An Introduction to Mathematical Proofs

Nicholas A. Loehr

April 2022

Contents

1 Logic

1.1 Propositions, Logical Connectives, and Truth Tables

1.1.1 Propositions

Propostional logic **studies how the truth of a complex statement is determined by the truth or falsehoods of its parts.**

A *propositon* is a statement that is either true or false, but not both.

1.1.2 Propositional Forms

An expression which is built up by combining *propositional variables* (capital letters) using logical symbols, is called a *propositional form*

The following tables shows the symbols used in propositional logic and their meaning:

Logical Symbol	English Translation
$\neg P$	P is not true
$P \wedge Q$	P and Q.
$P \vee Q$	P or Q.
$P \oplus Q$	P or Q, but not both
$P \implies Q$	if P, then Q
$P \iff Q$	P if and only if Q

We can also combine this symbols:
$(P \wedge (\neg Q)) \vee ((\neg P) \wedge Q)$

Any expression which is built up by combining *propositional variables* (capital letters) using logical symbols, is called a *propositional form*

1.1.3 Definitions of NOT and AND

In order to properly *define* the exact meaning of logical connective words like NOT, AND, OR, IF, etc. we use so-called *truth tables*. that show how to combine the truth values of propositions to obtain the truth value of a new proposition built from these using a logical connective. The letters T and F stand for *true* and *false* respectively.

Definition of NOT. For any proposition P, the truth value of $\neg P$ ("not P") is determined by the following table

P	$\neg P$
T	F
F	T

Definition of AND. For any propositions, P, Q, the truth value of $P \wedge Q$ ("P and Q") is determined by the following table.

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

1.1.4 Definitions of OR and XOR

$\vee$ stands for *inclusive*-OR, and $\oplus$ stands for *exclusive*-OR.

Definition of OR. For any propositions, P, Q, the truth value of $P \vee Q$ ("P or Q") is determined by the following table.

P	Q	$P \wedge Q$
T	T	T
T	F	T
F	T	T
F	F	F

Definition of XOR. For any propositions, P, Q, the truth value of $P \oplus Q$ ("P xor Q") is determined by the following table.

P	Q	$P \wedge Q$
T	T	F
T	F	T
F	T	T
F	F	F

1.1.5 Formal Definition of Propositional Forms

In the main text, we informally defined a propositional form to be any expression built up by combining propositional variables using logical symbols. To give a precise, rigorous definition of propositional forms, we need a *recursive definition*.

Specifically, propositional forms are defined recursively via the following rules:

(a) A single capital italic letter is a propositional form.

(b) If A is a propositional form, then $(\neg A)$ is a propositional form.

(c) If A and B are any propositional forms, then $(A \wedge B)$ is a propositional form.

(d) If A and B are any propositional forms, then $(A \vee B)$ is a propositional form.

(e) If A and B are any propositional forms, then $(A \oplus B)$ is a propositional form.

(f) If A and B are any propositional forms, then $(A \implies B)$ is a propositional form.

(g) If A and B are any propositional forms, then $(A \iff B)$ is a propositional form.

(h) An expression is a propositional form only if it can be formed by applying rules (a) through (g) finetly many times.

1.1.6 Remark: Terminology for Propositional Forms

Let A and B be any propositional forms. In some logic texts, $(\neg A)$ is called the *negation* of A; $(A \wedge B)$ is called the *conjunction* of A and B; $(A \vee B)$ is called the *disjunction* of A and B; $(A \implies B)$ is called an *implication* or *conditional* with *hypothesis* of A and B.

1.2 Logical Equivalences and IF-Statements

1.2.1 Logical Equivalence

Definition: Logically Equivalent Propositional Forms
Two propositional forms A and B are *logically equivalent* when the truth tables for A and B have outputs that agree in every row. We write $\boxed{A \equiv B}$ when A and B are logically equivalent; we write $\boxed{A \not\equiv B}$ when A and B are not logically equivalent.

Theorem on Logical Equivalence. For all propositional forms P, Q, and R, the following logical equivalences hold:

(a) *Commutative Laws:* $P \wedge Q \equiv Q \wedge P$, $P \vee Q \equiv Q \vee P$, and $P \oplus Q \equiv Q \oplus P$.

(b) *Associative Laws:* $P \wedge (Q \wedge R) \equiv (P \wedge Q) \wedge R$, $P \vee (Q \vee R) \equiv (P \vee Q) \vee R$, and $P \oplus (Q \oplus R) \equiv (P \oplus Q) \oplus R$.

(c) *Distributive Laws:* $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$, $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$, and $P \wedge (Q \oplus R) \equiv (P \wedge Q) \oplus (P \wedge R)$.

(d) *Idempotent Laws:* $P \wedge P \equiv P$ and $P \vee P \equiv P$

(e) *Absorption Laws:* $P \wedge (P \vee Q) \equiv P$ and $P \vee (P \wedge Q) \equiv P$.

