)? Well,

$$f_3^*(\mathrm{T}) = f_3'(z_1,\ldots,\mathrm{T},\ldots,z_k) = f_3(x_1,\ldots,x_j,\ldots,x_k) = \mathrm{F}.$$

And the value of f_3^*(F)?

$$f_3^*(\mathrm{F}) = f_3'(z_1,\ldots,\mathrm{F},\ldots,z_k) = f_3(y_1,\ldots,y_j,\ldots,y_k) = \mathrm{T}.$$

Therefore our defined $f_3^*(p)$ is $\sim p$.

Lemma 2 *Given f_1, f_2, f_3, and f_5, we can define both* **T** *and* **F**.

Proof: Since by Lemma 1 we know we can define $\sim$, all we need to do is to define one of these constants and the other will be definable as its negation. Consider $f_5(x_1,\ldots,x_n)$. Since it is not self-dual we know there is a row of the truth table $\langle z_1,\ldots,z_n\rangle$ such that both

$$f_5(z_1,\ldots,z_n) = y$$

$$f_5(-z_1,\ldots,-z_n) = y$$

(where y is either T or F and $-z_i$ is T iff z_i is F and is F otherwise). Now, in the former row of the truth table (the one without the $-$ marks), if z_i is T replace it by the propositional variable p and if it is F replace it by $\sim p$ (we are guaranteed the existence of $\sim$ by Lemma 1). This results in the (one-place) function $f_5^*(p)$. Note that if p is T, this is just the $f_5(z_1,\ldots,z_n)$ described above which has value y; but since f_5 is not self-dual, this is also $f_5(-z_1,\ldots,-z_n)$ which itself has value y. However $f_5(-z_1,\ldots,-z_n)$ is by definition $f_5^*(p)$ when p is F. So $f_5^*(p) = f_5^*(-p)$, and therefore $f_5^*(p)$ is a one-place constant. Whichever one it is, we generate the other by use of $\sim$.

Lemma 3 *Given f_1, f_2, f_3, f_4, and f_5, we can define a two-place function with an odd number of* T*'s in its truth table.*

Proof: By Lemmas 1 and 2 we can define **T**, **F**, and $\sim$, and so will use them in constructing the relevant two-place function out of f_4. Since f_4 is not counting we know that there are (at least) two n-tuples $\langle z_1,\ldots,z_n\rangle$ and $\langle y_1,\ldots,y_n\rangle$ such that both

$$f_4(z_1,\ldots,z_i,\ldots,z_n) = f_4(z_1,\ldots,-z_i,\ldots,z_n)$$

$$f_4(y_1,\ldots,y_i,\ldots,y_n) \neq f_4(y_1,\ldots,-y_i,\ldots,y_n).$$

The former is known since f_4 is not counting and therefore a change in an argument does not always cause a change in the value of the function; the latter is known because the ith position is not a dummy position and therefore there *is* a pair of rows where a change in the ith argument value causes a change in the function's value.

Let us now look at each of the values in the positions of the two n-tuples $\langle z_1,\ldots,z_i,\ldots,z_n\rangle$ and $\langle y_1,\ldots,y_i,\ldots,y_n\rangle$:

(a) In all cases k except the ith position (the one picked out above):
   (i) If $z_k = y_k = \mathrm{F}$, replace it with **F**.
   (ii) If $z_k = y_k = \mathrm{T}$, replace it with **T**.
   (iii) If $z_k = \mathrm{F}$ and $y_k = \mathrm{T}$, replace it with the variable p.
   (iv) If $z_k = \mathrm{T}$ and $y_k = \mathrm{F}$, replace it with $\sim p$.
(b) In the ith position, replace it with the variable q.

