LOGIC

BY PANKAJ KAMTHAN

1. INTRODUCTION

A good designer must rely on experience, on precise, logic thinking; and on pedantic exactness.
— Niklaus Wirth

It is not what was taught in the mathematics class that was important; it's the fact that it was mathematical.
— Keith Devlin

A **discrete structure** is a **mathematical model** of a **discrete object**. The properties of discrete structures are expressed using mathematical statements. Logic, specifically, **mathematical logic**, is a **formal** basis of mathematical statements. Indeed, logic plays such an important role in mathematics that it has been called as the **"language of mathematics"** [Roberts, 2010, Chapter 1]. It would therefore not be an overstatement to say that mathematical writing is logical writing.

There are several definitions of logic, one of which is the following:

Definition [Logic] [Roberts, 2010, Chapter 1]. The study of (correct and incorrect) reasoning.

Logic has its origins in philosophy [Swart, 2018]. There are many kinds of logic, including combinatory logic, description logic, modal logic, symbolic logic, and temporal logic. The two primary areas of **symbolic logic** are propositional logic and predicate logic.

This document provides an elementary introduction to propositional logic and predicate logic. In particular, it suggests a **channel** between **statements in mathematics and statements in natural languages (such as English)**, and a channel between **statements in mathematics and statements in computer languages (such as controlled languages, modeling languages, and programming languages)**.

2. THE SIGNIFICANCE OF LOGIC TO COMPUTER SCIENCE AND SOFTWARE ENGINEERING

The logic of mathematics and the logic of programming are similar, and improving skills in one will help the other. The beauty of a proof is similar to the beauty of a program.
— Marty Lewinter and Jeanine Meyer

It could be said that logic is the basis for **mathematical reasoning** as well as **computational reasoning**, and is considered significant to the science and engineering of all **computer hardware and software**.

Therefore, logic continues to be a subject of active study [Huth, Ryan, 2004; Parnas, Chik-Parnas, 2005; O'Donnell, Hall, Page, 2006, Chapter 6, Chapter 7; Vatsa, Vatsa, 2009, Chapter 1; Pratt, 2010, Section 7.3; Roberts, 2010, Chapter 1; Belcastro, 2012, Chapter 2; Ben-Ari, 2012; Cunningham, 2012, Chapter 2, Chapter 3; Gerstein, 2012, Chapter 1; Makinson, 2012, Chapter 8, Chapter 9; Pace, 2012, Chapter 2, Chapter 3; Rosen, 2012, Chapter 1; Genesereth, Kao, 2013; Hammack, 2013, Chapter 2; Moller, Struth, 2013, Chapter 1, Chapter 4; O'Regan, 2013, Chapter 3; Scheinerman, 2013; O'Regan, 2016, Chapter 15; Levin, 2016, Chapter 3; Genesereth, Kao, 2017; Hart, Sandefur, 2018; Kurgalin, Borzunov, 2018, Chapter 1; Swart, 2018].

2.1. USES OF LOGIC: "THE TRUTH IS IN HERE"

In computer science, software engineering, and cognate disciplines, logic is **ubiquitous**, as indicated by the following applications [Bollobas, 1990; O'Donnell, Hall, Page, 2006, Chapter 6, Chapter 7; Haan, Koppelaars, 2007; Rosen, 2012; Giordano, Fox, Horton, 2014, Chapter 8; Jorgensen, 2014, Chapter 3; Brath, Jonker, 2015; Vince, 2015]:

- For **instilling critical thinking** by improving **mathematical aptitude** [Houston, 2009; Bassham, Irwin, Nardone, Wallace, 2011].

- For **formulating technical arguments and supporting assertions** [Howard-Snyder, Howard-Snyder, Wasserman, 2009; Govier, 2014].

- For **providing means for clear and concise technical writing** (that, at least in computer science and software engineering, is often expressed in English) and **mathematical writing** [Krantz, 1998]. (Indeed, such clarity and concision is necessary for the **specifications, guides, and manuals** of many software systems [Dooley, 2011, Chapter 4; Wallwork, 2014].)

- For **proving mathematical statements** [Cunningham, 2012, Preface].

- For **specifying hardware or software systems** [O'Donnell, Hall, Page, 2006, Chapter 6; Rosen, 2012, Chapter 1].