(f) *Negation Laws:* $\neg(\neg P) \equiv P$, $\neg(P \wedge Q) \equiv (\neg P) \vee (\neg Q)$, and $\neg(P \vee Q) \equiv (\neg P) \wedge (\neg Q)$.

1.2.2 Converse and Contrapositive

Definition: Converse and Contrapositive
For any propositional forms P and Q: the $\boxed{\text{converse of } P \implies Q}$ is $\boxed{Q \implies P}$; the $\boxed{\text{contrapositive of } P \implies Q}$ is $\boxed{(\neg Q) \implies (\neg P)}$.

Theorem on IF. Let P and Q be distinct propositional variables.

(a) *Non-equivalence of Converse:* $P \Rightarrow Q \not\equiv Q \Rightarrow P$

(b) *Equivalence of Contrapositive:* $P \Rightarrow Q \equiv (\neg Q) \Rightarrow (\neg P)$

(c) *Elimination of IF:* $P \Rightarrow Q \equiv (\neg P) \vee Q$

(d) *Denial of IF:* $\neg(P \Rightarrow Q) \equiv P \wedge (\neg Q)$

1.2.3 Five Remarks on Logical Equivalence

(1). The Theorem on Logical Equivalence shows that many of the laws of logic have an algebraic character analogous to algebraic laws for arithmetical operations on real numbers. To make this analogy more explicit, we introduce the convention of representing true (T) by 1 and false (F) by 0. The symbols 0 and 1 are called *bits*. In this notation, the defining truth tables for the six logical operators (including *IFF*, discussed in the next section) look like this:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \oplus Q$	$P \Rightarrow Q$	$P \iff Q$
0	0	1	0	0	0	1	1
0	1	1	0	1	1	1	0
1	0	0	0	1	1	0	0
1	1	0	1	1	0	1	1

If we think of P and Q as bits (the actual integers 0 and 1, rather than propositions), we see that $\neg P = 1 - P$; $P \wedge Q = P \cdot Q = min(P, Q)$, the product or minimum of P and Q; $P \vee Q = max(P, Q)$, maximum of P and Q; and $P \oplus Q$ is the mod-2 sum of P and Q. We can also think of $\oplus$, $\Rightarrow$, and $\iff$ as *relations* comparing their two inputs. In this interpretation, $P \oplus Q$ is true precisely when $P \neq Q$, so $\oplus$A models non-equality of bits; $P \iff Q$ is true precisely when $P = Q$, so $\iff$ models equality of bits; and $P \Rightarrow Q$ is true precisely when $P \leq Q$, so $\Rightarrow$ models ordering of bits. Now the various identities in the Theorem on Logical Equivalences translate into identities for these algebraic operations on bits. For instance, supposing we already know that $x(yz) = (xy)z$ for all integers x, y, z we can deduce the associativity law $P \wedge (Q \wedge R) \equiv (P \wedge Q) \wedge R$ for propositional forms.

(2). Logical equivalence of propositional forms (denoted $\equiv$) is similar to, but not the same as, equality of forms (denoted $=$). Thinking of propositional forms A and B as strings of symbols, $A = B$ means that A has exactly the same symbols as B, in the same order. For example, $P \vee Q$ is not *equal* to $Q \vee P$ since the first symbols do not match. Yet, as we know, $P \vee Q \equiv Q \vee P$ since the two truth tables agree. For most purposes of logic, two logically equivalent forms are interchangeable with each other (since their truth values must agree), even though such forms are often not equal as strings.

(3). Although logical equivalence of propositional forms (denoted $\equiv$) is not the same concept as logical equality (denoted $=$), these two concepts share many common properties. For example, given any objects x, y, z whatsoever, the following facts are true about equality:

$$x = x, \text{ if } x = y \text{ then } y = x; \text{ if } x = y \text{ and } y = z \text{ then } x = z.$$

These facts are called *reflexivity*, *symmetry*, and *transitivity* of equality. Three analogous facts also hold for logical equivalence: given any propositional forms A, B, C,

$$A \equiv A; \text{ if } A \equiv B \text{ then } B \equiv A; \text{ if } A \equiv B \text{ and } B \equiv C \text{ then } A \equiv C.$$

These facts follow from the corresponding facts for equality, applied to the rows of the truth tables. For instance, if the output in every row of A's truth table equals the corresponding output of B's truth table, then the output in every row of B's truth table equals the corresponding output of A's truth table, which is why the second fact is true. The text implicitly uses the reflexivity, symmetry and transitivity of logical equivalence in many place. For instance, the derivation has the form $A \equiv B \equiv C \equiv D$, which is really an abbreviation for "$A \equiv B$ and $B \equiv C$ and $C \equiv D$." Using transitivity once, we see that $A \equiv C$. Using transitivity again, we see that $A \equiv D$, which is the required conclusion. Similarly, symmetry of $\equiv$ is needed whenever we want to use known logical equivalence in reverse, e.g., to replace $(P \wedge Q) \vee (P \wedge R)$ by $P \wedge (Q \vee R)$ by invoking the Distributive Law.