This new function has two propositional variables, p and q. Call it $f_4^*(p,q)$. Note that $f_4^*(\mathrm{F,F}) = f_4(z_1,\ldots,z_{i-1},\mathrm{F},z_{i+1},\ldots,z_n) = f_4(z_1,\ldots,z_{i-1},\mathrm{T},z_{i+1},\ldots,z_n) = f_4^*(\mathrm{F,T})$. But also note that $f_4^*(\mathrm{T,F}) = f_4(y_1,\ldots,y_{i-1},\mathrm{F},y_{i+1},\ldots,y_n)$ which is *not* equal to $f_4(y_1,\ldots,y_{i-1},\mathrm{T},y_{i+1},\ldots,y_n) = f_4^*(\mathrm{T,T})$. Thus we have:

$$f_4^*(\mathrm{T,T}) \neq f_4^*(\mathrm{T,F})$$

$$f_4^*(\mathrm{F,F}) = f_4^*(\mathrm{F,T})$$

and hence there must be an odd number of T's in $f_4^*(p,q)$'s truth table.

Theorem 1 *Given f_1, f_2, f_3, f_4, and f_5, all (2-valued) truth functions can be generated.*

Proof: By Lemma 3 we can generate some two-place truth function with an odd number of T's. It must be one of the following, since these are all there are:

p	q	1	2	3	4	5	6	7	8
T	T	T	T	T	T	F	F	F	F
T	F	T	F	F	T	F	T	T	F
F	T	T	F	T	F	F	T	F	T
F	F	F	F	T	T	T	T	F	F.

Note that columns 1–4 are just the negations of columns 5–8 respectively. Since Lemma 1 guarantees that we have negation, anything we can generate from a member of 1–4 could also have been generated from the corresponding member of 5–8. Therefore we will look only at what is the case assuming that Lemma 3 has given us one of 1–4.

Now, column 1 is just $\vee$, which is well known to be functionally complete in the presence of $\sim$. (And therefore column 5 is also functionally complete in the presence of $\sim$.) Column 2 is just &, which again is obviously functionally complete in the presence of $\sim$. (And therefore column 6 is also functionally complete in the presence of $\sim$.) Column 3 is just $\rightarrow$ which is also functionally complete in the presence of $\sim$. (And hence column 7 is functionally complete in the presence of $\sim$.) The only truth function which is not obviously functionally complete is column 4, $(p \leftarrow q)$, "reverse implication". But using the definition: $(p \vee q) =_{def} (p \leftarrow \sim q)$ we can define $\vee$, which is obviously functionally complete with $\sim$. (And therefore the function in column 8 is also functionally complete in the presence of $\sim$.)

Therefore, no matter which truth function was generated from Lemma 3, we can use the $\sim$ guaranteed by Lemma 1 to form a functionally complete set of connectives.

We now move on to prove the main theorem in the opposite direction: That if we have a functionally complete set of connectives, then for each of the five classes there must be a connective which is not in that class. Our strategy will be to show that each of the five properties "inherits upwards" in the sense that if all available connectives have that property then any compound made up from them must have it also. Having shown this, we present a truth table which has none of the five properties, and conclude that if, for each property, there weren't a connective which didn't have that property, then this truth table could not be described. Therefore, for each of the five properties, there must be a connective which does not have that property.

We prove the "upwards inheritedness" of each of these properties by (strong) induction on the depth of embedding of the number of truth functions within a given truth function. A simple sentence letter has depth of embedding 0; $f(g_1(x_1,\dots,x_i), \dots, g_n(x_1,\dots,x_k))$ has depth of embedding (1 + max(depth of embedding of g_j)). As remarked above, the monadic identity function belongs to each of the five classes; therefore a sentence letter (i.e., a truth function with depth of embedding 0) manifests each of the properties. This will form the basis clause of each induction, and will not be mentioned below. The point is that the five properties were defined by what is true of a truth function's *immediate constituents*, and we would like to show that we could use instead the properties of the individual atomic sentence letters making them up. Thus, for example, a function f being closed under T was defined in terms of whether it generated a T when its immediate constituents were all T. We would like instead to know: If all members of the set X are truth functions closed under T, will every function definable from members of X also be closed under T? The way to answer this is to show that if f is definable using only members of X, then if all atomic sentences are T so is the value of f. Since f can be arbitrarily complex (but finite) we use induction on the depth of embedding of the sentence letters within f.