- For **expressing software requirements** [Berry, Kamsties, 2005].

- For **designing electronic circuits** [Rosen, 2012, Chapter 1; Reba, Shier, 2015].

- For **developing computer algorithms** and proving that they are correct [Cormen, Leiserson, Rivest, Stein, 2009].

- For **retrieving information** by deciding the properties that should be present in information in order for it to be considered relevant [O'Donnell, Hall, Page, 2006, Chapter 6].

- For **providing a scientific basis of programming and programming languages** [Sebesta, 2012], such as using lambda calculus for specifying the meaning of a computer program [O'Donnell, Hall, Page, 2006, Chapter 6].

- For **appreciating the basis for logic programming languages** [O'Donnell, Hall, Page, 2006, Chapter 6; Sebesta, 2012].

- For **constructing a compiler**, specifically, **type checking** [O'Donnell, Hall, Page, 2006, Chapter 6].

- For **software testing**, specifically, **discovering errors in conditional expressions, and constructing decision tables and equivalence classes of test cases** [Ammann, Offutt, 2008, Chapter 3; Naik, Tripathy, 2008; Jorgensen, 2014, Chapter 3; Roman, 2018].

2.2. THE RELATIONSHIP OF LOGIC TO COMPUTER SCIENCE

In [Baldwin, Walker, Henderson, 2013], the role of mathematics in computer science is explored. The relationship of logic to computer science is shown in Figure 1.

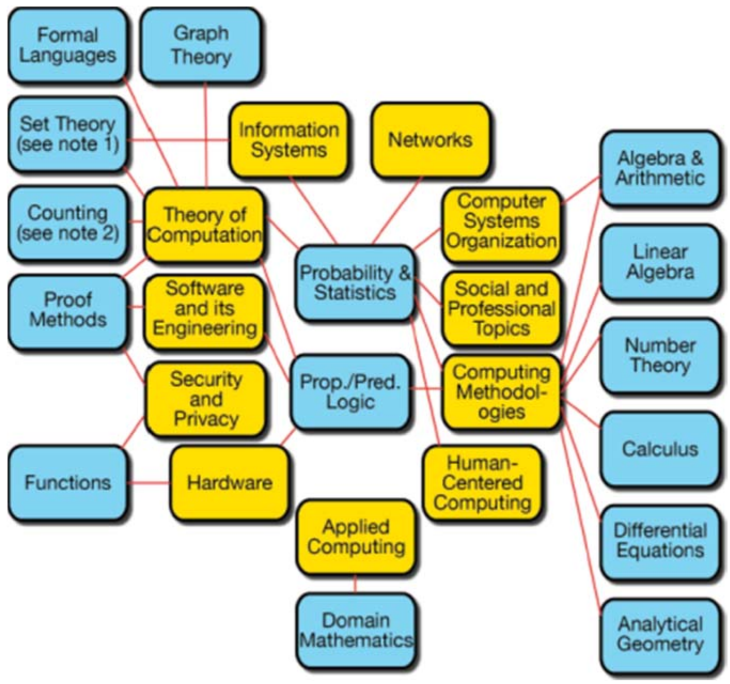

Figure 1. The areas in mathematics associated with areas in computer science. (Source: [Baldwin, Walker, Henderson, 2013].)

2.3. THE RELATIONSHIP OF LOGIC TO SOFTWARE ENGINEERING

SWEBOK®

The **Guide to the Software Engineering Body of Knowledge (SWEBOK)** “describes the sum of knowledge within the profession of software engineering” [IEEE, 2014]. In SWEBOK, there are a number of Knowledge Areas (KAs). The study of logic is part of **Mathematical Foundations KA** of the SWEBOK3, as shown in Figure 2.

