Company Confidential

PROJECT STRETCH

FILE MEMO # 26

SUBJECT: Logical Connectives
BY: W. Wolensky
DATE: February 29th 1956

With two binary variables, p and q (binary in that they are either 1 or 0) the different combinations of p and q equal 4. The number of possible entries in a truth table are then 2^4 or 16. The full truth table is generated to be:

p	q	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Analyzing the columns in the truth table reveals several things.

1. Columns C_9 through C_{16} are the ones complement of C_1 through C_8.

2. C_1 shows that regardless of the combination of p and q C_1 is always 1 and C_{16} is always 0. These two columns C_1 and C_{16} are trivial, and insignificant because they do not refer to p and q.

3. C_6 is insignificant because it is a function of p only and not a function of p and q, likewise C_4 is a function of q only.

4. C_{11} is insignificant because it is a function of $\bar{p}$ alone, and C_{13} is function of $\bar{q}$.

The truth table is now reduced in size, the remaining 10 columns will be looked at, and analyzed closely to establish their value as logical connectives.

p	q	C_2	C_3	C_5	C_7	C_8	C_9	C_{10}	C_{12}	C_{14}	C_{15}
1	1	1	1	1	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	0	0
1	0	1	0	1	0	0	1	1	0	1	0
0	0	0	1	1	1	0	1	0	0	0	1

Logical connectives can be thought of as operators, which operate on p, q to provide the truth-table of results based upon the type of operator used on p, q.

C_2 represents the operator 'V' or logical inclusive 'OR'. ($p + q$, $p \cdot q$) (eg. $C_2 = 1$ if p or q = 1 or if p and q = 1)

C_8 represents the operator '$\cdot$' or logical 'AND'. ($p \cdot q$)

C_{10} represents the operator '⊻' or logical exclusive 'OR'. $p + q$ ($\overline{p \cdot q}$) ; p or q but not p and q.

C_7 represents the operator '$\equiv$' or "if and only if." $C_7 = 1$ if $p \equiv q$ (this can be considered an identity and an inverse of the logical "exclusive OR").

C_9 represents the operator '|' or "not both", the result takes the value 0 if both p and q are 1. It can also be defined as the inverse operator of the logical 'AND'.

C_{15} represents the operator '$\downarrow$' or "neither-nor", the result takes the value 1 if both variables p and q are 0. This can also be defined as the inverse of the logical inclusive 'OR'.

C_3 represents the operator $p \supset q$ or the logical "if-then". (if $p = 1$, $C_3 = q$ if $p = 0$, $C_3 = 1$).

C_5 represents the operator $q \supset p$

C_{14} represents the operator $p \not\supset q$, the logical "not if-then". (if $p = 1$ then $C_{14} = \overline{q}$ if $p = 0$ then $C_{14} = 0$. This is the inverse of C_3).

C_{12} represents the operator $q \not\supset p$.

C_2, C_7, C_8 have the common characteristic that if the values of p and q are interchanged in the table, the values of C_2, C_7 and C_8 are unaffected, and they are said to be commutative. This is also true for C_{15}, C_{10} and C_9 which are the inverse functions of C_2, C_7 and C_8.

C_3 and C_5 have the common characteristic that if the values of p and q are interchanged in the table, the values of C_3 and C_5 are also interchanged in the table, hence they are not commutative, but are interchange equivalent. C_3 (p,q) = C_5 (q,p). C_{14} and C_{12} are the inverse functions of C_3 and C_5.

A truth table can be constructed for a system containing three variables p, q and r. Such a system would be in effect a binary adder where p is the addend, q is the augend and r is the carry in. The truth table resulting from a system of three variables can be analyzed in much the same manner as the two variables truth table.

The full adder as presently known contains the basics of a ternary system, it is only necessary to control one input (the carry in) of the adder in a prescribed manner to realize a specific logical function of the two uncontrolled inputs.

A normal adder operates as follows:

p = Ad	=	0 0 1 1
q = Au	=	0 1 0 1
r = Ci	=	1 1 1 0
Sum	=	1 0 0 0

p the addend, and q the augend are the variable inputs, in the process of addition the carry in Ci, is generated by the results of the preceding bit addition. If the carry in Ci is controlled according to some pre-determined scheme, the relationships between p and q are specifically unique. The unique logical connective qualities are represented in the sum and carry out patterns resulting.

p = Ad	=	0 1 0 1
q = Au	=	0 0 1 1
Ci	=	1 1 1 1
S	=	1 0 0 1
Co	=	0 1 1 1

The resultant S pattern is $q \equiv p$, and the resultant carry out Co pattern is q V p.

Further use can be realized in the adoption of techniques mentioned by controlling either p or q as well as the carry in. Given a random binary word p, and a specifically designed binary word q, then controlling the carry ins Ci to be zero's the result in the carry out pattern is an extraction of p as defined by q.

p =	1	0	1	1	0	1	1	1	0	0	1	0	1	0	1	1	1
q =	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0
Ci =	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S =	1	0	1	1	1	0	0	0	1	1	0	0	1	0	1	1	1
Co =	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0

The one bits in q define the bits of p which are to be extracted into Co, and can also be defined as the result of a logical And on p q. The pattern resulting in S is an exclusive OR of p q, p ⊻ q.

The purpose of this memorandum is to illustratively define logical connectives, subsequent memos are contemplated which will attempt to define the uses of logical connectives in machine organization.

W.W.

W. Wolensky

Bibliography: IBM Technical Report 001.011.580 - The Multipurpose Bias Device - B.Dunham
Project Beta File Memo #17 - Logical Connectives - W.A. Hunt