Lemma 4 *Being closed under* T *is upwards inherited.*

Proof: We suppose we are given a class of truth functions all of whose members are closed under T. For induction, we assume that every function made from this class with depth of embedding $<n$ will yield a T when all sentence parameters in it are assigned T. We prove that any function from this class with depth of embedding n will also yield a T when sentence parameters in it are all T.

Any function with depth of embedding n will look like $f(g_1(x_1,\dots,x_j), \dots, g_m(x_1,\dots,x_k))$, where the greatest depth of embedding of any of the g_i's is $(n-1)$. Therefore, by the inductive hypothesis, each g_i yields T if all the sentence parameters are T. But since f itself is closed under T, then it too must yield T in this circumstance. By induction, then, any truth function made up from functions closed under T will itself be closed under T.

Lemma 5 *Being closed under* F *is upwards inherited.*

Proof: Exactly like that of Lemma 4, replacing T by F.

Lemma 6 *Being monotonic is upwards inherited.*

Proof: We suppose we are given a class of truth functions all of whose members are monotonic. For induction assume that every function g with depth of embedding $<n$ made from this class has the property: if $\langle y_1,\ldots,y_m\rangle$ and $\langle z_1,\ldots,z_m\rangle$ are rows of a truth table (= assignments of values to the sentence parameters), then if $\langle y_1,\ldots,y_m\rangle \leq \langle z_1,\ldots,z_m\rangle$, then $g(y_1,\ldots,y_m) \leq g(z_1,\ldots,z_m)$. We now prove that this holds of all functions from this class with depth of embedding n.

Any function f with depth of embedding n looks like $f(g_1(x_1,\ldots,x_j),\ldots, g_p(x_1,\ldots,x_k))$, where the greatest depth of embedding of any g_i is $(n-1)$. Now, suppose there are q distinct variables in this function, and that we have two rows of a truth table $\langle y_1,\ldots,y_q\rangle$ and $\langle z_1,\ldots,z_q\rangle$ such that $\langle y_1,\ldots,y_q\rangle \leq \langle z_1,\ldots,z_q\rangle$. Since by induction each of the g_i's has the property that $g_i(y_1,\ldots, y_q) \leq g_i(z_1,\ldots,z_q)$, and since f itself is monotonic, it follows that $f(g_1(y_1, \ldots,y_j),\ldots,g_p(y_1,\ldots,y_k)) \leq f(g_1(z_1,\ldots,z_j),\ldots,g_p(z_1,\ldots,z_k))$. So, by induction, any truth function made up from monotonic functions is itself monotonic.

Lemma 7 *Being self-dual is upwards inherited.*

Proof: We assume we are given a class of truth functions all of whose members are self-dual. For induction, assume that every function g with depth of embedding $<n$ made from this class is self-dual, i.e., has the property: $g(-x_1, \ldots,-x_k) = -g(x_1,\ldots,x_k)$, for assignments $x_1\ldots x_k$ of T and F to sentence parameters, where '$-$' applied to T yields F and applied to F yields T. We now prove that all functions from this class with depth of embedding n have the property.

Any function with depth of embedding n made from this class will look like $f(g_1(x_1,\ldots,x_j),\ldots,g_p(x_1,\ldots,x_k))$, where the g_i's maximum depth of embedding is $(n-1)$ and so the induction hypothesis holds for them. Assume that there are q distinct sentence parameters in this function, and consider any arbitrary row of the truth table $\langle y_1,\ldots,y_q\rangle$. Then we get the following pair of equalities:

$$f(g_1(-y_1,\ldots,-y_j),\ldots,g_p(-y_1,\ldots,-y_k))$$
$$= f(-g_1(y_1,\ldots,y_j),\ldots,-g_p(y_1,\ldots,y_k)) \text{ (by induction hypothesis)}$$
$$= -f(g_1(y_1,\ldots,y_j),\ldots,g_p(y_1,\ldots,y_k)) \text{ (f is a self-dual function).}$$

Therefore, the function applied to the negations of the assignments to the sentence parameters is identical to the negation of the function applied to the unnegated assignments to the sentence parameters. And this holds of an arbitrary row of the truth table, and so for all. Thus by induction we conclude that self-duality is upwards inherited.