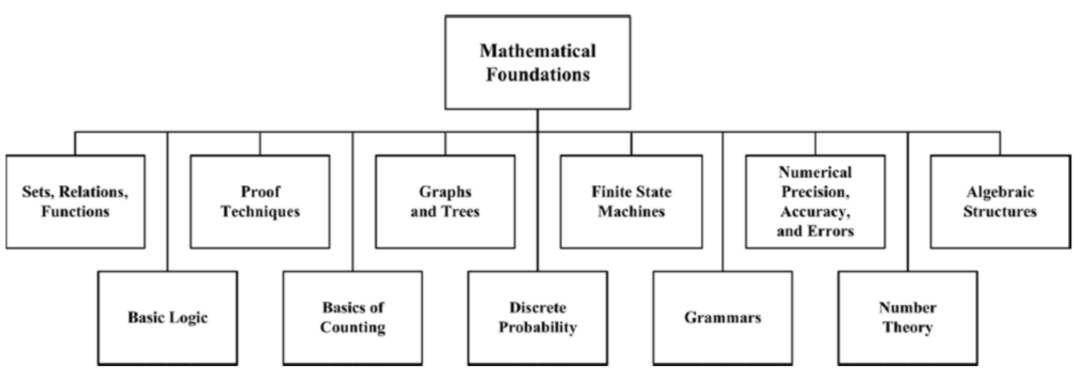

Figure 2. Mathematical Foundations is a Knowledge Area in the Guide to the Software Engineering Body of Knowledge (SWEBOK). (Source: SWEBOK [IEEE, 2014].)

3. STATEMENTS

An **axiom** is a logical statement that is evident or assumed to be true [Roberts, 2010, Chapter 1]. It serves as a starting point from which other statements are logically derived.

Axiom [Truth Axiom]. True is opposite of false.

In linguistics, sentences are classified according to their usage [Mclnerny, 2004; Roberts, 2010, Chapter 1]. A **declarative or evaluative sentence** makes a statement. An **imperative sentence** gives a command or makes a request. An **interrogative sentence** asks a question. An **exclamatory sentence** expresses strong feeling.

Definition [Statement]. A sentence with a definite conclusion.

REMARKS

A sentence is a **grammatical entity**, whereas a statement is a **logical entity**.

EXAMPLE

$1^{232} < 2$.

- Linguistic Structure: Statement
- Truth Value: True

EXAMPLE

Montreal is in New York State.

- Linguistic Structure: Statement
- Truth Value: False

EXAMPLE

The system is reliable, is it not? (or "Is the Earth round?", or "What is the meaning of life?")

- Linguistic Structure: Question
- Truth Value: True? False?

It is an **interrogative sentence**, but **not** a statement.

EXAMPLE

If only the moon were flat! (or “Eat!”, or “Help!”)

- Linguistic Structure: Exclamation
- Truth Value: True? False?

It is an **exclamatory sentence**, but **not** a statement.

EXAMPLE

Press $\boxed{1}$ if you want to press $\boxed{2}$.

- Linguistic Structure: Command
- Truth Value: True? False?

It is an **imperative sentence**, but **not** a statement.

EXAMPLE

$\{0, 1\} \setminus \{-1, 1\}$.

- Linguistic Structure: Expression
- Truth Value: True? False?

It is a mathematical expression, **not** a statement, as it lacks verb.

EXAMPLE

It is sunny today.

- Linguistic Structure: Sentence
- Truth Value: True? False?

This sentence is **neither true nor false** as the answer **depends** on **context**, specifically, space (the geographic location) and time (of the day).

EXAMPLE

Paradox is a thought construction, which leads to an unexpected contradiction.
— Piotr Łukowski

I am lying to you.

- Linguistic Structure: Statement
- Truth Value: True? False?

Let S: I am lying to you.

There are two possibilities:

1. S is True $\Longrightarrow$ I am lying to you. $\Longrightarrow$ S is False.
2. S is False $\Longrightarrow$ I am not lying to you. $\Longrightarrow$ S is True.

Therefore, S is **both** true and false. This is **impossible** by the Truth Axiom.

REMARKS

- This is an example of a **paradox**. Indeed, this is known as the **Liar's Paradox** [Mendelson, 2015].

  A paradox is a kind of **contradiction**. In a **technical discourse**, including **mathematical arguments**, the goal is to **avoid contradictions**.

  There are many examples of paradoxes [Bunch, 1982; Łukowski, 2011; Clark, 2012; Cook, 2013]. For example, Figure 3 shows a **geometrical paradox**.

Figure 3. Impossible Cube. (Source: Wikipedia.)

- This example shows that notion of **statement** alone is **insufficient** as foundation of logic, and further **restrictions** on the definition of statement are necessary.

4. PROPOSITIONS

Definition [Proposition]. A statement that is either true or false, but not both.

