## Introduction to Symbolic Logic- the Use of the Truth Table for Determining Validity

Truth tables are useful formal tools for determining validity of arguments because they specify the truth value of every premise in every possible case

Truth tables are constructed of logical symbols used to represent the validity- determining aspects of an argument

**-Symbols**:

**the dot (.)** is used to represent any word that joins two conjuncts (ex. 'and', 'moreover', 'furthermore', 'but', 'yet', 'still', 'however', 'also', 'nevertheless', 'although')

**the wedge (v)** is used to represent any word that joins two disjuncts, most frequently representing the word “or” in an inclusive sense (that is, the inclusive “or” asserts that at least one disjunct is true, while the exclusive “or” asserts that at least one disjunct is true, but not both are true) **the tilde (~)** is used the represent the negation of any simple statement (ex. p= lead is heavy; ~p= lead is not heavy)

**the horseshoe (>)** is used to represent the equivalent of ~(p . ~q); it is used for any conditional statement; for any conditional “if, then” statement to be true, p>q, the negation of the conjunction of its antecedent with the negation of its consequent, must be true also

Examples:

Truth Table for: p . q

| p | q | p . q |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

Truth Table for: p v q

| p | q | p v q |
|---|---|---|
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

Truth Table** for: A > (B v C)
B
Therefore, A > ~C

| A | B | C | B v C | A > (B v C) | ~C | A > ~C |
|---|---|---|---|---|---|---|
| T | T | T | T | T | F | F |
| T | T | F | T | T | T | T |
| T | F | T | T | T | F | T |
| T | F | F | F | F | T | T |
| F | T | T | T | T | F | T |
| F | T | F | T | T | T | T |
| F | F | T | T | T | F | T |
| F | F | F | F | T | T | T |

**This argument is shown to be INVALID by the above truth table because, in the first row, both premises are true, but the conclusion is false.

Either Atlanta wins their conference championship and Baltimore wins their conference championship or Chicago wins the superbowl.

Translation: (A . B) v C

Truth table:

| A | B | C | A . B | (A . B) v C |
|---|---|---|---|---|
| T | T | T | T | T |
| T | T | F | T | T |
| T | F | T | F | T |
| T | F | F | F | F |
| F | T | T | F | T |
| F | T | F | F | F |
| F | F | T | F | T |
| F | F | F | F | F |

The truth of the statement can be assessed, but, because this is not an argument, validity cannot be assessed.

A **truth table** is a mathematical table used in logic—specifically in connection with Boolean algebra, boolean functions, and propositional calculus—which sets out the functional values of logical expressions on each of their functional arguments, that is, for each combination of values taken by their logical variables (Enderton, 2001). In particular, truth tables can be used to show whether a propositional expression is true for all legitimate input values, that is, logically valid.

A truth table has one column for each input variable (for example, P and Q), and one final column showing all of the possible results of the logical operation that the table represents (for example, P XOR Q). Each row of the truth table contains one possible configuration of the input variables (for instance, P=true Q=false), and the result of the operation for those values. See the examples below for further clarification.

There are 4 unary operations:

- Always true
- Never true, unary *falsum*
- Unary *Identity*
- Unary *negation*

The output value is always true, regardless of the input value of p

Logical True

| ***p*** | ***T*** |
|---|---|

| | |
|---|---|
| T | T |
| F | T |

The output value is never true: that is, always false, regardless of the input value of p

Logical False

| *p* | *F* |
|---|---|
| T | F |
| F | F |

Logical identity is an operation on one logical value p, for which the output value remains p.

The truth table for the logical identity operator is as follows:

Logical Identity

| *p* | *p* |
|---|---|
| T | T |
| F | F |

Logical negation is an operation on one logical value, typically the value of a proposition, that produces a value of *true* if its operand is false and a value of *false* if its operand is true.

The truth table for **NOT p** (also written as **¬p**, **Np**, **Fpq**, or **~p**) is as follows:

Logical Negation

| *p* | *¬p* |
|---|---|
| T | F |
| F | T |

Logical conjunction is an operation on two logical values, typically the values of two propositions, that produces a value of *true* if both of its operands are true.

The truth table for **p AND q** (also written as **p ∧ q**, **Kpq**, **p & q**, or **p**  **q**) is as follows:

Logical conjunction

| *p* | *q* | *p* ∧ *q* |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

In ordinary language terms, if both *p* and *q* are true, then the conjunction *p* ∧ *q* is true. For all other assignments of logical values to *p* and to *q* the conjunction *p* ∧ *q*is false.

It can also be said that if *p*, then *p* ∧ *q* is *q*, otherwise *p* ∧ *q* is *p*.

Logical disjunction is an operation on two logical values, typically the values of two propositions, that produces a value of *true* if at least one of its operands is true.

