**Harold’s Logic Cheat Sheet**

26 October 2022

**The 7 Basic Logical Symbols**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Symbol** | **Example** | **English** |
| **1) Intersection** | ∧, **∧**, ∧, ⋀, **∧** | p ∧ q | * Conjunction
* p and q
* p, but q
* despite the fact that p, q
* although p, q
* overlap
 |
| **2) Union** | ∨, **∨**, ∨, ⋁, **∨** | p ∨ q | * Disjunction
* p or q
* inclusive or
* both combined
 |
| **3) Negation** | ¬, ￢ | ¬p | * not p
 |
| **4) Conditional** | **→, →**,→, ⟶, ⇒, ⟹ | p → q | * if p then q
* if p, q
* q if p
* p implies q
* p only if q
* q in case that p
* p is sufficient for q
* q is necessary for p
 |
| **5) Biconditional** | **↔,** ⟷, **↔**, ⇔, ⟺ | p ⟷q | * p iff q
* p if and only if q
* p is necessary and sufficient for q
* if p then q, and conversely
 |
| **6) Universal Quantifier** | ∀*x* | ∀*x p(x)* | * for all
* for any
* for each
 |
| **7) Existential Quantifier** | ∃*x* | ∃*x p(x)* | * there exists
* there is at least one
 |
| **Equivalence** | ≡ | expression1 ≡ expression2 | * is identical to
* is equivalent to
* the two expressions always have the same truth value
 |
| * The structure of all mathematical statements can be understood using these symbols.
* All mathematical reasoning can be analyzed in terms of the proper use of these symbols.
 |

**Logical Connective Laws**

|  |  |  |
| --- | --- | --- |
| **Law** | **Union Example** | **Intersection Example** |
| **Identity Laws** | p ∨ F ≡ p | p ∧ T ≡ p |
| **Domination or Null Laws** | p ∨ T ≡ T | p ∧ F ≡ F |
| **Idempotent Laws** | p ∨ p ≡ p | p ∧ p ≡ p |
| **Double Negations or****Involution Law** | ¬ ¬p ≡ p |
| **Complement or Complementary Laws** | p ∨ ¬p ≡ T¬F ≡ T | p ∧ ¬p ≡ F¬T ≡ F |
| **Commutative Laws** | p ∨ q ≡ q ∨ p | p ∧ q ≡ q ∧ p |
| **Associative Laws** | (p ∨ q) ∨ r ≡ p ∨ (q ∨ r) | (p ∧ q) ∧ r ≡ p ∧ (q ∧ r) |
| **Distributive Laws** | p ∧ (q ∨ r) ≡ (p ∧ q) ∨ (p ∧ r) | p ∨ (q ∧ r) ≡ (p ∨ q) ∧ (p ∨ r) |
| **Uniting Laws** | (p ∧ q) ∨ (p ∧ ¬q) ≡ p | (p ∨ q) ∧ (p ∨ ¬q) ≡ p |
| **Absorption Laws** | p ∨ (p ∧ q) ≡ p | p ∧ (p ∨ q) ≡ p |
| **De Morgan’s Law (Propositional Logic)** | p ∨ q ≡ ¬(¬p ∧ ¬q)¬(p ∨ q) ≡ ¬p ∧ ¬q(p ∨ ¬q) → r ≡ ¬r → (¬p ∧ q) | p ∧ q ≡ ¬(¬p ∨ ¬q)¬(p ∧ q) ≡ ¬p ∨ ¬q |
| **Multiplying and Factoring Laws** | (p ∨ q) ∧ (¬p ∨ r) ≡ (p ∧ r) ∨ (¬p ∧ q) | (p ∧ q) ∨ (¬p ∧ r) ≡ (p ∨ r) ∧ (¬p ∨ q) |
| **Consensus Laws** | (p ∧ q) ∨ (q ∧ r) ∨ (¬p ∧ r) ≡ (p ∧ q) ∨ (¬p ∧ r) | (p ∨ q) ∧ (q ∨ r) ∧ (¬p ∨ r) ≡ (p ∨ q) ∧ (¬p ∨ r) |
| **Tautology Laws (**⊤**)** | p ∨ (⊤) ≡ ⊤p ∨ ¬p ≡ ⊤ (True) | p ∧ (⊤) ≡ p |
| ¬(⊤) = ⊥ |
| **Contradiction Laws (**⊥**)** | p ∨ (⊥) ≡ p | p ∧ (⊥) ≡ ⊥p ∧ ¬p ≡ ⊥ (False) |
| ¬(⊥) ≡ ⊤ |

