# Formal Logic

* **Logic** is a language for reasoning.
* An argument is a sequence of statements aimed at demonstrating the truth of an assertion.
  + The assertion at the end of the sequence is called the **conclusion**, and the preceding statements are called **premises**.
* We are interested in whether a **statement** is true or false, and in determining truth/falsehood of statements from other statements.
* **Statement** (or **proposition**)**:** a sentence that is true or false, but not both.
* Much of Mathematics is about *proving* a statement is true, or *demonstrating* a statement is false.

## Logical Connectives

* **Connectives** are key words/symbols that connect two or more simple statements to form new, longer ones.
* We use *p*, *q*, *r*, ... to denote simple statements (**statement variables**) i.e. *p*: I need to work hard in MATH221.
* There are 5 Connectives:

|  |  |  |
| --- | --- | --- |
| Name | Example | Definition |
| Negation | ~ *p* | NOT *p* |
| Disjunction | *p* ∨ *q* | *p* OR *q* |
| Conjunction | *p* ∧ *q* | *p* AND *q* |
| Conditional | *p => q* (*p -> q*) | *p* IMPLIES *q* |
| Biconditional | *p <=> q* (*p <-> q*) | *p* IF AND ONLY IF *q* |

* **Compound statement:** an expression of simple statements and connectives.
  + Each simple statement has a truth value T for true and F for false.
* The truth value of a compound statement is determined by logic, using the simple statement values and the connectives.
* We do this by constructing *truth tables*.

## Negation

* If *p* is a statement variable, then “NOT *p*”, denoted by ~*p*, has the opposite value.
* If *p* is true, ~*p* is false.
* If *p* is false. ~*p* is true.

|  |
| --- |
| **Definition** |
| If *p* is a statement variable, the **negation** of *p* is “not *p*” or “it is not the case that *p*” and is denoted ~*p*. It has the opposite truth value from *p*: if *p* is true, ~*p* is false; if *p* is false, ~*p* is true. |

* The truth values for **negation** are summarised in a table.

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

* NOTE:
* The truth table above tells us that for any statement *p*, exactly *one* of *p* and ~*p* is true.
* This gives us 2 options for proving *p* is true:
* Show it directly
* Show indirectly by proving ~*p* is false **(proof by contradiction)**

## Priority

* In expressions that include the symbol ~ as well as ∧ or ∨, the **order of operations** specifies that ~ is performed first.
* ~*p* ∨ *q* means (~*p*) ∨ *q*, which is different from ~(*p* ∨ *q*)
* In logical expressions, as in ordinary algebraic expressions, the order of operations can be overridden through the uses of parentheses.

## Conjunction

* If *p* and *q* are statement variables, the conjunction is “*p* AND *q*”, denoted by *p* ∧ *q*.
* If *p* AND *q* are both true, then *p* ∧ *q* is true.
* Otherwise, *p* ∧ *q* is false.

|  |
| --- |
| **Definition** |
| If *p* and *q* are statement variables, the **conjunction** of *p* and *q* is “*p* and *q*,’ denoted by *p* ∧ *q*. It is true when, and only when, both *p* and *q* are true. If either p or q is false, or if both are false, *p* ∧ *q* is false. |

* The truth table for conjunction is:

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

## Disjunction

* If *p* and *q* are statement variables, the disjunction is “*p* OR *q*”, denoted by *p* ∨ *q*.
* If *p* and *q* are both false, *p* ∨ *q* is false.
* Otherwise, *p* ∨ *q* is true.

|  |
| --- |
| **Definition** |
| If *p* and *q* are statement variables, the **disjunction** of *p* and *q* is “*p* or *q,*” denoted by *p* ∨ *q*. It is true when either *p* is true, or *q* is true, or both *p* and *q* are true; it is false only when both *p* and *q* are false. |

* The truth table for disjunction is:

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

* NOTE:
* The word “OR” can be used in an exclusive sense, i.e. *p* OR *q* but not both.

