

# Notes on Post's paper

Randall Holmes

October 7, 2020

## 1 Introduction

I have never read this paper before. I think it has a significant place in the history of the subject (it dates from 1921). It is reputed to contain the first proof of the completeness of propositional logic (in the particular form axiomatized in Principia Mathematica, but carried out in a very general way which can be used for other axiomatizations).

It is manifestly related to Russell (and gives me a chance to keep you out of reading too much of Principia Mathematica, which is a hard read). Some of the terminology should also remind you of Frege to some extent.

The point he makes at the bottom of the first page is an important one. Earlier treatments of symbolic logic informally used in their proofs the very propositions whose formal statements they try to prove. In this paper, Post keeps the sentences of propositional logic that we are reasoning about quite distinct from the language in which we are actually conducting our reasoning (though the latter may have the same basic ideas “or” and “not” appearing in it, it is only the appearances of or and not in the expressions that we are reasoning about that we undertake to prove theorems about).

Post says clearly that the theorems of this paper are not theorems in the propositional logic which is their subject. The assertions of propositional logic are theorems in one sense: the main theorems in this paper are not assertions of propositional logic, but assertions of a mathematical theory with more content (arithmetic, or set theory). We will talk later in this class, at least briefly, about what theory Post might be understood to be working in.

## 2 All the elementary propositions

Post describes the system of Principia Mathematica as having as its subject a certain class of expressions. We supply infinitely many propositional variables  $p_1, p_2, p_3, \dots, q_1, q_2, q_3, \dots, r_1, r_2, r_3, \dots$ . I will assume that these are exactly the ones we have, for concreteness:  $p, q, r$  suffixed with numerals.

He then says that we have two elementary functions  $\sim P$  and  $(P \vee Q)$  which can be used to construct further elementary propositions. I think that we are to view these as functions in a rather Fregean way: what is meant is simply that the letters  $P$  and  $Q$  may freely be replaced with propositions (including complex ones). Parentheses may be dropped where no ambiguity is introduced. So, for example, we are given  $p_1, q_7, r_{17}$  as specific functions. We can construct  $\sim q_7$  as the negation of a previously given expression. We can construct  $\sim q_7 \vee r_{17}$  as the disjunction of two previously given propositions. We can construct  $p_1 \vee (\sim q_7 \vee r_{17})$  as the disjunction of two previously given propositions. In this way, we can construct ever more complex propositions.

I use the letters  $p, q, t_i$  for variables ranging over atomic propositions (the  $p_i$ 's,  $q_i$ 's, and  $r_i$ 's). Post is less careful about drawing such a distinction, which I think is needed. I use capital letters (such as  $P, Q$  in the paragraph above where Post uses  $p, q$ ) as variables standing for any elementary proposition at all, simple or complex. Again, I think Post is less careful about a very important distinction here.

This is a description of the space of all possible elementary propositions.

## 3 The asserted elementary propositions (the theorems of propositional logic)

What really interests us is the space of propositions  $\vdash P$  which are asserted as theorems. Here I am following my own rule, which is that when I introduce a capital letter, this stands for an undetermined complex elementary proposition; if I use  $p$  it varies over  $p_i$ 's,  $q_i$ 's, and  $r_i$ 's.

Post gives a definition of the set of all elementary propositions  $\vdash P$  to be asserted.

He first gives rules for deriving new theorems from old, then he gives the theorems we start with.

**II.** If we have  $\vdash f(p)$  (this standing for any expression containing  $p$ , which

is any variable) then for any variable  $q$  we have  $\vdash f(q)$  and  $\vdash f(\sim q)$ , and for any variables  $q$  and  $r$  we have  $\vdash f(q \vee r)$ .

The effect of this is that we can replace any variable  $p$  with any complex elementary proposition  $A$  throughout an asserted  $\vdash P$  to get another asserted  $\vdash P'$ .

We may in practice need to apply this rule several times to substitute a complex expression  $A$  for  $p$ . Post never actually exhibits this process, but we give an example.

Start with  $\vdash p_1 \vee \sim p_1$  (assuming that we have already derived this).