4.1. CHARACTERISTICS OF PROPOSITIONS

There are a number of defining characteristics of a proposition:

- A proposition is **declarative** (that is, a statement that declares a fact).

- A person may not know whether or not a given proposition is true, but it must have a **definite value**, namely either true or false.

- An **opinion** or a **paradox** is **not** a proposition.

EXAMPLE

$1^{420} < 2$.

- Linguistic Structure: Proposition
- Truth Value: True

EXAMPLE

New York State is in Montreal.

- Linguistic Structure: Proposition
- Truth Value: False

EXAMPLE

It is cloudy today.

- Linguistic Structure: Not a Proposition

EXAMPLE

Is the system usable?

- Linguistic Structure: Not a Proposition

EXAMPLE

$(0, 1] \setminus \{0, 1\}$

- Linguistic Structure: Not a Proposition

EXAMPLE

$x^{666} + 1 = 2$.

- Linguistic Structure: Not a Proposition

This sentence is **neither true nor false**, as the answer **depends** on the value of x.

EXAMPLE

This sentence is false.

- Linguistic Structure: Not a Proposition

5. PROPOSITIONAL LOGIC

The **semantic content of a statement** means that the only important property of that statement is its **truth value**, and it cannot have any further meaning other than whether it is true or false.

From the perspective of computer science, propositional logic is a **static discipline** of **statements that lack semantic content**. For example, a proposition may be **purely mathematical** in nature.

EXAMPLE

p: 1729 = 1725 + 2.

This translates in English to "one thousand seven hundred twenty nine is the sum of one thousand seven hundred twenty five and two".

The proposition is obviously false, but apart from that, nothing further (such as, the **significance** of the number **one thousand seven hundred twenty nine**[1]) can be inferred. In other words, **a proposition does not tell a reader whether a number has any interesting properties or not** [Posamentier, Thaller, 2015].

5.1. COMPOUND PROPOSITIONS

The expressive capability of simple propositions is somewhat **limited**. An important topic in propositional logic is the study of **how** simple propositions can interact to produce **complex propositions**.

It is assumed that a collection of **"atomic"** propositions is given. These propositions can be denoted by: p, q, r, … A **compound proposition** is formed by using **logical connectives (specifically, logical operators)** to form propositional **"molecules"**.

REMARKS

As an analogy, **compound sentences in English** make for **sophisticated writing**. For example:

> If you keep thinking about what you want to do or what you hope will happen, you don't do it, and it won't happen.

As another analogy, **compound expressions in mathematics** make for **profound expressions**. For example:

$$w(x) = \prod_{i=1}^{20}(x - i) = (x-1)(x-2)\cdots(x-20).$$

(This is known as the **Perfidious Polynomial or Wilkinson's Polynomial**.)

As yet another analogy, **psychoactive drugs** make for **interesting chemical compounds**, as, for example, shown in Figure 4.

[1] URL: http://mathworld.wolfram.com/1729.html .

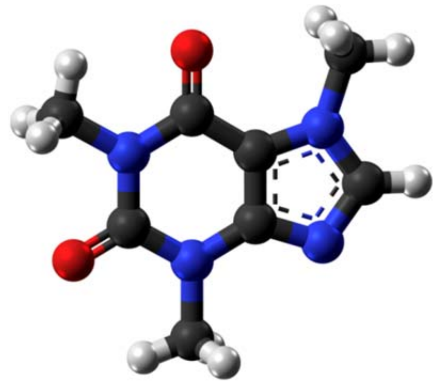

Figure 4. A three-dimensional rendering of the caffeine molecule. (Source: Wikipedia.)

6. LOGICAL CONNECTIVES

A **logical operator** (or, equivalently, a logical connective) ‘operates’ on an operand. An operand is a proposition. Table 1(a) includes the specifics of some common **basic logical operators**.

Operator	**Symbol**	**Usage**	**Precedence**	**Java**
Negation	$\neg$	NOT	1	`!`
Conjunction	$\wedge$	AND	2	`&&`
Disjunction	$\vee$	OR	3	`||`
Exclusive-Or	$\oplus$	XOR	3	`(p || q) && (!p || !q)`
Conditional	$\longrightarrow$	if, then	4	`p?q:true`
Biconditional	$\longleftrightarrow$	iff	5	`(p && q) || (!p && !q)`

Table 1(a). A collection of notations, semantics, use, and precedence of some basic logical operators.

