A Short Course in Discrtete Mathematics for the Computer Science Undergraduate

Dale Fletter

UC Davis, ECS20, Summer Session 2

Propositional Logic

Proposition

A proposition is a declarative sentence which is true or false, but not both. Also called a statement. In symbolic logic we use single lowercase letters beginning with p to represent propositions. These are called *propositional variables*.

Compound Proposition

A **compound proposition** is a proposition formed from atomic propositions with logical connectives or logical operators. Also called a logical or Boolean expression. An atomic propositions is one that is not a compound propositions.

Negation

Let p be a proposition. The *negation of p*, denoted by $\neg p$ (also denoted by $\overline{p}$, is the statement *"It is not the case that p."* The proposition $\neg p$ is read "not p". The truth value of the negation of p, $\neg p$, is the opposite of the truth value of p.

Conjunction

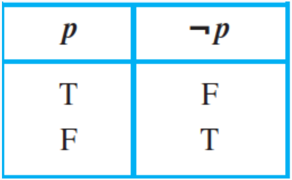
Let p and q be propositions. The *conjunction* of p and q, denoted by $p \wedge q$, is the proposition *p and q*. The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

Disjunction

Let p and q be propositIons. The disjunction of p and q, denoted by $p \vee q$, is the proposition *p or q*. The disjunction $p \vee q$ is false when both p and q are false and true otherwise.

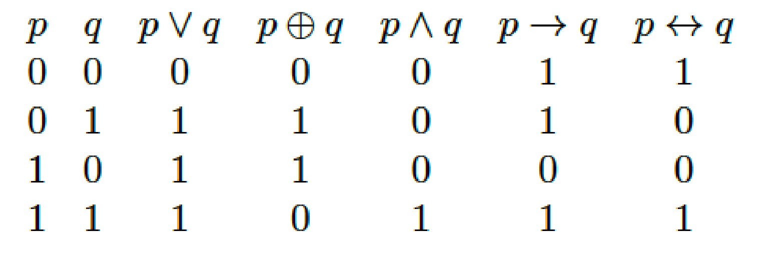
Exclusive OR

Let p and q b propsition. The *exclusive or* of p and q, denoted by $p \oplus q$, is the proposition that is true when exactly one of p and q is true and is false otherwise.



p	$\neg p$
T	F
F	T

Table 1: The Truth Table for the Negation of a Proposition



p	q	$p \vee q$	$p \oplus q$	$p \wedge q$	$p \rightarrow q$	$p \leftrightarrow q$
0	0	0	0	0	1	1
0	1	1	1	0	1	0
1	0	1	1	0	0	0
1	1	1	0	1	1	1

Table 2: Bitwise Logic

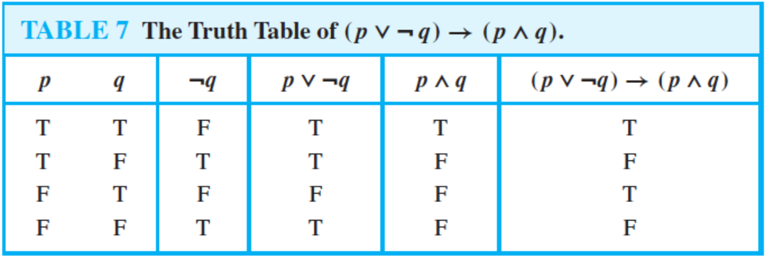
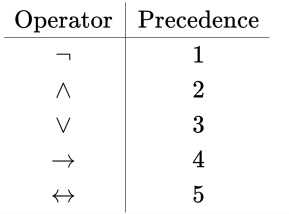


TABLE 7 The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$.

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

Table 3: Evaluation of a Compound Proposition

Precedence of Elementary Logical Operators



Operator	Precedence
$\neg$	**1**
$\wedge$	**2**
$\vee$	**3**
$\rightarrow$	**4**
$\leftrightarrow$	**5**

Table 4: Precedence of Logical Operators

Bitstring Operations

A bit is a binary digit, typically a zero or a one. A bit string is a sequence of zero or more bits. The length of this string is the number of bits in the string. Given two bit strings of the same length we define bitwise operations:

1. bitwise OR,
2. bitwise AND,
3. bitwise XOR.

Material Implication

Implication/Conditional

Let p and q be propositions. The conditional statement $p \rightarrow q$ is the proposition "if p then q". The conditional statement $p \rightarrow q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \rightarrow q$, p is called the **hypothesis** (or **antecedent** or **premise**) and q is called the **conclusion** or **consequent**.