The truth table for **p OR q** (also written as **p ∨ q**, **Apq**, **p || q**, or **p + q**) is as follows:

Logical disjunction

| *p* | *q* | *p* ∨ *q* |
|---|---|---|
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

Stated in English, if *p*, then *p* ∨ *q* is *p*, otherwise *p* ∨ *q* is *q*.

Logical implication and the material conditional are both associated with an operation on two logical values, typically the values of two propositions, which produces a value of *false* if the first operand is true and the second operand is false, and a value of *true* otherwise.

The truth table associated with the logical implication **p implies q** (symbolized as **p ⇒q**, or more rarely **Cpq**) is as follows:

Logical implication

| *p* | *q* | *p* ⇒ *q* |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

The truth table associated with the material conditional **if p then q** (symbolized as **p → q**) is as follows:

Material conditional

| *p* | *q* | *p* → *q* |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

It may also be useful to note that **p ⇒q** and **p → q** are equivalent to **¬p ∨ q**.

Logical equality (also known as biconditional) is an operation on two logical values, typically the values of two propositions, that produces a value of *true* if both operands are false or both operands are true.

The truth table for **p XNOR q** (also written as **p ↔ q**, **Epq**, **p = q**, or **p ≡ q**) is as follows:

Logical equality

| *p* | *q* | *p* ↔ *q* |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | T |

So p EQ q is true if p and q have the same truth value (both true or both false), and false if they have different truth values.

Exclusive disjunction is an operation on two logical values, typically the values of two propositions, that produces a value of *true* if one but not both of its operands is true.

The truth table for **p XOR q** (also written as **p ⊕q**) is as follows:

Exclusive disjunction

| *p* | *q* | *¬pv¬q* |
|---|---|---|
| T | T | F |
| T | F | T |
| F | T | T |
| F | F | F |

For two propositions, **XOR** can also be written as (p ∧ ¬q) ∨ (¬p ∧ q).

The logical NAND is an operation on two logical values, typically the values of two propositions, that produces a value of *false* if both of its operands are true. In other words, it produces a value of *true* if at least one of its operands is false.

The truth table for **p NAND q** (also written as **p ↑ q**, **Dpq**, or **p | q**) is as follows:

Logical NAND

| *p* | *q* | *p ↑ q* |
|---|---|---|
| T | T | F |
| T | F | T |
| F | T | T |
| F | F | T |

It is frequently useful to express a logical operation as a compound operation, that is, as an operation that is built up or composed from other operations. Many such compositions are possible, depending on the operations that are taken as basic or "primitive" and the operations that are taken as composite or "derivative".

In the case of logical NAND, it is clearly expressible as a compound of NOT and AND.

The negation of a conjunction: $\neg(p \land q)$, and the disjunction of negations: $(\neg p) \lor (\neg q)$ can be tabulated as follows:

| *p* | *q* | *p ∧ q* | *¬(p ∧ q)* | *¬p* | *¬q* | *(¬p) ∨ (¬q)* |
|---|---|---|---|---|---|---|
| T | T | T | F | F | F | F |
| T | F | F | T | F | T | T |
| F | T | F | T | T | F | T |
| F | F | F | T | T | T | T |

The logical NOR is an operation on two logical values, typically the values of two propositions, that produces a value of *true* if both of its operands are false. In other words, it produces a value of *false* if at least one of its operands is true. ↓ is also known as the Peirce arrow after its inventor, Charles Sanders Peirce, and is a Sole sufficient operator.

The truth table for **p NOR q** (also written as **p ↓ q**, or **Xpq**) is as follows:

Logical NOR

| *p* | *q* | *p ↓ q* |
|---|---|---|
| T | T | F |

| | | |
|---|---|---|
| T | F | F |
| F | T | F |
| F | F | T |

The negation of a disjunction $\neg(p \lor q)$, and the conjunction of negations $(\neg p) \land (\neg q)$ can be tabulated as follows:

| ***p*** | ***q*** | ***p* ∨ *q*** | **¬(*p* ∨ *q*)** | **¬*p*** | **¬*q*** | **(¬*p*) ∧ (¬*q*)** |
|---|---|---|---|---|---|---|
| T | T | T | F | F | F | F |
| T | F | T | F | F | T | F |
| F | T | T | F | T | F | F |
| F | F | F | T | T | T | T |

Inspection of the tabular derivations for NAND and NOR, under each assignment of logical values to the functional arguments $p$ and $q$, produces the identical patterns of functional values for $\neg(p \land q)$ as for $(\neg p) \lor (\neg q)$, and for $\neg(p \lor q)$ as for $(\neg p) \land (\neg q)$. Thus the first and second expressions in each pair are logically equivalent, and may be substituted for each other in all contexts that pertain solely to their logical values.

This equivalence is one of De Morgan's laws.