(4). The logical equivalences in this section's theorems apply to *all* propositional forms P, Q, R which means that we can replace each letter P, Q, R (wherever it appears in an identity) by a longer propositional form. For instance, replacing P by $A \Rightarrow B$ and Q by $B \oplus C$ in the second negation law, we see that $\neg((A \Rightarrow B) \wedge (B \oplus C)) \equiv (\neg(A \Rightarrow B)) \vee (\neg(B \oplus C))$. In contrast, consider the non-logical equivalence asserted in part (a) of the Theorem on IF. In that theorem, P and Q are distinct propositional variables (as opposed to general some choices of the form P and the form Q, the logical equivalence will hold; for instance, suppose P and Q are both R. It can be checked that when P and Q are logically equivalent forms, $(P \Rightarrow Q) \equiv (Q \Rightarrow P)$, because in this case we must always be in row 1 or row 4 of the truth table shown in the proof (In fact, using terminology from the next section, $P \Rightarrow Q$ and $Q \Rightarrow P$ are both tautologies in this situation.) For example, $((A \vee B) \Rightarrow (B \vee A))$ is logically equivalent to $((B \vee A) \Rightarrow (A \vee B))$. On the other hand, when P and Q are not logically equivalent, there must exist some assignment of truth values to the propositional variables appearing in P and Q that cause P and Q themselves to have opposite truth values. In this row of the truth table, $P \Rightarrow Q$ and $Q \Rightarrow P$ disagree, so that these forms are not logically equivalent. For instance, $((A \vee B) \Rightarrow (A \oplus B))$ and $((A \oplus B) \Rightarrow (A \vee B))$ are not logically equivalent, as we see by considering the row of the truth table where A and B are both true.

(5). In algebra, if we know two variable represent the same number, we can replace one variable by the other in longer expressions. We can express this fact symbolically by *substitution rules* such as the following: for all numbers x, y, z if $y = z$ then $x + y = x + z$ and $xy = xz$. There are analogous substitution rules for logically equivalent propositional forms. For instance, given any propositional forms A, B and C if $B \equiv C$, then $(A \vee B) \equiv (A \vee C)$, $(AB) \equiv (A \wedge C)$, $(A \Rightarrow B) \equiv (A \Rightarrow C)$, $(B \Rightarrow A) \equiv (C \Rightarrow A)$, and so on. We freely use these substitution rules without justification in the main text, but it is possible to provide each rule by comparing truth tables. As an example, assume $B \equiv C$, and consider a fixed row in the truth table for $A \Rightarrow B$ and $A \Rightarrow C$. Suppose that, for the values of the input propositional variables appearing in this row, A is true B is false. On one hand, $A \Rightarrow B$ must be false in this row. On the other hand, since B and C were assumed to be logically equivalent, C is false in this row, so $A \Rightarrow C$ is also false in this row. Thus, $A \Rightarrow B$ and $A \Rightarrow C$ have the same truth value (namely, false) in this row. Considering the other possible cases, we see that $A \Rightarrow B$ and $A \Rightarrow C$ have the same truth value in every row of the truth table, so they are logically equivalent. We can summarize the cases in a *meta-truth table* that looks like this:

A	B	C	$A \Rightarrow B$	$A \Rightarrow C$
T	T	T	T	T
T	F	F	F	F
F	T	T	T	T
F	F	F	T	T

We call this a meta-truth table because the underlying propositional variables appearing in A, B and C are not explicitly listed. (For example, these forms might involve five propositional variables P, Q, R, S and T, so that the actual truth table would have 32 rows.) However, to prove the logical equivalence of $A \Rightarrow B$ and $A \Rightarrow C$, we only need to know the truth values of the propositional forms, A, B and C. Row 2 of the table is an abbreviation for the discussion preceding the table, and the other rows cover the remaining cases. The key point is that the table has four rows, not eight, because we have assumed that B and C are logically equivalent forms, hence always have the same truth value. You can construct similar meta-truth tables to prove the other substitution principles.

1.3 IF, IFF, Tautologies, and Contradiction

1.3.1 Translations of $P \Rightarrow Q$

The fastest way to translate $P \Rightarrow Q$ is "if P then Q", however, there are many other translations that refer to the same thing.

There are some translations that are often confusing:

- $\boxed{P \text{ if } Q}$ means $\boxed{Q \Rightarrow P}$.
- $\boxed{P \text{ only if } Q}$ means $\boxed{P \Rightarrow Q}$.