Table 1(b) includes the specifics of some not-so-common **basic logical operators**.

Operator	Symbol	Usage	Meaning
Negated Conjunction	$\uparrow$	NAND	$\neg(p \wedge q) = \neg p \vee \neg q$
Negated Disjunction	$\downarrow$	NOR	$\neg(p \vee q) = \neg p \wedge \neg q$
Negated Conditional	$\nrightarrow$		$\neg(p \longrightarrow q)$

Table 1(b). A collection of notations, semantics, and use of some not-so-common basic logical operators.

REMARKS

It is common for propositions and logical connectives to occur in computer programs [Lehman, Leighton, Meyer, 2012, Section 3.2].

EXAMPLE

For example, consider the following Java source code fragment:

```
if !(x == y) {
  // ...
}
```

EXAMPLE

For example, consider the following Java source code fragment:

```
if ( x > 0 || (x <= 0 && y > 100) )
```

This expression is composed of two **simpler** propositions.

Let p be the proposition that `x > 0`, and let q be the proposition that `y > 100`. Then, the condition can be expressed as

$$p \vee (\neg p \wedge q).$$

The use of **parenthesis**—a **pairwise combination** of '(' and ')' (or [and])—can be made to **avoid ambiguity in compound propositions** [Smullyan, 2009, Page 48; Pace, 2012, Page 14]. (The **precedence of operators** helps, but cannot always avoid ambiguity, especially if the operators have the **same precedence**.)

EXAMPLE

The following propositions are given:

> p : Complex software is built only by large companies.
> q : Complex software is built by IBM.
> r : IBM is a large company.

Then,

> $\neg r$: IBM is not a large company.
>
> $p \wedge q$: Complex software is built only by large companies and by IBM.
>
> $p \wedge q \longrightarrow r$: If complex software is built only by large companies and by IBM, then IBM is a large company.

7. TRUTH TABLES

A **natural language description** of the conditions under which a certain truth value of a proposition occurs can get **verbose**, perhaps even **error-prone**.

Every proposition (simple or compound) can take only one of the **two truth values**. Therefore, the relationships at each level of operation can be represented in **two dimensions**, that is, in a **table**.

Definition [Truth Table]. A table displaying **relationships** between the truth values of a proposition.

The logical operators are defined by truth tables. A truth table gives the truth value of the **proposition under study** (output of the operation) in the **right-most column**.

NOTATION

In a truth table, or even otherwise, True is often abbreviated as **"T"**, and False is often abbreviated as **"F"**.

7.1. CONSTRUCTION OF A TRUTH TABLE

The steps for construction of a truth table:

1. For each **simple proposition**, allocate **one column**.
2. For each **compound proposition**, allocate **one column**.
3. The truth value of the proposition under study is given by the **right-most column**.

The **order of the propositions** in the columns should preferably match the propositions in given textual or mathematical notation. It could be noted that for n propositions there will be 2^n rows.

8. NEGATION

Negation turns a true proposition to false, and a false proposition to true. Let *p* be a proposition. If *p* is true, then $\neg p$ is false.

Negation is a **unary operator**. It operates on **one** proposition when creating a compound proposition.

The truth table for negation:

p	$\neg p$
T	F
F	T

EXAMPLE

p: 25 = 17 + 7
p is false, so $\neg p$ is true.

REMARKS

In the design of electronic circuits, **basic circuits** are called **gates**. These basic circuits could be used for the design of more sophisticated circuits. Negation is a basic circuit, as shown in Figure 5, and is called an inverter.

Figure 5. An inverter. ([Rosen, 2012, Chapter 1].)

9. CONJUNCTION

Let p and q be two propositions. For $p \wedge q$ to be true, it must be the case that **both** p is true, as well as q is true. If **one** of these is false, then the compound statement is false as well.

Conjunction is supposed to encapsulate what happens when people use the word “and” in English.

Conjunction is a **binary operator**. It operates on two propositions when creating a compound proposition.

The truth table for conjunction:

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

OBSERVATIONS

- If p is false, then the truth value of q does not affect the truth value of $p \wedge q$.
- If q is false, then the truth value of p does not affect the truth value of $p \wedge q$.