We would now like to show that being a counting function is upwards inherited. In the strict sense (which is what we will prove) this is true: If $g_1(p_1, \ldots,p_j)$ and $g_2(q_1,\ldots,q_k)$ are counting functions (which means that a change in value of any nondummy variable in them will result in a change in value of the function), and if f is similarly a counting function, then the composition $f(g_1(p_1,\ldots,p_j),g_2(q_1,\ldots,q_k))$ will be a counting function in the sense that any change in value of a nondummy variable in this formula will result in a change in value of the function. But it does not follows that $f(g_1(p_1,\ldots,p_j),g_2(q_1, \ldots,q_k))$ will be a counting function in the sense of having its truth value determined by whether or not there are an even or odd number of T's assigned to $p_1, \ldots, p_j$, $q_1, \ldots, q_k$. For in the process of composition some of these variables might become dummy. Some examples: (a) p and q are themselves counting functions (that is, the identity function on each of p and q is a counting function), and $\leftrightarrow$ is a counting function. Yet $p \leftrightarrow (p \leftrightarrow q)$ is not, in this last sense, a counting function, since when both of p and q are T this formula gives us a T, and when p is F and q is T we also get a T. (b) Similarly, $\sim$ is a counting function, yet $\sim(p \leftrightarrow q) \leftrightarrow (p \leftrightarrow r)$ yields T when $p = r =$ T and $q =$ F, and also when $r =$ T and $p = q =$ F. The problem in these kinds of cases is that p has become a dummy variable in the composed function, even though it is not dummy in the subfunctions which make it up, and even though the positions in the composed function are not dummy positions.

We therefore will take a somewhat circuitous route here, discussing what happens with regard to the "dummy-ness" of variables when a counting function is embedded as an argument in another counting function. We first define what an n-adic counting function is.

Definition $f^n(p_1, \ldots, p_n)$ is an n-adic counting function iff none of positions $1 \ldots n$ are dummy positions, there are no variables in common among $p_1, \ldots, p_n$, and either the value of $f^n(p_1, \ldots, p_n) =$ T exactly when an even number of the p_i's are T (an even n-adic counting function) or the value of $f^n(p_1, \ldots, p_n) =$ F exactly when an even number of the p_i's are T (an odd n-adic counting function).

Sublemma a *For each adicity n there are exactly two counting functions of that adicity: an even one (called* e^n*) and an odd one (called* o^n*).*

Proof: This lemma is obvious, following from the definition of an n-adic counting function. Note that if n is even and if the value of $f(p_1, \ldots, p_n) =$ T when all of $p_1, \ldots, p_n$ are T, then this is e^n. And if the value of $f(p_1, \ldots, p_n) =$ F in this case, then it is o^n. If n is odd, exactly the reverse is true. (Changing one value among $p_1, \ldots, p_n$ must change the function's value; changing yet another value returns the function's value to the original one, etc. Thus there can be only two counting functions of any given adicity.)

Sublemma b *The result of substituting an m-adic counting function into an n-adic counting function at the ith position is identical to one of the following counting functions of adicity* $(n + m - 1)$:

$$\begin{aligned}
&(1)\ \mathrm{e}^n(p_1,\ldots,\mathrm{e}^m(q_1,\ldots,q_m),\ldots,p_n) = \mathrm{o}^{n+m-1}(p_1,\ldots,q_1,\ldots,q_m,\ldots,p_n)\\
&(2)\ \mathrm{e}^n(p_1,\ldots,\mathrm{o}^m(q_1,\ldots,q_m),\ldots,p_n) = \mathrm{e}^{n+m-1}(p_1,\ldots,q_1,\ldots,q_m,\ldots,p_n)\\
&(3)\ \mathrm{o}^n(p_1,\ldots,\mathrm{e}^m(q_1,\ldots,q_m),\ldots,p_n) = \mathrm{e}^{n+m-1}(p_1,\ldots,q_1,\ldots,q_m,\ldots,p_n)\\
&(4)\ \mathrm{o}^n(p_1,\ldots,\mathrm{o}^m(q_1,\ldots,q_m),\ldots,p_n) = \mathrm{o}^{n+m-1}(p_1,\ldots,q_1,\ldots,q_m,\ldots,p_n).
\end{aligned}$$