The phrase "P if Q" is obtained by changing the order of the clauses "if Q then P". In both phrases, the proposition Q immediately follows the word IF, so Q is the hypothesis and P is the conclusion of the conditional statement. However, replacing IF by ONLY IF changes the meaning: in the sentence "P only if Q", P is the hypothesis and Q is the conclusion!

Now consider the meaning of the words NECESSARY and SUFFICIENT. By definition:

- $\boxed{P \text{ is sufficient } Q}$ means $\boxed{P \Rightarrow Q}$.
- $\boxed{P \text{ is necessary } Q}$ means $\boxed{Q \Rightarrow P}$.

All of the following phrases can be used to translate $P \Rightarrow Q$:

- P is sufficient for Q.
- For Q to be true, it is sufficient that P be true.
- A sufficient condition for Q is P.
- Q is necessary for P.
- A necessary condition for P is Q.
- For P to be true, it is necessary that Q be true.

Notice the key preposition *FOR*, which can help you locate the correct hypothesis and conclusion in each phrase. Yet other possible translations of $P \Rightarrow Q$ include:

- P implies Q.
- Q is implied by P.
- When P is true, Q is true.
- Q whenever P.
- Q follows from P.
- Q is a consequence of P.

In particular, when reading a form such as $P \Rightarrow Q$ aloud, you may read the symbol $\Rightarrow$ as the word IMPLIES. But it is *wrong* to read $\Rightarrow$ as the word IF, since "P if Q" means $Q \Rightarrow P$, not $P \Rightarrow Q$. It is correct, but potentially confusing, to pronounce the symbol $\Rightarrow$ as ONLY IF. Read $P \Rightarrow Q$ as "if P, then Q.".

We should also mention the logical meaning of a few other common English phrases:

- "P but Q" means $P \wedge Q$.
- "Although P, Q" means $P \wedge Q$.
- "P, Q are both true" means $P \wedge Q$.
- "Neither P nor Q" means $(\neg P) \wedge (\neg Q)$, or equivalently $\neg(P \vee Q)$.

The words BUT and ALTHOUGH suggest some kind of contrast between the two clauses joined by these words, but this contrast is ignored in logic. For logical purposes, BUT and ALTHOUGH (and BOTH) have exactly the same meaning as AND.

1.3.2 What IFF Means

IFF pronounced "if and only if" is symbolized by $\iff$.

Definition of IFF. For any propositions P and Q, the truth value of $P \iff Q$ ("P iff Q") is determined by the following table.

P	Q	$P \iff Q$
T	T	T
T	F	F
F	T	F
F	F	T

1.3.3 Theorem on IFF

For all propositional forms P and Q:

(a) *Reduction to IF:* $P \iff Q \equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P)$.

(b) *Reduction to AND and OR:* $P \iff Q \equiv (P \wedge Q) \vee ((\neg P) \wedge (\neg Q))$.

(c) *Symmetry of IFF:* $P \iff Q \equiv Q \iff P$.

(d) *Denial of IFF:* $\neg(P \iff Q) \equiv P \oplus Q$.

(e) *Negating Both Sides:* $P \iff Q \equiv (\neg P) \iff (\neg Q)$.

1.3.4 Translations of IFF.

We can translate $P \iff Q$ by several English phrases, including:

(a) P iff Q.

(b) P if and only if Q.

(c) P is necessary and sufficient for Q.

(d) P when and only when Q.

(e) P precisely when Q.

(f) P is equivalent to Q.

(g) P has the same truth value as Q.

1.3.5 Theorem on XOR.

For all propositional forms P, Q, R:

(a) *Reduction to AND and OR:* $P \oplus Q \equiv (P \wedge (\neg Q)) \vee (Q \wedge (\neg P))$.

(b) *Negating Both Sides:* $P \oplus Q \equiv (\neg P) \oplus (\neg Q)$.

(c) *Second Reduction to AND and OR:* $P \oplus Q \equiv (P \vee Q) \wedge (\neg(P \wedge Q))$.

(d) *Commutativity of XOR:* $P \oplus Q \equiv Q \oplus P$.

(e) *Denial of XOR:* $\neg(P \oplus Q) \equiv P \iff Q$.

(f) *Reduction to IFF:* $P \oplus Q \equiv P \iff (\neg Q) \equiv (\neg P) \iff Q$.

(g) *Associativity of XOR:* $P \oplus (Q \oplus R) \equiv (P \oplus Q) \oplus R$.