We can replace  $p_1$  by rule II uniformly with  $q_1 \vee r_1$ .

Thus we have  $\vdash (q_1 \vee r_1) \vee \sim (q_1 \vee r_1)$ .

Now we can replace  $r_1$  with  $\sim r_1$ .

Thus we have  $\vdash (q_1 \vee \sim r_1) \vee \sim (q_1 \vee \sim r_1)$ .

Notice that we replaced  $p_1$  uniformly with  $q_1 \vee \sim r_1$ , which is more complex than the expressions that the text of rule II tell us explicitly can replace a variable. In fact, we can replace a variable with any elementary proposition by repeated applications of the rule, and Post always presumes this.

**III.** If we have  $\vdash P$  and  $\vdash \sim P \vee Q$ , we have also  $\vdash Q$ . This is an implementation of the familiar rule of *modus ponens*, as one can see if one recalls that implication  $P \supset Q$  is defined by Russell as  $\neg P \vee Q$ .

Now is as good a time as any to remark that I am more used to writing  $\neg P$  for negation and  $P \rightarrow Q$  for implication, and if either of these symbols appear they should be understood as meaning the same thing as  $\sim P$  and  $P \supset Q$ , respectively. Moreover, you are welcome to use these symbols if you prefer them or cannot help writing them.

**IV:** Where  $p, q, r$  are any variables, the following are initial assertions (axioms):

1.  $\sim (p \vee p) \vee p$
2.  $\sim q \vee (p \vee q)$
3.  $\sim (p \vee q) \vee (q \vee p)$
4.  $\sim [p \vee (q \vee r)] \vee (q \vee (p \vee r))$

$$5. \sim (\sim q \vee r) \vee (\sim (p \vee q) \vee (p \vee r))$$

I write the axioms in a form using parentheses in a style familiar to you (brackets being an acceptable alternative, and I decided I liked Post's brackets in the fourth axiom), with every needed parenthesis being written except those on the outside. There is no need to put negations in parentheses: parentheses or brackets are written only to signal disjunctions.

I do not use the device of dots which Post inherits from its inventors (the authors of *Principia Mathematica*) but it might be good to have some explanation of them for reading Post's text.  $P.Q$  is used simply to mean what we would write  $P \wedge Q$ , defined as  $\sim (\sim P \vee \sim Q)$ : I will use dots for conjunctions, though I might use  $\wedge$  as well and you are welcome to. A dot or group of dots next to a binary connective is a left or right parenthesis (as appropriate) for which the appropriate right or left parenthesis (respectively) is not written: it will appear as far to the right (or left) as possible with the proviso that one stops before a group of the same number or a larger number of dots, and one must have sensible relationships to any explicit parentheses that are present. It is a tricky device for those not accustomed to it. I'll talk through producing undotted forms of the axioms, and I will be happy to undot any expressions in the paper on request. You are not required to write dots.

The asserted propositions are exactly the ones which can be derived from the axioms by the rules above.

## 4 Truth table methods

Post introduces the truth values  $+$  and  $-$ , and is at pains to say that he is simply using these as formal symbols.

He could have inductively defined the truth table of a function  $f(t_1, \dots, t_n)$  (by which he simply means an expression with distinct atomic propositions  $t_1, \dots, t_n$  in it, and with no other atomic propositions in it; I have noted above that I use  $t_i$ 's as variables ranging over the constant atomic expressions  $p_i, q_i, r_i$ ; Post is not careful about drawing such a distinction). The table will have  $2^n$  rows, each row beginning with one of the  $2^n$  strings of  $n$

$+$ 's and  $-$ 's. Each row will contain one more symbol, which we define by recursion on the structure of  $f(t_1, \dots, t_n)$ :

if  $f(t_1, \dots, t_n)$  is an atomic elementary proposition, it will be one of the  $t_i$ 's, and the last sign in each row will be the same as the  $i$ th sign in the row.