## Exclusive OR

* The **exclusive or** statement is sometimes denoted by ⊗
* It can also be represented by AND/OR/NOT symbols. i.e. *p* ⊗ *q* = “*p* OR *q*, but NOT BOTH”
* Note that when *or* is used in its exclusive sense, the statement “*p* or *q*” means “*p* or *q* but not both” or “*p* or *q* and not both *p* and *q,*” which can be denoted as: (p ∨ q) ∧ ~(p ∧ q).
* The truth table for exclusive or is:

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

# Conditionals

* When you make a logical inference or deduction, you reason from a *hypothesis* to a *conclusion.*
* The statement has the form “if something is true, then something else is true.”
* If *p* and *q* are statement variables, the **conditional** of *q* by *p* is “If *p*, then *q*” or “*p* IMPLIES *q*”, denoted by *p* => *q*.
* If *p* is true and *q* is false, then *p* => *q* is false.
* Otherwise, *p* => *q* is true
* *p* is the **hypothesis** (**antecedent**)
* *q* is the **conclusion** (**consequent**)
* Conditionals take priority over conjunctions and disjunctions.

|  |
| --- |
| **Definition** |
| IF *p* and *q* are statement variables, the **conditional** of *q* by *p* is “if *p*, then *q”* or “*p* implies *q,*” denoted by *p* => *q*. If *p* is true and *q* is false, then *p* => *q* is false. Otherwise, *p* => *q* is true. |

* The truth table for conditionals is:

|  |  |  |
| --- | --- | --- |
| ***p*** | ***q*** | ***p* => *q*** |
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

* NOTE:
* Why is *p* => *q* true when *p* is false.
* If a statement cannot be said to be false, then it is true.
* If *p* is false, then we cannot say that *p* => *q* is false, it is true.
* Consider the claim, “if it rains, then I will go home.”
* Only if it rains can we make a judgement on its truth.
* If it doesn’t rain, then regardless of whether or not I go home, we cannot claim that the statement is false. So, it is true.

Exercise:

Write using connectives: “If x2 = 4, then x = 2 or x = -2.”



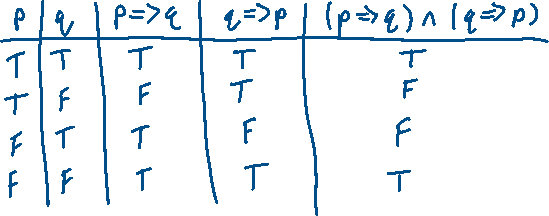
## Biconditionals

* A **biconditional** statement has the form “*p* if and only if *q*” or “*p* IFF *q.*”
* It’s true only if both variables have the same value.
* It is denoted by *p* <=> *q*, and is read
* *p* IFF *q*
* *p* is EQUIVALENT to *q*
* *p* IMPLIES AND IS IMPLIED by *q*
* *p* is NECESSARY AND SUFFICIENT for *q*
* The truth table for biconditionals is:

|  |  |  |
| --- | --- | --- |
| ***p*** | ***q*** | ***p <*=> *q*** |
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | T |

Exercise:

a) *p*: x3 = -8, *q*: x = -2, *p* <=> *q.*



b) Write using connectives: “Michael is a bachelor if and only if he is male and never married.”



EXERCISE:

Complete the table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **p** | **q** | **p => q** | **q => p** | **p <=> q** | **(p => q) ∧ (q => p)** |
| **T** | **T** |  |  |  |  |
| **T** | **F** |  |  |  |  |
| **F** | **T** |  |  |  |  |
| **F** | **F** |  |  |  |  |

* Notice that the last two columns are the same.
* This means that p <=> q and (p => q) ∧ (q => p) are **logically equivalent**.
* NOTE:
* Notice that *p* <=> *q* means (*p* => *q*) ∧ (*q* => *p*)

## Main Connectives

* When building compound statements, use parentheses to avoid ambiguity.
* The **main connective** is the one that binds the whole statement together.
* We must know the ranking of all connectives in a statement.
* E.g.

(*p* ∨ ~*q*) => (*p* ∧ *q*)



~[(*p* ∧ *q*) ∨ (~*p* ∧ *q*)]

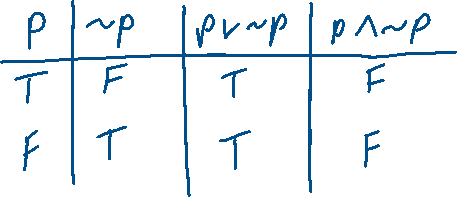
## Tautology and Fallacy

* A **tautology** is a compound statement that is always true, for all values of the basic statements
* E.g. *p* ∨ ~*p*
* A **fallacy** is a compound statement that is always false, for all values of the basic statements
* E.g. *p* ∧*~p*
* Any statement that is neither a tautology nor a fallacy is called **contingent** or **intermediate**.
* Note that the negative of a tautology is a fallacy, and vice versa.

|  |
| --- |
| **Definition** |
| A **tautology** is a statement form that is always true regardless of the truth values of the individual statements substituted for its statement variables. A statement whose form is a tautology is a **tautological statement**. |
| A **contradiction** (**fallacy**) is a statement that is always false regardless of the truth values of the individual statements substituted for its statement variables. A statement whose form is a contradiction is a **contradictory statement**. |