EXAMPLE

p: The number of real numbers is greater than the number of natural numbers.
q: Nazlie is a trillionaire.
r: The number of resources indexed by Google Search are less than the total number of natural numbers.

Out of $p \wedge q$, $p \wedge r$, $q \wedge r$, only $p \wedge r$ is true.

EXAMPLE

Figure 6 shows a scenario where there is a **difference** between “and” as in English and “and” as in logic.

Figure 6. The transliteration of the text from Polish to English is [Roman, 2018, Section 7.4.1]: We do not sell alcohol to people who are **under the age of 18 and intoxicated**.

REMARKS

Conjunction is a basic circuit, as shown in Figure 7, and is called an AND gate.

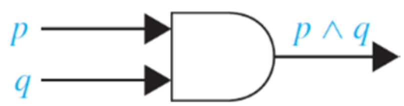

Figure 7. An AND gate. ([Rosen, 2012, Chapter 1].)

The negation of an AND is OR, not DNA.

10. DISJUNCTION

Let p and q be two propositions. For $p \vee q$ to be true, it must be the case that **one** of p or q is true. If **both** of these are false, then the compound statement is false as well.

Disjunction is also known as **inclusive-or**.

The truth table for disjunction:

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

OBSERVATIONS

- If p is true, then the truth value of q does not affect the truth value of $p \vee q$.

- If q is true, then the truth value of p does not affect the truth value of $p \vee q$.

REMARKS

In daily language, a disjunction can have two forms [Bramanti, Travaglini, 2018, Section 3.2]: **inclusive-or or exclusive-or**.

EXAMPLE

Let a restaurant serve multi-course meals, and offer the following options as part of its hors d'oeuvres:

- Antipasto
- Bruschetta
- Carpaccio

Then, it is most likely that the **options are exclusive**, meaning it is possible for a restaurant patron to have one and only one of the options.

EXAMPLE

Let a restaurant serve pizzas, and offer the following options as part of its toppings:

- Anchovies
- Olives
- Red Peppers

Then, it is most likely that the **options are inclusive**, meaning it is possible for a restaurant patron to have one or more of the options.

EXAMPLE

Disjunction is a basic circuit, as shown in Figure 8, and is called an OR gate.

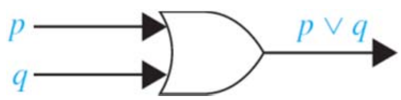

Figure 8. An OR gate. ([Rosen, 2012, Chapter 1].)

Using basic circuits it is possible to design sophisticated circuits, as shown in Figure 9.

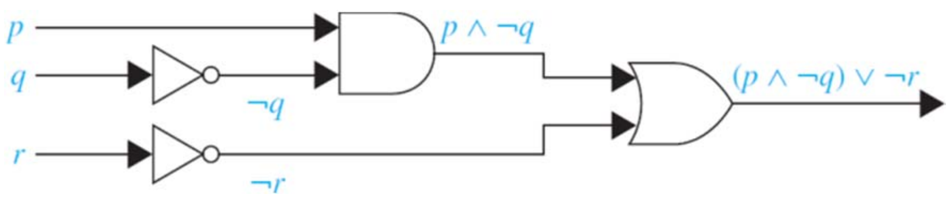

Figure 9. A combinatorial circuit. ([Rosen, 2012, Chapter 1].)

REMARKS

The English version of disjunction "or" **does not** always satisfies the assumption that one of *p*/*q* being true implies that "*p* or *q*" is true. To address this situation, **exclusive-or** is introduced.

11. EXCLUSIVE-OR

Let p and q be two propositions. For $p \oplus q$ to be true, it must be the case that **exactly one** of p and q is **true**. If **both** p and q are true, or if **both** p and q are **false**, then the compound statement is **false** as well.

The truth table for exclusive-or:

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

OBSERVATIONS

$\oplus$ has exactly the **same** truth table as $(p \wedge \neg q) \vee (q \wedge \neg p)$.

EXAMPLE

p: The airline provides the choice of vegetarian **or** non-vegetarian food.

In general, the airlines do **not** allow a passenger to get **both** vegetarian and non-vegetarian food. Therefore, *p* is false when both vegetarian and non-vegetarian food is served. The "or" in *p* is in fact "exclusive-or" (abbreviated, sometimes, as "xor").