**Logical Conditional Connective Laws**

|  |  |  |  |
| --- | --- | --- | --- |
| **Law or Statement** | **Logical Expression** | **Is Equivalent To****(≡)** | **Description** |
| **Conditional Laws** | p ⟶ q | ¬p ∨ q¬(p ∧ ¬q) | Conditional, If ... Then, Implication |
| **Biconditional Laws** | p ⟷q | (p ⟶ q) ∧ (q ⟶ p)(p ⟶ q) ∧ (¬p ⟶ ¬q)(p ∧ q) ∨ (¬p ∧ ¬q) | Bi-conditional, If and only If, iff, XNOR |
| **Sufficient Condition** | p is a sufficient condition for q | The truth of p suffices to guarantee the truth of q. |
| **Necessary Condition** | q is a necessary condition for p | In order for p to be true, it is necessary for q to be true also.¬q ⟶¬p |
| **Equivalence** | p ⟷q | p ≡ qp ⟹ q | Is logically equivalent to (p ≡ ¬ ¬ p)Is equivalent to |
| **Contrapositive** | p ⟶ q | ≡ ¬q ⟶¬p | True |
| **Converse\*** | p ⟶ q | ≢ q ⟶ p | False |
| **Inverse\*** | p ⟶ q | ≢ ¬p ⟶ ¬q | False |

**Logical Predicates**

|  |  |  |  |
| --- | --- | --- | --- |
| **Definition** | **Logical Expression** | **Is Equivalent To (≡)** | **Description** |
| **Universe of Discourse** | ***U*** | All possible inputs in a given range | * Universe of Discourse
* Universal Set
* Universe
 |
| **Domain of Discourse** | 𝔻 | All possible inputs in a given range | * Domain of Discourse
* Universe of Discourse
 |
| **Proposition or Logical Statement** | *p: “Roxy is a mammal”* | *p* | * Must be True or False
* Cannot be a question
* Cannot be a command
 |
| **Predicate** | *P(x): “x is a mammal”* | *P(x)* | * A logical statement whose truth value is a function of one or more variables
* Truth depends upon the input variable *x*
* P(x) ≠ a number
* P(5) is a proposition
 |
| **Example Statements** | *q:* ∀*x* ∈𝔻*, P(x)*: “x is a mammal” | “*For all x in the domain of discourse, P(x) is a mammal.*” | * Is either True or False
* A quantified predicate turns it into a logical statement
 |
| *T(x, y)* | “*x* is a twin of *y*.” | Predicate with two input variables |
| **Truth Set**(Single Free Variable) | *T = P(x)* | *T* = {a | *P(a)*}*T* = {a ∈ *A* | *P(a)*}a ∈ T | The set of all values of x that make the statement *p(x)* true |
| Example: | *P*(x1), *P*(x2), and *P*(x3) are True |
| **Truth Set**(Ordered Pair) | *T = P(x, y)* | {*(a, b)* ∈ *A* × *B* | *P(a, b)*}(a, b) ∈ T | Cross product truth set |
| Examples: | {(p, n) ∈ P × ℕ | the person p has n children} = {(John, 2), …}{(p, c, n) ∈ P × C × ℕ | the person p has lived in the city c for n years} |