Proof: We are concerned to show that the two functions on either side of the = sign yield the same output for identical inputs. Since the functions are all counting functions, we need only consider cases where there are an odd or even number of T's among the p_j's and q_i's. Still, this will yield a number of cases. We prove the relevant cases for the first identity. The remaining identities are proved similarly. Recall that by the definition of an n-adic counting function all of the p_j's and q_i's are distinct.

Equation #1:

$$e^n(p_1,\ldots,e^m(q_1,\ldots,q_m),\ldots,p_n) = o^{n+m-1}(p_1,\ldots,q_1,\ldots,q_m,\ldots,p_n).$$

First we note that the adicity of the right hand side (rhs) is correct. There are $(n-1)$ p_j's plus m q_i's.

Case 1: Suppose that an even number of q_i's are T.

a: Suppose that an even number of the p_j's (that is, of all but the replaced p_i) are T. Then, since e^m is an even function, $e^m(q_1,\ldots,q_m) = T$. Since there are an even number of p_j's that are T, this yields an odd number of true arguments for e^n, so the left hand side (lhs) of the equation is F. As for the rhs of the equation: an even number of p_j's are T and an even number of $q_1,\ldots,q_n$ are T, so an even number of the arguments are T; therefore the rhs is F.

b: Suppose an odd number of the p_j's are T. Again the embedded function is T. So now there are an even number of T's to the e^n function and thus the lhs is T. The rhs has an odd number of T's (an even number of q_i's and an odd number of p_j's are T), so the rhs is T.

Case 2: Suppose that an odd number of q_i's are T.

a: Suppose that an even number of the p_j's (that is, of all but the replaced p_i) are T. Then, since e^m is an even function, $e^m(q_1,\ldots,q_m) = F$. Since there are an even number of p_j's that are T, this yields an even number of true arguments for e^n, so the lhs of the equation is T. As for the rhs of the equation: an even number of p_j's are T and an odd number of $q_1,\ldots,q_m$ are T, so an odd number of the arguments are T; therefore the rhs is T.

b: Suppose an odd number of the p_j's are T. Again the embedded function is F. So now there are an odd number of T's to the e^n function and thus the lhs is F. The rhs has an even number of T's (an odd number of q_i's and an odd number of p_j's are T), so the rhs is F.

The other three equations are shown to be correct in the same manner.

Sublemma c *Suppose that $f^n(p_1,\ldots,p_n)$ is a counting function, and that $p_1,\ldots,p_n$ are sentence letters. If an even number, m, of $p_1,\ldots,p_n$ are in fact one and the same sentence letter, then that sentence letter is a dummy variable in $f(p_1,\ldots,p_n)$ and hence $f(p_1,\ldots,p_n)$ has the same input-output pairings as $f^{n-m}(q_1,\ldots,q_{n-m})$.* [$q_1,\ldots,q_{n-m}$ are the remaining sentence letters of $p_1,\ldots,p_n$ after removing the identical p's. Note that of course f^n and f^{n-m} are either both even or both odd.]