1.3.6 Tautologies and Contradictions

Definition: Tautologies and Contradictions. A propositional form A is called a *tautology* iff every row of the truth table for A has output true. A propositional form B is called a *contradiction* iff every row of the truth table for B has output false.
Note that most propositional forms have a mixture of true and false outputs, so *most propositional forms* are *neither tautologies nor contradictions.*

Example: The propositional form $R(\neg R)$ is a tautology, whereas the form $R \wedge (\neg R)$ as shown in the following truth table:

R	$\neg R$	$R \vee (\neg R)$	$R \wedge (\neg R)$
T	F	T	F
F	T	T	F

1.4 Propositions, Logical Connectives, and Truth Tables

1.4.1 Remark on tautologies.

Let A and B be any propositional forms, and let T be any tautology.

(a) $A \wedge T \equiv A$.

(b) $A \vee T$ is a tautology.

(c) A is a tautology iff $\neg A$ is a contradiction.

(d) $A \equiv B$ iff $A \iff B$ is a tautology.

1.4.2 Variables and Quantifiers

A statement containing a variable x, such as $x^2 = x$ (where x is a *variable* varying over all real numbers, is called an ***open sentence*** and is denoted by a symbol such as $P(x)$, $Q(x)$, etc. Open sentences can involve more than one variable; for instance, we could let $R(x, y)$ be the formula $x^2 + y^2 = 25$. ***An open sentence does not have a truth value; it is not a proposition***. One way to turn an open sentence into a proposition is to assign specific values to every variable appearing in it.

There are two other ways to turn an open sentence $P(x)$ into a proposition: we may *quantify* the variable x with a phrase such as "for all x" or "there exists an x". We use the *universal quantifier* $\forall$ to abbreviate "for all," and the *existential quantifier symbol* $\exists$ to abbreviate "there exists".

Definition of $\forall$ (Universal Quantifier).

Let $P(x)$ be an open sentence involving the variable x. The statement "$\forall x, P(x)$" is a proposition that is true iff for all objects x_0, $P(x_0)$ is true. The quantifier $\forall x$ may be translated "for all x", "for every x", "for each x", "for any x", among other possibilities. The symbol $\forall$ looks like an upside-down capital A, reminding us of the word ALL.

Definition of $\exists$ (Existential Quantifier).

Let $P(x)$ be an open sentence involving the variable x. The statement "$\exists x, P(x)$ is a proposition that is true iff there is a t least one object x_0 making $P(x_0)$ true. The quantifier $\exists x$ may be translated "there is an x", "there exists an x", "there is at least one x", "for some x", among other possibilities. The symbol $\exists$ looks like a backwards capital E, reminding us of the word EXISTS.

1.4.3 Universes; Restricted Quantifiers

Definition: Restricted Quantifiers.

Let U be any set, and let $P(x)$ be an open sentence involving x. The statement "$\forall x \in U, P(x)$ means that for every object x_0 in U, $P(x_0)$ is true. The statement "$\exists x \in U, P(x)$ means there exists at least one object x_0 in U for which $P(x_0)$ is true. The symbols "$\forall x \in U$" and "$\exists x \in U$" are *restricted quantifiers*; in contrast, the previous symbols "$\forall x$" and "$\exists x$" are *unrestricted quantifiers.*

Notation for Number Systems.

(a) $\mathbb{Z}$ is the set of all *integers*, namely $\mathbb{Z} = \ldots, -3, -2, -1, 0, 1, 2, 3, \ldots$.

(b) $\mathbb{Q}$ is the set of all *rational numbers*, which are rations a/b where a is an integer and b is a nonzero integer.

(c) $\mathbb{R}$ is the set of all *real numbers*.

(d) $\mathbb{C}$ is the set of all *complex numbers*, which are expressions $a + bi$ with a, b real, and where i satisfies $i^2 = -1$.

(e) We write $\mathbb{Z}_{\geq 0} = \{0, 1, 2, 3, \ldots\}$ for the set of nonnegative integers, and $\mathbb{Z}_{>0} = \{1, 2, 3, \ldots\}$ for the set of positive integers. Some authors use $\mathbb{N}$ (the set of *natural numbers*) to denote the set $\mathbb{Z}_{\geq 0}$, whereas other authors use $\mathbb{N}$ to denote the set $Z_{>0}$.

(f) By analogy with (e), let $\mathbb{R}_{>0}$ be the set of strictly positive real numbers (not including zero). For a fixed integer b, let $\mathbb{Z}_{\geq b}$ be the set of integers that are at least b, and let $\mathbb{Z}_{>b}$ be the set of integers strictly greater than b. We similarly define $\mathbb{Z}_{\neq b}$, $\mathbb{R}_{>x_0}$, $\mathbb{R}_{\leq x_0}$, and so on.

(g) Given n distinct objects $x_1, x_2, \ldots, x_n$, let $U = \{x_1, x_2, \ldots, x_n\}$ denote the finite universal set consisting of these n objects.