Inverse, Converse, Contrapositive

Given a conditional statement $p \to q$,

1. its **inverse** is the statement $q \to p$,
2. its **converse** is $\neg p \to \neg p$ and
3. its **contrapositive** is the statement $\neg q \to \neg p$

Biconditional

Let p and q be propositions. The *biconditional statement* $p \leftrightarrow q$ is the proposition "p if and only if q." The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, an d is false otherwise. Biconditional statements are also called *bi-implicatons*. We can also read this as "p is logically equivalent to q".

Tautology and Contradiction

A compound proposition that is always true , no matter what the truth value of the poisitons that occur in it, is called a *tautology*. A compound proposition that is always false is called a *contradiction*. A compound proposition that is neither a tautology nor a contradiction is called a *contingency*.

Logical Equivalence

The compound proposition p and q are called *logically equivalent* if $p \leftrightarrow q$ is tautology. The notation $p \equiv q$ denotes that p and q are logically equivalent.

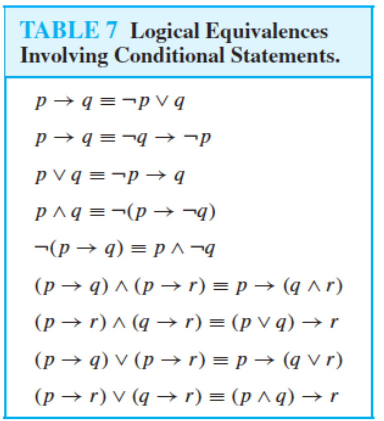


TABLE 7 Logical Equivalences Involving Conditional Statements.
$p \to q \equiv \neg p \vee q$
$p \to q \equiv \neg q \to \neg p$
$p \vee q \equiv \neg p \to q$
$p \wedge q \equiv \neg(p \to \neg q)$
$\neg(p \to q) \equiv p \wedge \neg q$
$(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$
$(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$
$(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$
$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$

Table 5: Logical Equivalences Involving Conditional Statements

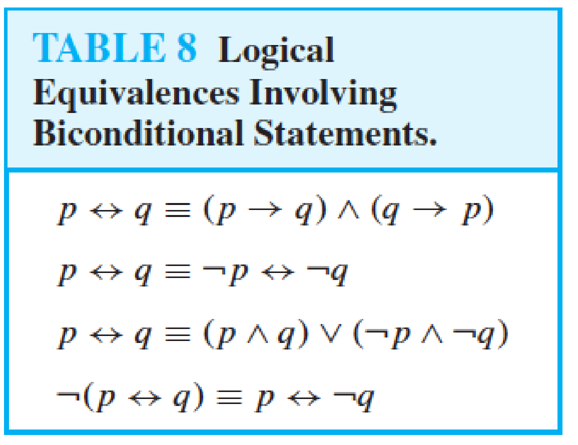


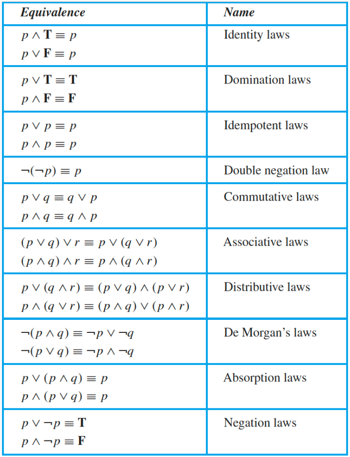
TABLE 8 **Logical Equivalences Involving Biconditional Statements.**
$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Table 6: Logical Equivalences Involving Bi-Conditional Statements

Rule of Substitution

Whenever two compound propositions are equivalent, one may be substituted for the other with no impact on the truth value.

An Algebra of Propositional Logic



Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

Table 7: Logical Equivalences

Consistency

A group of propositions are said to be *consistent* if and only if there is some assignment of truth values to the atomic propositions that make all of the given propositions true in at least one case. NOTE: system specifications that are not consistent can never be satisfied.

1. Abbott is a gunslinger.
2. Bob is a gunslinger or Charleen is a pugalist.
3. Charleen is a pugalist and Davida is a bartender.

Extract the atomic propositions...

1. p: Abbott is a gunslinger
2. q: Bob is a gunslinger
3. r: Charleen is a pugalist.
4. s: Davida is a bartender.

Now restate original sentences symbolically...