**Logical Quantifiers**

|  |  |  |  |
| --- | --- | --- | --- |
| **Definition** | **Logical Expression** | **Is Equivalent To (≡)** | **English** |
| **Universal Quantifier** | ∀*x P(x)*∀*x* ∈ *P(x)*∀*x* ∈ 𝔻*, P(x)*∀*x, if x is in* 𝔻 *then P(x)* | “For all *x* in the domain, *P(x)* is true”∀*x* ∈ *A P(x)* ≡∀*x (x* ∈ *A* ⟶ *P(x))*For the finite set domain of discourse {a1, a2, …, ak}, ∀*x P(x)* ≡ *P(a1)* ∧ *P(a2)* ∧ *…* ∧ *P(ak)* | * for all
* all elements
* for each member
* any
* every
* everyone
* everybody
* everything
* x could be anything at all
 |
| **Existential Quantifier** | ∃*x P(x)*∃*x* ∈ *P(x)*∃*x* ∈𝔻*, P(x)* | “There exists x in the domain, such that *P(x)* is true”For the finite set domain of discourse {a1, a2, …, ak}, ∃*x P(x)* ≡ *P(a1)* ∨ *P(a2)* ∨ *…* ∨ *P(ak)* | * there exists an x
* there is
* some
* someone
* somebody
* at least one value of x
* there is at least one x
* it is the case that
* the truth set is not equal to ∅
 |
| **Uniqueness Quantifier** | ∃!*x P(x)* | there is a unique x in *P(x)* such that …∃*x* (*P(x)* ∧ ¬ *y (P(y)* ∧ *y* ≠ *x))*∃*x* (*P(x)* ∧ ∀*y (P(y)* → *y* = *x))*​∃*x* ∀*y (P(y)* ↔ *y* = *x*)​∃*x* *P(x)* ∧ ∀*y* ∀*z((P(y)* ∧ *P(z))* → *y* = *z*) | * unique
* there is a unique x
* there exists exactly one
* there is exactly one x such that *P(x)*
 |
| **Negated Existential Quantifier** | ¬ [∃*x P(x)*] | ∀*x* ¬*P(x)* | * nobody
* no one
* not one
* there does not exist
 |
| ¬ [∀*x P(x)*] | ∃*x* ¬*P(x)* |
| **Order of Precedence** | PEMDAS for Logic :1. Parenthesis ()
2. Logical NOT (¬)
3. Quantifiers (∀*,* ∃)
4. Logical AND (∧)
5. Logical OR (∨)
6. Logical Conditional (→)
7. Logical Biconditional (⟷)
 | Applied Left to RightExample :∀x *P(x)* ∧ *Q(x)* ≡(∀x *P(x)*) ∧ *Q(x)* |

**Quantifier Laws**

|  |  |  |  |
| --- | --- | --- | --- |
|  **Definition** | **Logical Expression** | **Is Equivalent To (≡)** | **Description / Example / • English** |
| **Abbreviation** | ∃x (x ∈ A ∧ ¬*P(x)*) | ∃x ∈ A ¬*P(x)* | Simplification |
| **Expanding Abbreviation** | ∀x ∈ A *P(x)* | ∀x (x ∈ A → *P(x)*) | Complication |
| **Quantifier Negation Laws** | ∀x ¬*P(x)* | ¬∃x *P(x)* | * nobody’s perfect
 |
| ¬∀x *P(x)* | ∃x ¬*P(x)* | * not everyone is perfect
* someone is imperfect
 |
| **Conditional Law** | x ∈ A ⟶ *P(x)* | x ∉ A ∨ *P(x)* | p ⟶ q ≡ ¬p ∨ q |
| **Subset Negation Law** | x ∈ A | ¬(x ∉ A) | Swap ∈ with ∉, or vice versa |
| **De Morgan’s Law (Quantifier Negation)** | ¬∀*x P(x)* ≡ ∃*x* ¬*P(x)*¬∃*x P(x)* ≡∀*x* ¬*P(x)*¬∀x ∀y *P(x, y)* ≡ ∃*x* ∃*y* ¬*P(x, y)*¬∀x ∃y *P(x, y)* ≡ ∃*x* ∀*y* ¬*P(x, y)*¬∃x ∀y *P(x, y)* ≡ ∀*x* ∃*y* ¬*P(x, y)*¬∃x ∃y *P(x, y)* ≡ ∀*x* ∀*y* ¬*P(x, y)* | De Morgan’s Law for single and nested quantifiers |
| **Nested / Multiple- Quantified Statements** | ∀x ∀y | ∀y ∀x | * for all objects x and y, …
 |
| ∃x ∃y | ∃y ∃x | * there are objects x and y such that …
 |
| ∀x ∃y *P(x, y)* ≢ ∃x ∀y *P(x, y)* | FalseCounterexample for x, y ∈ ℤ: ∀x ∃y (*x + y = 0*) ≡ True∃x ∀y (*x + y = 0*) ≡ False |
| ¬(∀x ∃y *P(x, y)*) | ∃x ∀y ¬*P(x, y)* | Negation of multiply-quantified statements |
| ¬(∃x ∀y *P(x, y)*) | ∀x ∃y ¬*P(x, y)* |
| **Moving Quantifiers** | ∀x (*P(x)* → ∃y *Q(x, y)*) ≡∀x ∃y (*P(x)* → *Q(x, y)*) | You can move a quantifier left if variable is not used yet |