1.4.4 Quantifiers vs. Logical Operators

The quantifiers $\forall$ and $\exists$ can be viewed as generalizations of the logical operators $\wedge$ (AND) and $\vee$ (inclusive-OR), respectively. To see why, consider a finite universe $U = \{z_1, z_2, \ldots, z_n\}$. The universal statement "$\forall x \in U, P(x)$ is true iff $P(a)$ is true for each fixed object $a \in \{z_1, \ldots, z_n\}$. In other words, "$\forall x \in U, P(x)$ is true iff $P(z_1$ is true and $P(z_2$ is true and $\ldots$ and $P(z_n)$ is true. Similarly, "$\exists x \in U, P(x)$ is true iff there exists an object $b \in \{z_1, \ldots, z_n\}$ making $P(b)$ true, which holds iff $P(z_1$ is true or $P(z_2)$ is true or $\ldots$ or $P(z_n)$ is true. In summary,

$$\boxed{\forall x \in \{z_1, z_2, \ldots, z_n\}, P(x)} \iff \boxed{P(z_1) \wedge P(z_2) \wedge \cdots \wedge P(z_n)}$$
$$\boxed{\exists x \in \{z_1, z_2, \ldots, z_n\}, P(x)} \iff \boxed{P(z_1) \vee P(z_2) \vee \cdots \vee P(z_n)}$$

In the general case of an infinite universe U, "$\forall x \in U, P(x)$" is a statement that (informally speaking) AND's together the infinitely many statements $P(a)$ as a ranges through all objects in U. Similarly, "$\exists x \in U, P(x)$" is a statement that OR's together all the statement $P(b)$ for b ranging through U.

1.5 Quantifier Properties and Useful Denials

1.5.1 Conversion rules for restricted quantifiers

Let U be any set, and consider the statement $\exists x \in U, P(x)$. This statement is true iff there is an object x_0, *in* the set U for which $P(x_0)$ is true. Intuitively, this condition holds iff there is an object x_0 (not initially required to be in U) for which "$x_0 \in U and P(x_0)$ is true. Therefore, we obtain the rule

$$\boxed{\exists x \in U, P(x)} \qquad \text{iff} \qquad \boxed{\exists x, (x \in U \wedge P(x))}$$

for converting a restricted existential quantifier to an unrestricted existential quantifier.

We can also convert a restricted universal quantifier to an unrestricted universal quantifier using the following rule:

$$\boxed{\forall x \in U, P(x)} \qquad \text{iff} \qquad \boxed{\forall x, (x \in U \Rightarrow P(x))}$$

1.5.2 Translation examples

Here are some example illustrating the process of translating back and forth between English statements and logical symbolism. We use these universes and open sentences: U is the set of roses; V is the set of violets; C is the set of carrots; $R(x)$ means x is red; $P(x)$ means x is purple; $O(x)$ means x is orange; and $B(x)$ means x is beautiful.

(a) "All violets are purple". The word "all" signifies a universal statement, so we write $\forall x \in V, P(x)$. Eliminating the restricted quantifier gives $\forall x, (x \in V \Rightarrow P(x))$.

(b) "Some roses are orange.". Here, "some" indicates an existential statement, so we get $\exists x \in U, O(x)$, which in turn becomes $\exists x, (x \in U \wedge O(x))$. Note that the logical statement would be true even if there were only one orange rose, despite the use of the plural "roses" in the English statement.

(c) "Every carrot is orange or purple." We encode this as $\forall x \in C, (O(x) \vee P(x))$ or equivalently, $\forall x, (x \in C \Rightarrow (O(x) \vee P(x)))$.

(d) "There is a purple and orange carrot." We encode this as $\exists x \in C, (P(x) \wedge O(x))$ or equivalently $\exists x, (x \in C \wedge P(x) \wedge O(x))$. On the other hand, "Purple carrots exist, and orange carrots exist" is a different statement, which could be encoded as

$$(\exists x, (x \in C \wedge P(x))) \wedge (\exists y, (y \in C \wedge O(y))).$$

(e) "Any red rose is beautiful." We have not introduced a universe of red roses, but we can use unrestricted quantifiers to write $\forall x, (R(x) \wedge x \in U) \Rightarrow B(x)$. We could also say $\forall x \in U, (R(x) \Rightarrow B(x))$.

(f) "If purple carrots exist, then each white rose is beautiful." In this sentence, two quantifiers appear within an IF-THEN construction. One possible encoding is

$$[\exists x, (x \in C \wedge P(x))] \Rightarrow [\forall y, ((W(y) \wedge y \in U) \Rightarrow B(y))].$$

Note that adjective modifying a noun produce $\wedge$ symbols in the encoding, whether the noun is existentially or universally quantified.

(g) "$x + y = y + x$ for all real numbers x and y". Although English grammar allows a quantifier to appear after the variable it quantifies, formal logic requires the quantifier to come first. Thus we write $\forall x \in \mathbb{R}, \forall y \in \mathbb{R}, x + y = y + x$. Note that the word "and" in this sentence does not translate into the logical operator $\wedge$, but allows the single phrase "for all" to stand for two universal quantifiers.