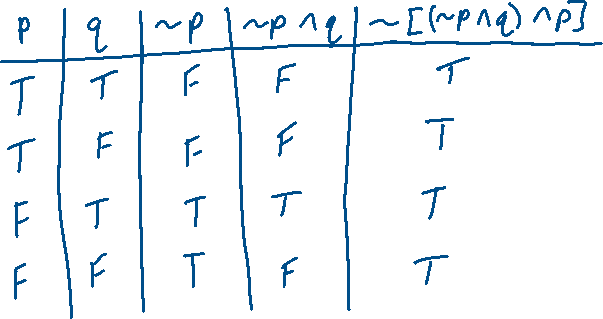
Exercise:

Show that for any statement p, p ∨ ~p is a tautology and p ∧~p is a fallacy.



Exercise:

Determine whether ~[(*~p* ∧ *q*) ∧ *p*] is a tautology, fallacy, or contingent statement.



## “Quick” Method of Identifying Tautologies

* With truth tables, 2n rows are required.
* This gets big and impractical quickly (i.e. 4 statements requires 16 rows, 5 statements requires 32 rows, … )
* There is a quicker method we can use:
* NOTE:
* Truth tables are reliable; it’s not easy to make mistakes. The quick method can be more difficult in that respect.
* It relies on the fact that if a false can occur under the main connective, then the statement is not a tautology.
* If a false is not possible, it is a tautology.
* They method is:
* Assume the main connective yields a false, then work backwards to see if a valid combination of values exists.
* Firstly, place an F under the main connective.

(*p* ∧ *q*) => (*r* ∧ *s*)



* Remember the conditional table: for this to happen, *p* ∧ *q* must be true and *r* ∧ *s* must be false.

(*p* ∧ *q*) => (*r* ∧ *s*)



* Since these are perfectly valid values for *p*, *q*, *r*, *s*, we have that (*p* ∧ *q*) => (*r* ∧ *s*) is *not* a tautology.

Exercise:

a) Is [(*p* => *q*) ∧ (*q* => *r*)] => (*p* => *r*) a tautology?



b) Make the truth table for this statement, and verify that the last column is all T.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **p** | **q** | **r** | ***p* => *q*** | ***q* => *r*** | ***p* => *r*** | **[(*p* => *q*) ∧ (*q* => *r*)] => (*p* => *r*)** |
| **T** | **T** | **T** |  |  |  |  |
| **T** | **T** | **F** |  |  |  |  |
| **T** | **F** | **T** |  |  |  |  |
| **T** | **F** | **F** |  |  |  |  |
| **F** | **T** | **T** |  |  |  |  |
| **F** | **T** | **F** |  |  |  |  |
| **F** | **F** | **T** |  |  |  |  |
| **F** | **F** | **F** |  |  |  |  |

## Logical Equivalence

* Two statements are called **logically equivalent** if, and only if, they have identical truth tables.
* The logical equivalence of *p* and *q* is denoted by *p* ≡ *q.*
* *p* and *q* are logically equivalent IFF *p* <=> *q* is a tautology.

Exercise:

Is *p* ≡ ~ (~*p*)?

|  |  |  |
| --- | --- | --- |
| ***p*** | ***~p*** | **~(~*p*)** |
| T | F | T |
| F | T | F |



## Substitution of Equivalence

* We can make substitutions in statements, using equivalence expressions.
* There are 2 rules:

### Rule of Substitution

* If in a tautology all occurrences of a variable are replaced by the same statement, the result is another tautology:

**Example:**

*p* ∨ ~*p* is a tautology, so *q* ∨ ~*q* is as well, and [(*p* ∨ *q*) => *r*] ∨ ~[(*p* ∨ *q*) => *r*] is as well.

### Rule of Substitution of Equivalence

* If in a tautology we replace any part of a statement by a statement equivalent to that part, the result is another tautology.

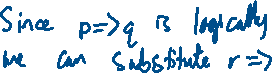
**Example:**

*p* ≡ ~(~*p*), so the tautology *p* ∨ ~*p* can be written ~(~*p*) ∨ ~*p*.

Exercise:

*p => q* is logically equivalent to ~*p* ∨ *q*. *q* => (*p* => *q*) is a tautology.

Prove that *s* => (*~r* ∨ *s*) is a tautology.