**Quantifier Logic Examples**

|  |  |  |
| --- | --- | --- |
| **Action** | **Logical Statement** | **English** |
| **Everyone** | ∀x ∀y *P(x, y)* NOTE: includes (x = y) | * everyone <did something> to everyone
 |
| **Everyone Else** | ∀x ∀y (*x* ≠ *y*) → *P(x, y)* NOTE: excludes (x = y) | * everyone <did something> to everyone else
 |
| **Someone Else** | ∀x ∃y ((*x* ≠ *y*) ∧ *P(x, y)*) NOTE: excludes (x = y) | * everyone <did something> to someone else
 |
| **Exactly One** | ∃x (*P(x)* ∧ ∀y ((*x* ≠ *y*) → ¬*P(y)*)) ≡ ∃!*x P(x)* | * exactly one person <did something>
 |
| **No One** | ¬∃x (*P(x)* | * no one <did something>
 |

**Valid Quantifier Formulas**

|  |  |  |
| --- | --- | --- |
| **A** |  | **B** |
| ∀x (*P(x)* ∧ *Q(x)*) | ≡ | (∀x *P(x)* ∧ ∀x *Q(x)*) |
| ∃x (*P(x)* ∧ *Q(x)*) | → | (∃x *P(x)* ∧ ∃x *Q(x)*) |
| ∀x (*P(x)* ∨ *Q(x)*) | ← | (∀x *P(x)* ∨ ∀x *Q(x)*) |
| ∃x (*P(x)* ∨ *Q(x)*) | ≡ | (∃x *P(x)* ∨ ∃x *Q(x)*) |
| ∀x (*P(x)* → *Q(x)*) | ← | (∃x *P(x)* → ∀x *Q(x)*) |
| ∃x (*P(x)* → *Q(x)*) | ≡ | (∀x *P(x)* → ∃x *Q(x)*) |
| ∀x ¬*P(x)* | ≡ | ¬∃x *P(x)* |
| ∃x ¬*P(x)* | ≡ | ¬∀x *P(x)* |
| ∀x ∃y *T(x, y)* | ← | ∃y ∀x *T(x, y)* |
| ∀x ∀y *T(x, y)* | ≡ | ∀y ∀x *T(x, y)* |
| ∃x ∃y *T(x, y)* | ≡ | ∃y ∃x *T(x, y)* |
| ∀x (*P(x)* ∨ *R*) | ≡ | (∀x *P(x)* ∨ *R*) |
| ∃x (*P(x)* ∧ *R*) | ≡ | (∃x *P(x)* ∧ *R*) |
| ∀x (*P(x)* → *R*) | ≡ | (∃x *P(x)* → *R*) |
| ∃x (*P(x)* → *R*) | → | (∀x *P(x)* → *R*) |
| ∀x (*R* → *Q(x)*) | ≡ | (R → ∀x *Q(x)*) |
| ∃x (*R* → *Q(x)*) | → | (R → ∃x *Q(x)*) |
| ∀x *R* | ← | *R* |
| ∃x *R* | → | *R* |

**Note**: The above formulas are valid in classical [first-order logic](https://en.wikipedia.org/wiki/First-order_logic) assuming that *x* does not occur free in *R*.