(h) "$n = 2k$ for some integer k". Here too, the quantifier at the end of the English sentence must be moved to the front in the logical version, giving $\exists k \in \mathbb{Z}, n = 2k$.

(i) "Roses are red, carrots are orange." This sentence reveals an unpleasant feature of English: ***hidden quantifiers and logical operators***. The use of the plural "roses" and "carrots" implicitly indicates that these statements are intended to apply to *all* roses and *all* carrots. Moreover, the comma joining the two phrases has the same meaning as AND. Thus we write $(\forall x, (x \in U \Rightarrow R(x))) \wedge (\forall y, (y \in C \Rightarrow O(y)))$. In this case, we could get away with a single quantifier for both clauses, writing

$$\forall z, ((z \in U \Rightarrow R(z)) \wedge (z \in C \Rightarrow O(z))) \ .$$

But if $\wedge$ had been $\vee$ in both statements, the two encodings above would *not* be equivalent. The first encoding now says "All roses are red OR all carrots are orange," which is false (as white roses and purple carrots do exist!). But the second encoding (upon eliminating IF) states that every object is not a rose or is red or is not a carrot or is orange, which is true.

(j) "Each even integer greater than 2 is the sum of two primes." This sentence (called *Goldbach's Conjecture*) contains two hidden existential quantifiers signaled by the verb "is". We can encode it as follows:

$$\forall x \in \mathbb{Z}, (x \text{ is even} \wedge x > 2) \Rightarrow [\exists y \in \mathbb{Z}, (y \text{ is prime} \wedge z \text{ is prime} \wedge x = y + z)].$$

As this example illustrates, we often allows ourselves to use restricted quantifiers involving a "standard" universe (like $\mathbb{Z}$), but we apply the conversion rules to move the non-standard restrictions on the variables (like being even, or being prime) into the propositional part of the statement, rather than inventing new universes for the set of even integers large than 2 or the set of prime integers.

1.5.3 Quantifier Negation Rules

For any universe U and any open sentence $P(x)$, we have the ***quantifier negation rules***:

$$\boxed{\neg\exists x \in U, P(x)} \quad \text{iff} \quad \boxed{\forall x \in U, \neg P(x)};$$

$$\boxed{\neg\forall x \in U, P(x)} \quad \text{iff} \quad \boxed{\exists x \in U, \neg P(x)}.$$

1.5.4 Negations and Denials

Given any proposition P, recall that the *negation* of P is the proposition $\neg P$, which has the opposite truth value as P. We say that a proposition Q is a *denial* if P iff Q is logically equivalent to $\neg P$.

In general, if P is a complicated statement built from logical operators and quantifiers, then the negation $\neg P$ can be difficult to work with directly. Thus we need to develop techniques for passing from the particular denial $\neg P$ of P to another denial that *does not begin with the negation symbol*. **It turns out that the most useful denials of P are those in which *the negation symbol is not applied to any substatement* involving a logical operator or quantifier symbol.**

How can we find a *useful denial* of P? We have already derived the rules we need to simplify expressions that begin with NOT. These rules are summarized in the following table. We usually need to apply several rules recursively to convert the negation $\neg P$ into the most useful denial in which no subexpression begins with NOT.

Statement	Denial of Statement	Symbol Version of Rule
A and B	(denial of A) or (denial of B)	$\neg(A \wedge B) \equiv (\neg A) \vee (\neg B)$
A or B	(denial of A) and (denial of B)	$\neg(A \vee B) \equiv (\neg A) \wedge (\neg B)$
if A, then B	A and (denial of B)	$\neg(A \Rightarrow B) \equiv A \wedge (\neg B)$
not A	A	$\neg(\neg A) \equiv A$
For all x, $P(x)$	There is x, (denial of $P(x)$)	$\neg\forall x \in U, P(x)$ iff $\exists x \in U, \neg P(x)$
There is x, $P(x)$	For all x, (denial of $P(x)$)	$\neg\exists x \in U, P(x)$ iff $\forall x \in U, \neg P(x)$
A iff B	A or B, not both	$\neg(A \iff B) \equiv (A \oplus B)$
A or B, not both	A iff B	$\neg(A \oplus B) \equiv (A \iff B)$

1.5.5 The denial of an IF-statement is an AND-statement, not another IF-statement!

One way to understand this is to recall that "if A then B" is equivalent to "(not A) or B." Negating the latter expression produces "A and (not B)" as asserted in the table.

1.6 Denial Practice and Uniqueness Statements

1.6.1 Uniqueness

An object is *unique* iff it is the only one of its kind.