If the function  $f(t_1, \dots, t_n)$  is of the form  $\neg f_1(t_1, \dots, t_n)$ , first construct the truth table of  $f_1(t_1, \dots, t_n)$ : the last sign of each row in the table of  $f$  will be  $-$  if the last sign of the corresponding row in the table for  $f_1$  is  $+$ , and  $+$  if the last sign of the corresponding row in the table for  $f_1$  is  $-$ .

If the function  $f(t_1, \dots, t_n)$  is of the form  $f_1(t_1, \dots, t_n) \vee f_2(t_1, \dots, t_n)$ , then compute the tables for  $f_1$  and  $f_2$ : the last sign in each row of the table for  $f$  will be  $-$  if the last signs of the corresponding rows in the tables for  $f_1$  and  $f_2$  are both  $-$ , and otherwise will be  $+$ .

Actually our definition is more modern than his, but may make it clearer that the truth table of a function  $f(t_1, \dots, t_n)$  can be computed mechanically from the expression  $f(t_1, \dots, t_n)$  without naively reading  $\vee$  and  $\sim$  as the words "or" and "not" of the language we are speaking as we carry out the proof.

I will talk through this nasty-looking formal definition in class, and point out that it actually corresponds precisely to a procedure that students can reasonably use to write down truth tables. In fact, there is clear evidence in the paper that this is how Post was thinking of it, too.

Post then proves a theorem which is important. Every possible truth table is actually the truth table of some proposition.

He proves this by mathematical induction on the order of the truth table (which is one less than the number of symbols in a row: it is the same as the number of variables used).

The four order one tables are handled by functions  $p \vee p$ ,  $\neg p$ ,  $p \vee \sim p$ , and  $\sim (p \vee \sim p)$ , as you can demonstrate by building these truth tables.

He then points out that if we have got functions with each order  $m$  table, we can construct a function with any desired order  $m + 1$  table as follows:

From the order  $m + 1$  table  $T$  you are given, construct two order  $m$  tables  $T^+$  and  $T^-$ .  $T^+$  is obtained by deleting all the rows from the table which have  $-$  in column  $m + 1$ , then deleting column  $m + 1$ .  $T^-$  is obtained by deleting all the rows from the table which have  $+$  in column  $m + 1$ , then deleting column  $m + 1$ . The idea is to replace  $t_{m+1}$  with either a true statement or a false statement. By inductive hypothesis we have  $f_1(t_1, \dots, t_m)$  with table  $T^+$  and  $f_2(t_1, \dots, t_m)$  with table  $T^-$ .

Post of course doesn't use the notations  $T^+$  and  $T^-$ . But I am more

explicitly defining the exact operation he uses.

Post then says that it is easy to see that

$$(t_{m+1}.f_1(t_1, \dots, t_m) \vee (\sim t_{m+1}).f_2(t_1, \dots, t_m))$$

has  $T$  as its table.

One might actually have a better chance of directly verifying Post's last assertion using my formal definition of the truth table of an expression above.

I went through some explicit examples of this procedure in lecture.

I will give a simpler way to produce a function realizing any truth table next time.

## 5 Discussion of Oct 7

### 5.1 A sample proof in Post's system

We derive  $\vdash p \supset p$  in Post's system. Implication is a defined concept: we are actually deriving  $\sim p \vee p$ .

1.  $\vdash \sim q \vee (p \vee q)$  an axiom (IV number 2)
2.  $\vdash \sim [p \vee (q \vee r)] \vee (q \vee (p \vee r))$  an axiom (IV number 4)
3.  $\vdash \sim [\sim q \vee (p \vee q)] \vee (p \vee (\sim q \vee q))$  II applied to line 2: replace  $p$  with  $\sim q$ ,  $q$  with  $p$ ,  $r$  with  $q$ .
4.  $\vdash p \vee (\sim q \vee q)$  III line 1, line 3
5.  $\vdash \sim [\sim q \vee (p \vee q)] \vee (\sim q \vee q)$  II line 4: replace  $p$  with  $\sim [\sim q \vee (p \vee q)]$
6.  $\vdash \sim q \vee q$  III 1,5
7.  $\vdash \sim p \vee p$  II line 6, replace  $q$  with  $p$
8.  $\vdash p \supset p$  line 7, definition of  $\supset$

## 6 Another way to get an assertion with a given truth table

We illustrate by example another general method of getting an expression with a desired truth table.