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Truth Tables Cheat Sheet**Conjunction**  |  |  |  | | --- | --- | --- | | *p* | *q* | *p* ∧ *q* | | T | T | T | | T | F | F | | F | T | F | | F | F | F |  **Disjunction**  |  |  |  | | --- | --- | --- | | *p* | *q* | *p* ∨ *q* | | T | T | T | | T | F | T | | F | T | T | | F | F | F |  **Conditionals**  |  |  |  | | --- | --- | --- | | *p* | *q* | *p* => *q* | | T | T | T | | T | F | F | | F | T | T | | F | F | T |  **Biconditionals**  |  |  |  | | --- | --- | --- | | *p* | *q* | *p <*=> *q* | | T | T | T | | T | F | F | | F | T | F | | F | F | T | | Logical Equivalence Laws**Commutative Laws**  1. ***p* ∨ *q* ≡ *q* ∨ *p*** 2. ***p* ∧ *q* ≡ *q* ∧ *p*** 3. ***p* <=> *q* ≡ *q* <=> *p***  **Associative Laws**  1. **(*p* ∨ *q*) ∨ *r* ≡ *p* ∨ (*q* ∨ *r*)** 2. **(*p* ∧ *q*) ∧ *r* ≡ *p* ∧ (*q* ∧ *r*)** 3. **(*p* <=> *q*) <=> *r* ≡ *p* <=> (*q* <=> *r*)**  **Distributive Laws**  1. ***p* ∨ (*q* ∧ *r*) ≡ (*p* ∨ *q*) ∧ (*q* ∨ *r*)** 2. ***p* ∧ (*q* ∨ *r*) ≡ (*p* ∧ *q*) ∨ (*p* ∧ *r*)** 3. ***p* => (*q* ∨ *r*) ≡ (*p* => *q*) ∨ (*p* => *r*)** 4. ***p* => (*q* ∧ *r*) ≡ (*p* => *q*) ∧ (*p* => *r*)**  **Double Negative Law**  1. **~ (~ *p*) ≡ *p***  **De Morgan’s Laws**  1. **~ (*p* ∨ *q*) ≡ ~ *p* ∧ ~ *q*** 2. **~ (*p* ∧ *q*) ≡ ~ *p* ∨ ~ *q***  **Implication Laws**  1. ***p* <=> *q* ≡ (*p* => *q*) ∧ (*q* => *p*)** 2. ***p* => *q* ≡ ~ *p* ∨ *q*** 3. ***p* => *q* ≡ ~ *q* => ~ *p*** 4. **~ (*p* => *q*) ≡ *p* ∧ ~ *q*** |

### De Morgan’s Laws

|  |
| --- |
| **Definition** |
| The negation of an *and* statement is logically equivalent to the *or* statement in which each component is negated. |
| The negation of an *or* statement is logically equivalent to the *and* statement in which each component is negated. |

Exercise:

To understand De Morgan’s Laws, write negatives of these:

a) John is 6 feet tall and he weighs at least 200 pounds



b) The bus was late or Tom’s watch was slow



Exercise:

Prove De Morgan’s Laws using truth tables.

~ (*p* ∧ *q*) ≡ *~ p* ∨ ~ *q*

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

~ (*p* ∨ *q*) ≡ *~ p* ∧ ~ *q*

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

### Inequalities and De Morgan’s Laws

Textbook Exercise:

Use De Morgan’s laws to write the negation of -1 < x ≤ 4.



* De Morgan’s laws are frequently used in writing computer programs.
* For instance, supposed you the program to delete all files modified outside a certain range of dates, say from date1 through date2 inclusive.
* You would use the fact that:

~(*date1* ≤ *file\_modification\_date* ≤ *date2*)

is equivalent to:

(*file\_modification\_date < date1*) or (*date2* < *file\_modification\_date*)

Exercise:

Is (*p* ∧ *~q*) ∧ (~*p* ∨ *q*) a tautology or a fallacy?



Exercise:

Is (*p* <=> *q*) <=> (~*p* <=> *q*) a tautology or a fallacy?



Exercise:

Prove the equivalence (*p* => *q*) => *r* ≡ [(~*p* => *r*) ∧ (*q* => *r*)]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***p*** | ***q*** | **r** | **(p => q)** | **(~p => r)** | **(q => r)** | **(p => q) => r** | **[(~*p* => *r*) ∧ (*q* => *r*)]** |
| T | T | T |  |  |  |  |  |
| T | T | F |  |  |  |  |  |
| T | F | T |  |  |  |  |  |
| T | F | F |  |  |  |  |  |
| F | T | T |  |  |  |  |  |
| F | T | F |  |  |  |  |  |
| F | F | T |  |  |  |  |  |
| F | F | F |  |  |  |  |  |