Definition: Uniqueness Symbol. Let U be a set, and let $P(x)$ be an open sentence. The statement $\boxed{\exists! x \in U, P(x)}$ means that there exists exactly one object x_0 in U for which $P(x_0)$ is true. Equivalently, there exists a unique object x_0 in U making $P(x_0)$ true. The unrestricted quantifier $\exists! x, P(x)$ is defined similarly; this statement means that there exists one and only one object x_0 making $P(x_0)$ true. The exclamation mark following the existential quantifier is called the ***uniqueness symbol***.

1.6.2 Eliminating the uniqueness symbol

It is possible to eliminate the uniqueness symbol, replacing $\exists! x, P(x)$ by an equivalent statement using previously introduced logical symbols. This elimination rule gives us insight into the precise meaning of uniqueness, and we often need the rule when giving proofs of uniqueness. To derive the rule, let us consider the logical encoding of several related statements first.

Step 1. How can we encode the following statement ? "There exist at least two objects x_0 making $P(x_0)$ true." A first attempt might be to write $\exists x, \exists y, P(x) \wedge P(y)$. However, this does not quite work, because we are allowed to pick the same object for x and for y. For instance, we see that $\exists x, \exists y, x + 1 = 3 \wedge y + 1 = 3$ is true by making $x = y = 2$. If we intend the variables x and y to represent different (distinct) objects, we must explicitly say so by making $x \neq y$. *So the given statement can be encoded as follows*:

$$\exists x, \exists y, [(P(x) \wedge P(y)) \wedge x \not= y].$$

Step 2. Now consider how to encode "There exists *at mot one* object x_0 making $P(x_0)$ true." The key is to recognize that the situation described here (having at most one object that works) is the exact logical opposite of the situation in Step 1 (having at least two objects that work). Thus we can obtain the answer by denying the statement from Step 1. One possible denial looks like this:

$$\forall x, \forall y, [\neg(P(x) \wedge P(y)) \vee x = y].$$

We could continue to simplify the denial by replacing $\neg(P(x) \wedge P(y))$ by $(\neg P(x)) \vee (\neg P(y))$. However, another way to proceed is to remember the equivalence $A \Rightarrow B \equiv (\neg A) \vee B$ from the Theorem on IF. Using this equivalence in reverse, we obtain the following IF-statement as a possible denial of the statement in Step 1:

$$\forall x, \forall y, [(P(x) \wedge P(y)) \Rightarrow x = y]. \tag{1.5}$$

Reading this in English, with some words added for emphasis, (1.5) says that "for all objects x_0 and y_0 if, $P(x_0)$ and $P(y_0)$ both happen to be true, then it must be the case that x_0 actually equals y_0." This statement does have the intended effect of preventing more than one object from satisfying the open sentence $P(x)$.

Although it already follows from the denial rules, let us confirm directly that (1.5) is true when $P(x)$ is satisfied by zero objects, or by exactly one object. First suppose no objects make $P(x)$ true. Then for any objects x_0 and y_0, $P(x_0) \wedge P(y_0)$ is false, so the IF-statement is true. Thus, (1.5) is true in this situation. Now suppose exactly one object z_0 makes $P(x)$ true. IN this case, when $x = z_0$ and $y = z_0$, the IF-statement says $(P(z_0) \wedge P(z_0)) \Rightarrow z_0 = z_0$, which is true. On the other hand, for all other choices x and y, the IF-statement has a false hypothesis, hence is true.

Step 3. We are now ready to encode the target statement. "There exists *exactly* object x_0 making $P(x_0)$ true." The key is to rewrite this assertion using AND, as follows: "There exists *at least one* object x_0 making $P(x_0)$ true, AND there exists *at most one* object x_0 making $P(x_0)$ true." The first clause can be handled by an ordinary existential quantifier, and we encoded the second clause in Step 2. To summarize, our *elimination rule for the uniqueness symbol is*:

$$\boxed{\exists! x, P(x)} \qquad \Longleftrightarrow \qquad \boxed{(\exists x, P(x)) \wedge (\forall x, \forall y, [(P(x) \wedge P(y)) \Rightarrow x = y])}$$

A similar rule holds for restricted quantifiers, For instance, $\exists! x \in \mathbb{R}, x^3 = 8$ can be rewritten as

$$(\exists x \in \mathbb{R}, x^3 = 8) \wedge (\forall x \in \mathbb{R}, \forall y \in \mathbb{R}, (x^3 = 8 \wedge y^3 = 8) \Rightarrow x = y).$$

We can also adapt the reasoning in Steps 1 through 3 to make statements such as "there are at least three objects x_0 making $P(x_0)$ true," "there are at most two objects x_0 making $P(x_0)$ true," "there are exactly two objects x_0 making $P(x_0)$ true," and so on. We also mention that a statement such as, "the solution is unique, if it exists" is asserting the existence of *at most one* solution, hence could be encoded by the formula (1.5).