$p$	$q$	$r$	???
+	+	+	—
+	+	—	+
+	—	+	+
+	—	—	—
—	+	+	—
—	+	—	+
—	—	+	+
—	—	—	—

Above I have given a truth table in the final column of which I have sprinkled random +’s and —’s. There is a uniform way to write down an expression with this truth table, which I now demonstrate using ordinary logical notation.

$(p \wedge q \wedge \neg r)$  from row 2

$\vee(p \wedge \neg q \wedge r)$  from row 3

$\vee(\neg p \wedge q \wedge \neg r)$  from row 6

$\vee(\neg p \wedge \neg q \wedge r)$  from row 7

giving a complete expression  $(p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge r) \vee (\neg p \wedge q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge r)$  with the given table.

This method will work for any table except one with a column of —’s on the right, which can be represented by  $p \wedge \neg p$  (for example).

The expression obtained is in a very stereotyped form, being a disjunction (or statement) linking conjunctions (and statements) composed of atomic letters and negations of letters (such an expression is said to be in disjunctive normal form). In addition, each expression contains all three letters in use in a fixed order.

This method could be used to give an alternative proof of Post’s theorem that there is an expression with any given truth table.

## 7 Boolean algebra

There is a presentation of propositional operations (which is at the same time a presentation of certain basic operations on sets) which supports computations which are involved in the motivation of Post's proof of his main theorem.

You may notice Post referring to disjunction (or) as "logical addition" and conjunction (and) as "logical multiplication". In boolean algebra, we write  $x + y$  for  $x \vee y$  and  $xy$  for  $x \wedge y$  (and  $\bar{x}$  for  $\neg x$ ). The advantage of doing this is that we can present a set of computational rules which suggests a method of taking any expression of propositional logic and converting it to a form which represents its truth table, from which it is easy to see whether it is a theorem or not.

We also use 0 to stand for  $-$  or false, and 1 to stand for  $+$  or true.

Where we say one expression equals another, we say that they have the same truth table.

$$\begin{array}{ll}
 x + y = y + x & xy = yx \\
 (x + y) + z = x + (y + z) & (xy)z = x(yz) \\
 x(y + z) = xy + xz & x + yz = (x + y)(x + z)^* \\
 x + 0 = x & x1 = x \\
 x + 1 = 1^* & x0 = 0 \\
 x + x = x^* & xx = x^* \\
 \overline{x + y} = \bar{x}\bar{y} & \overline{xy} = \bar{x} + \bar{y} \\
 & \overline{\bar{x}} = x \\
 x + \bar{x} = 1 & x\bar{x} = 0
 \end{array}$$

These rules formally resemble algebraic rules for the usual addition and multiplication, though there are some surprises (marked with stars). The negation operation doesn't correspond to any standard arithmetic operation, so we do not star its formulas.

Most of these are common sense if we read addition as intended as "or", multiplication as "and", overstrike as negation, 0 as false and 1 as true.

Something analogous to ordinary algebraic expansion will reduce any expression to a disjunction of conjunctions. The letters can be put in standard order using commutativity and associativity. Repeated letters or negated letters can be shortened to one letter or negated letter ( $xx = x, \bar{x}\bar{x} = \bar{x}$ ). Conjunctions with  $x\bar{x}$  in them will convert to 0 and can be eliminated. If



a conjunction (such as  $xy$ ) is missing a letter (such as  $z$ ) we can add it:  $xy = xy1 = xy(z + \bar{z}) = xyz + xy\bar{z}$ . So algebraic procedures can convert any expression to the form that would be obtained from its truth table by the procedure of the previous section, and this form can be recognized as true if it contains all lines (and actually can then be reduced to 1 by a boolean algebra calculation: for example,  $xy + x\bar{y} + \bar{x}y + \bar{x}\bar{y} = x(y + \bar{y}) + \bar{x}(y + \bar{y}) = x1 + \bar{x}1 = x + \bar{x} = 1$ .)