1. p
2. $q \vee r$
3. $r \wedge s$

Now build truth table and search for at least one row where all the original assertions can be true.

p	q	r	s	$q \vee r$	$r \wedge s$
T	T	T	T	T	T
T	T	T	F	T	F
T	T	F	T	T	F
T	T	F	F	T	F
T	F	T	T	T	T
T	F	T	F	T	F
T	F	F	T	F	F
T	F	F	F	F	F
F	T	T	T	T	T
F	T	T	F	T	F
F	T	F	T	T	F
F	T	F	F	T	F
F	F	T	T	T	T
F	F	T	F	T	F
F	F	F	T	F	F
F	F	F	F	F	F

Table 8: Are the propositions consistent?

The last problem can be solved through deduction. p must be true. r and s must be true. q can be either true or false.

The two combinations are TTTT and TFTT.

Paradox

Some declarative statements defy taking a truth value. For example, “This statement is false” cannot be assigned either true or false. It it is false, then the statement must be true. But if it is true, then the statement must be false. Such a statement is called a **paradox**.

A Smullyan Logic Puzzle

Knights always tell the truth.

Knaves always lie.

No other possibilities.

p: person who answered door is a knight

q: other person in house is a knight

p	q
T	T
T	F
F	T
F	F

Table 9: Who lives in the house?

Another Smullyan Puzzle

A census taker comes to the door where two people live. The person who answers says, “We are both knaves.” What do you conclude?

Translate into propositions. Two people, each can either be a knave or a knight. p: A is a knight q: B is a knight

Predicate Logic

Propositional logic is sometimes called “zeroeth order logic”. There is reasoning that cannot be done with it like, “Socrates is a man, all men are mortal, therefore Socrates is mortal”. The ancient Greeks had worked out a form of reasoning for this kind of challenge but we found a new way in the 19thc and now use a different form which we call *Predicate Logic*, *Predicate Calculus* or *first order logic*. There are other higher order logics.

Predicates

A *predicate* is a function which when applied to an object will evaluate to true when some property is present and false otherwise. We denote the predicate using a capital letter and write the expression using a functional notation.

We often define a predicate using a notation like, $P : x + 1 > 0$.

So the predicate P when applied to the variable x is written $P(x)$ and will take on the value of true or false when the value of x has been fixed (bound).

N-Place Predicates

Predicates are not limited to one argument but can have any number of arguments, called *n-place predicates*. For example the predicate S could be "...have the same color" and can accept pieces of fruit as objects. Then the expression $S(p, q, r, s, t)$ will be true if the color of each piece of fruit p, q, r, s, t matches the others and false otherwise.

Binding and Scope

Variables can be bound to values. But we often wish to assert a proposition over a range of values or to claim that there is an object with some property. This process of creating a proposition over some range of objects is called *quantification*. The two fundamental quantifications are the universal and the existential.

Universal Quantification

Quantification] The universal quantification of P(x) is the statement

$$"P(x) \text{ for all values of } x \text{ in the domain."}$$

The notation $\forall x P(x)$ denotes the universal quantification of $P(x)$. Here $\forall$ is called the universal quantifier. We read $\forall x P(x)$ as "for all x P(x)" or "for every x P(x)". An element for which $P(x)$ is false is called a counterexample of $\forall x P(x)$.

Existential Quantification

The *existential quantification* of $P(x)$ is the proposition

> "There exists an element x in the domain such that $P(x)$."

We use the notation $\exists x P(x)$ for the existential quantification of $P(x)$. Here $\exists$ is called the **existential quantifier**. The domain must always be specified when a statement $\exists P(x)$ is used. The meaning of $\exists P(x)$ changes when the domain changes. without specifying the domain, the statement $\exists P(x)$ has no meaning. The existential quantifier $\exists P(x)$ is read as, "There is an x such that $P(x)$, "There is at least one x such that $P(x)$", or "For some $xP(x)$".

Uniqueness

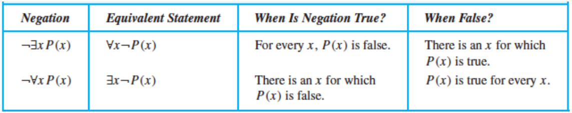
The *uniqueness quantification* of $P(x)$ is the proposition:

"There exists exactly one element x in the domain such that $P(x)$".

We use the notation $\exists!$ for the uniqueness quantification.