Proof:

Case 1: Suppose f^{n-m} is an even function.

a: Suppose an even number of T's are given to f^{n-m} as arguments. Then the output is T. But if the sentence letter that appeared in the m different places of f^n is assigned T, then all occurrences of it in f^n are T. And together with the even number of T's among $q_1,\ldots,q_n$ there are still an even number of T's, so the output of f^n would be T. Had this variable been assigned F, then all occurrences of it in f^n would be F. And therefore there would be an even number of T's as arguments to f^n, namely $n - m$, and so $f^n(p_1, \ldots, p_n)$ would have been T.

b: Suppose an odd number of T's are given to f^{n-m} as arguments. Then the output is F. But if the variable which appeared m times in f^n had been assigned T, then all occurrences of it in f^n would be T. And together with the odd number of T's among $q_1, \ldots, q_{n-m}$ there are still an odd number of T's, so the output of f^n would have been F. If this variable had been assigned F, then all occurrences of it in f^n would be F. And therefore there would have been an odd number of T's as arguments to f^n, and so $f^n(p_1, \ldots, p_n)$ would have been F.

Case 2: Suppose f^{n-m} is an odd function.

a: Suppose an even number of T's are given to f^{n-m} as arguments. Then the output is F. But if the sentence letter that appeared in the m different places of f^n is assigned T, then all occurrences of it in f^n are T. And together with the even number of T's among $q_1, \ldots, q_n$ there are still an even number of T's, so the output of f^n would be F. Had this variable been assigned F, then all occurrences of it in f^n would be F. And therefore there would be an even number of T's as arguments to f^n, namely $n - m$, and so $f^n(p_1, \ldots, p_n)$ would have been F.

b: Suppose an odd number of T's are given to f^{n-m} as arguments. Then the output is T. But if the variable which appeared m times in f^n had been assigned T, then all occurrences of it in f^n would be T. And together with the odd number of T's among $q_1, \ldots, q_{n-m}$ there are still an odd number of T's, so the output of f^n would have been T. If this variable had been assigned F, then all occurrences of it in f^n would be F. And therefore there would have been an odd number of T's as arguments to f^n, and so $f^n(p_1, \ldots, p_n)$ would have been T.

Sublemma 3 *Suppose that $f^n(p_1, \ldots, p_n)$ is a counting function, and that $p_1, \ldots, p_n$ are sentence letters. If an odd number, m, of $p_1, \ldots, p_n$ are in fact one and the same sentence letter, then that sentence letter is not a dummy variable in $f(p_1, \ldots, p_n)$, but this function is equivalent to one in which only one occurrence of this variable appears: $f(p_1, \ldots, p_n)$ has the same input-output pairings as $f^{n-m+1}(q_1, \ldots, q_{n-m+1})$.* [Here $q_1, \ldots, q_{n-m+1}$ are the remaining sentence letters of $p_1, \ldots, p_n$ after removing all but one of the m identical p's. Note that of course f^n and f^{n-m+1} are either both even or both odd.]

Proof: Same as for Sublemma (c), with necessary changes being made for m's being odd.

Lemma 8 *Being counting is upwards inherited.*

Proof: Consider an arbitrary composition of counting functions: $f(g_1, (y_1 \ldots y_j), \ldots, g_p(y_1 \ldots y_k))$. Sublemmas (c) and (d) tell us that the most deeply embedded of these counting functions (those with only sentence letters as arguments) are equivalent to certain other counting functions (depending on how many of the sentence letters are identical). Replace these deeply embedded functions by their equivalents; this does not change the value of f, but none of these new most deeply embedded functions has any dummy variables. Repeated application of Sublemma (b) now says that this new representation of the function is equivalent to a form in which there is no occurrence of these most deeply embedded functions. We can now repeat Sublemmas (c) and (d) to this result (if there are now repetitions of the same sentence letter), and so on. Since the original formula has only a finite depth of embedding, it follows by induction that the original composition of counting functions is equivalent to a counting function applied only to sentence letters. Therefore the original $f(g_1(y_1 \ldots y_j), \ldots, g_p(y_1 \ldots y_k))$ is a counting formula and thus the property is upwards inherited.