**Invalid Quantifier Formulas**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** |  | **B** | **Counterexample** |
| ∃x (*P(x)* ∧ *Q(x)*) | ← | (∃x *P(x)* ∧ ∃x *Q(x)*) | D = *{a, b}*, M = {*P(a)*, *Q(b)*} |
| ∀x (*P(x)* ∨ *Q(x)*) | → | (∀x *P(x)* ∨ ∀x *Q(x)*) | D = *{a, b}*, M = {*P(a)*, *Q(b)*} |
| ∀x (*P(x)* → *Q(x)*) | → | (∃x *P(x)* → ∀x *Q(x)*) | D = *{a, b}*, M = {*P(a)*, *Q(a)*} |
| ∀x ∃y *T(x, y)* | → | ∃y ∀x *T(x, y)* | D = *{a, b}*, M = {*T(a, b)*, *T(b, a)*} |
| ∃x (*P(x)* → *R*) | ← | (∀x *P(x)* → *R*) | D = Ø, M = {*R*} |
| ∃x (*R* → *Q(x)*) | ← | (*R* → ∃x *Q(x)*) | D = Ø, M = Ø |
| ∀x *R* | → | *R* | D = Ø, M = Ø |
| ∃x *R* | ← | *R* | D = Ø, M = {*R*} |

**Note**: if empty domains are not allowed, then the last four implications above are in fact valid.

**Sources**:

* [SNHU MAT 230](https://www.snhu.edu/admission/academic-catalogs/coce-catalog#/courses/4kVhSZLtg) - Discrete Mathematics, zyBooks.
* See also “Harold’s Proofs Cheat Sheet”.
* <https://byjus.com/maths/set-theory-symbols/>
* <https://en.wikipedia.org/wiki/List_of_logic_symbols>
* <https://nokyotsu.com/qscripts/2014/07/distribution-of-quantifiers-over-logic-connectives.html>

**Logical Truth Tables**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **p** | **q** | **Conjuntion****(and)**∧ | **NAND**⊼ | **Disjuntion****(or)**∨ | **NOR**⊽ | **XOR**⊻,⊕ | **XNOR**⊙ | **Negation****(not)**¬**P** |
| **F** | **F** | F | T | F | T | F | T |  |
| **F** | **T** | F | T | T | F | T | F | T |
| **T** | **F** | F | T | T | F | T | F | F |
| **T** | **T** | T | F | T | F | F | T |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **p** | **q** | **Material Implication****(If … Then)****→** | **Biconditional****(Iff)****↔** | **Tautology****(True)**⊤ | **Contradiction****(False)**⊥ |
| **F** | **F** | T | T | T | F |
| **F** | **T** | T | F | T | F |
| **T** | **F** | F | F | T | F |
| **T** | **T** | T | T | T | F |

**Blank Truth Tables**

|  |  |
| --- | --- |
| **Inputs** | **Output** |
| **p** | **q** | **r** | **s** | **x** | **y** | **z** |
| **F** | **F** | **F** | **F** |  |  |  |
| **F** | **F** | **F** | **T** |  |  |  |
| **F** | **F** | **T** | **F** |  |  |  |
| **F** | **F** | **T** | **T** |  |  |  |
| **F** | **T** | **F** | **F** |  |  |  |
| **F** | **T** | **F** | **T** |  |  |  |
| **F** | **T** | **T** | **F** |  |  |  |
| **F** | **T** | **T** | **T** |  |  |  |
| **T** | **F** | **F** | **F** |  |  |  |
| **T** | **F** | **F** | **T** |  |  |  |
| **T** | **F** | **T** | **F** |  |  |  |
| **T** | **F** | **T** | **T** |  |  |  |
| **T** | **T** | **F** | **F** |  |  |  |
| **T** | **T** | **F** | **T** |  |  |  |
| **T** | **T** | **T** | **F** |  |  |  |
| **T** | **T** | **T** | **T** |  |  |  |

|  |  |
| --- | --- |
| **Inputs** | **Output** |
| **p** | **q** | **r** | **x** | **y** |
| **F** | **F** | **F** |  |  |
| **F** | **F** | **T** |  |  |
| **F** | **T** | **F** |  |  |
| **F** | **T** | **T** |  |  |
| **T** | **F** | **F** |  |  |
| **T** | **F** | **T** |  |  |
| **T** | **T** | **F** |  |  |
| **T** | **T** | **T** |  |  |

|  |  |
| --- | --- |
| **Inputs** | **Output** |
| **p** | **q** | **x** |
| **F** | **F** |  |
| **F** | **T** |  |
| **T** | **F** |  |
| **T** | **T** |  |