Binding and Scope of Variables

When a variable has been assigned a value, we say the value has been **bound** to the variable. Any variable that has not yet been bound to a value is said to be **free**. The **scope** of the binding is controlled by the use of parentheses or other marks. The value of a variable is also bound by the use of a quantifier and the variable is either within or outside the scope of that quantifier.



Negation	*Equivalent Statement*	*When Is Negation True?*	*When False?*
$\neg\exists x P(x)$	$\forall x \neg P(x)$	For every x, $P(x)$ is false.	There is an x for which $P(x)$ is true.
$\neg\forall x P(x)$	$\exists x \neg P(x)$	There is an x for which $P(x)$ is false.	$P(x)$ is true for every x.

Table 10: DeMorgansForQuantifiedExpressions

Restricted Domains

An abbreviated notation is sometimes used to specify some subset of the domain. For example $\forall x < 0(x^2 > 0)$ in the domain of real numbers places the restrictive clause next to the quantifier.

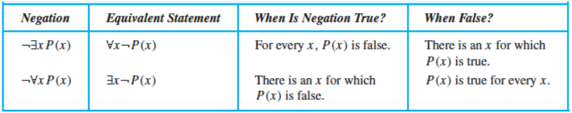
Variable Binding and Scope

When a variable has been assigned a value, we say the value has been **bound** to the variable. Any variable that has not yet been bound to a value is said to be **free**. The **scope** of the binding is controlled by the use of parentheses or other marks. The value of a variable is also bound by the use of a quantifier and the variable is either within or outside the scope of that quantifier.

Universal quantifiers at the outermost level can be omitted, i.e., free variables are interpreted as universally quantified at the outermost level. Quantifiers can be applied to more than one variable at once (e.g., $\forall x, y$).

Logical Equivalence of Quantified Expressions

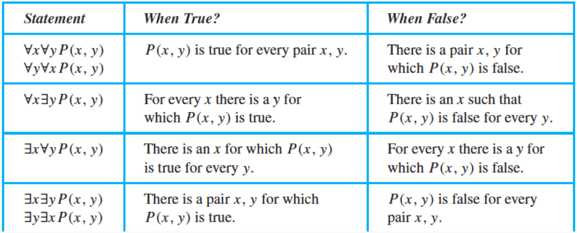
Statements involving predicates and quantifier are *logically equivalent* if and only if they have trhe same truth value no matter which predicates are substitued into these statements and which domain of discourse is used for the variables in these propositionan functions. We use the notation $S \equiv T$ to indicate that two statements S and T involving predicates and quantifiers are logically equivalent.



Negation	*Equivalent Statement*	*When Is Negation True?*	*When False?*
$\neg\exists x P(x)$	$\forall x \neg P(x)$	For every x, $P(x)$ is false.	There is an x for which $P(x)$ is true.
$\neg\forall x P(x)$	$\exists x \neg P(x)$	There is an x for which $P(x)$ is false.	$P(x)$ is true for every x.

Table 11: DeMorgansForQuantifiedExpressions

Quantification of Two Variables



Statement	*When True?*	*When False?*
$\forall x \forall y P(x, y)$ $\forall y \forall x P(x, y)$	$P(x, y)$ is true for every pair x, y.	There is a pair x, y for which $P(x, y)$ is false.
$\forall x \exists y P(x, y)$	For every x there is a y for which $P(x, y)$ is true.	There is an x such that $P(x, y)$ is false for every y.
$\exists x \forall y P(x, y)$	There is an x for which $P(x, y)$ is true for every y.	For every x there is a y for which $P(x, y)$ is false.
$\exists x \exists y P(x, y)$ $\exists y \exists x P(x, y)$	There is a pair x, y for which $P(x, y)$ is true.	$P(x, y)$ is false for every pair x, y.

Table 12: Quant Of 2 Var

The Order of Quantifiers in an Expression

It is important to note that the order of the quantifiers can make a difference. For example is this statement true or false for the domain of integers? $\forall x \exists y : (x + y = 10)$. This is easily proven true with a bit of algebra. But what about $\exists y \forall x : (x + y = 100)$. This is false since there is no such integer that will always give 10 when added to any other integer.

Seeing Nested Quantified Expressions as Loops

There Is No Largest Prime Number

Theorem
There is no largest prime number.

Proof.

1. Suppose p were the largest prime number.
2. Let q be the product of the first p numbers.
3. Then $q + 1$ is not divisible by any of them.
4. But $q + 1$ is greater than 1, thus divisible by some prime number not in the first p numbers. □