Examples Let us consider two simple examples. The formula $p \leftrightarrow (p \leftrightarrow q)$ is composed entirely of counting functions. Now, $\leftrightarrow$ is the even counting function of adicity 2, e^2. Our formula can therefore be represented as $e^2(p, e^2(p,q))$. Neither Sublemma (c) nor Sublemma (d) applies to the embedded function, so we apply Sublemma (b); embedding an even function within an even function. This yields $o^{2+2-1}(p,p,q)$, that is, $o^3(p,p,q)$. Sublemma (c) now applies to this, yielding $o^1(q)$. Since the odd function of adicity 1 is the identity function, this can be represented as q. It can be confirmed that the original formula is equivalent to q. Second, consider $\sim(p \leftrightarrow q) \leftrightarrow (p \leftrightarrow r)$. $\leftrightarrow$ is again e^2, and $\sim$ is e^1; so the formula can be represented as $e^2(e^1(e^2(p,q)), e^2(p,r))$. Sublemmas (c) and (d) do not apply to any of the most deeply embedded formulas, so we will consider $e^2(p,q)$ being embedded within e^1 and apply Sublemma (b). This yields $e^2(o^{1+2-1}(p,q), e^2(p,r))$, i.e., $e^2(o^2(p,q), e^2(p,r))$. Once again we apply Sublemma (b), this time to o^2 embedded within e^2, which yields $e^{2+2-1}(p,q,e^2(p,r))$, i.e., $e^3(p,q,e^2(p,r))$. Now we apply Sublemma (b) to the remaining embedding, yielding $o^{3+2-1}(p,q,p,r)$, i.e., $o^4(p,q,p,r)$. Sublemma (c) says that this is equivalent to $o^2(q,r)$. Since the odd counting function of adicity 2 is "exclusive disjunction", $\underline{\vee}$, this formula is therefore claimed to be equivalent to $(q \underline{\vee} r)$. One can verify that the original formula is indeed equivalent to this.

Theorem 2 *Given a set of connectives X, if one of the five properties (listed above) is manifested by every member of X, then X is not functionally complete.*

Proof: Consider the following truth table:

p	q	$f(p,q)$
T	T	F
T	F	F
F	T	F
F	F	T

(a) Since it has a F in the first row, it is not closed under T. Hence by Lemma 4 this truth function cannot be described by any set of connectives all of whose members are closed under T.
(b) Since it has a T in the last row, it is not closed under F. Hence by Lemma 5 this truth function cannot be described by any set of connectives all of whose members are closed under F.

(c) Since (for example) the first row of the function is F and the last row is T, the function is not monotonic. So by Lemma 6 this function cannot be described by any set of connectives all of whose members are monotonic.
(d) Note that rows 2 and 3 demonstrate that the function is not self-dual. Therefore by Lemma 7 this function cannot be described by any set of connectives all of whose members are self-dual.
(e) Inspection will show that there are no dummy variables in this function; furthermore, the first and second rows have only one change to the truth assignment of the arguments but yet have the same function value. Therefore it is not counting, and hence by Lemma 8 this truth function cannot be described by any formula all of whose connectives are counting functions.

From Theorems 1 and 2, Post's Functional Completeness theorem follows immediately.

REFERENCES

[1] Church, A., *Introduction to Mathematical Logic*, Princeton University Press, Princeton, NJ, 1956.

[2] Friedman, A. and P. Menon, *Theory and Design of Switching Circuits*, Computer Science Press, Woodland Hills, CA, 1975.

[3] Jevons, W., *Pure Logic*, London, 1864.

[4] Klir, G., *Introduction to the Methodology of Switching Circuits*, van Nostrand, New York, 1972.

[5] Mukhopadhay, A., "Complete sets of logic primitives," pp. 1–26 in *Recent Developments in Switching Theory*, edited by A. Mukhopadhay, Academic Press, New York, 1971.

[6] Post, E., *The Two-Valued Iterative Systems of Mathematical Logic*, Annals of Mathematics Series #5, Princeton University Press, Princeton, NJ, 1941.

Francis Jeffry Pelletier
Department of Philosophy
University of Alberta
Edmonton, Alberta
Canada T6G 2E5

Norman M. Martin
Department of Philosophy
University of Texas
Austin, Texas 78712