EXAMPLE

p: The entrée is served with soup **or** salad.

In general, most restaurants do **not** allow the customer to get **both** soup and salad. Therefore, *p* is false when both soup and salad are served. The "or" in *p* is in fact "xor".

REMARKS

The previous examples are **'weak'**. This is because, in the real world, it is possible for a passenger to not eat or a customer to decline or skip an entrée.

EXAMPLE

p: The integer is even **or** odd.

If an integer is even, it cannot be odd. If an integer is odd, it cannot be even. It is **not** possible for an integer to be both even and odd. It is **not** possible for an integer to be neither even, nor odd. This could a **perfect dichotomy**.

EXAMPLE

There is support for logic in modeling languages. The following is the case of the ⊕ operator and the **Unified Modeling Language (UML)**.

In UML, to express that an object of class A is to be associated with an object of class B, or an object of class C, but not with both, an **`xor`** constraint (**exclusive-or**) can be used. To indicate that two associations from the same class are **mutually exclusive**, they can be connected by a dashed line labeled **`{xor}`**.

For example, an **examination** can take place either in an **office**, or in a **lecture hall**, but **not** in both, as shown in Figure 10.

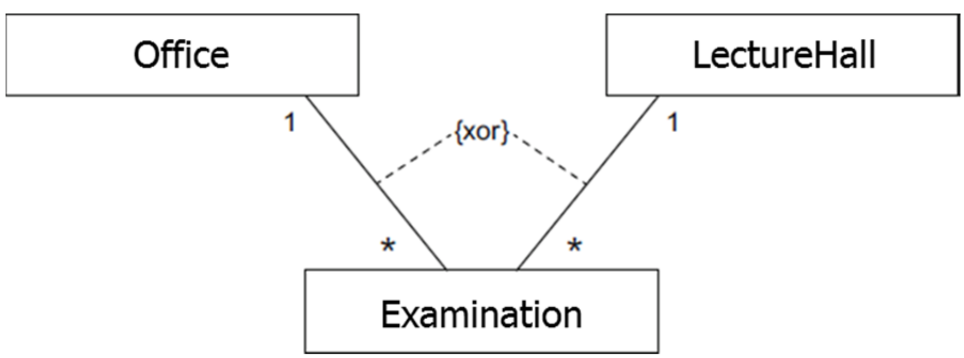

Figure 10. An illustration of the **`xor`** constraint in UML. (Source: [Seidl, Scholz, Huemer, Kappel, 2015, Section 4.3.1].)

12. IMPLICATION (CONDITIONAL)

The proposition $p \longrightarrow q$ is true (1) if both p and q are true, (2) if p is false, and (3) if q is true.

The proposition $p \longrightarrow q$ is false if p is true, but q is false.

In the proposition $p \longrightarrow q$, p is known as the **antecedent**, and q is known as the **consequent**.

The truth table for implication (conditional):

p	q	$p \longrightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

OBSERVATIONS

- If p is false, then the truth value of q does not affect the truth value of $p \longrightarrow q$.
- If q is true, then the truth value of p does not affect the truth value of $p \longrightarrow q$.

EXAMPLE

The following propositions are given:

p: I am going to town.
q: It is raining.

Then,

$p \longrightarrow q$: If I am going to town, then it is raining.

EXAMPLE

The following propositions are given:

> *p*: a > b.
> *q*: c = b – a.

Then,

> $p \longrightarrow q$: if a > b, then c = b – a.

NOTATION

There are **different ways** to express the conditional statement $p \longrightarrow q$. They all have the **same** connotation.

If *p*, then *q*.
p* implies *q.
If *p*, *q*.
***p* only if *q*.**
p* is sufficient for *q.

There are some ways that **reverse** the order of *p* and *q*.

q if *p*.
q whenever *p*.
q* is necessary for *p.

Thus, **necessity and sufficiency** are **inverse relationships**.

The different notations for the same concept provide flexibility. However, in general, it is important to be **consistent** in the use of notations: if a particular notation is used in a document, then the **same** notation should be used **throughout** that document. This is especially important in a team environment where the same document may be read by different people, perhaps in different locations.

To aid **readability**, the term “is true” can be inserted **after** each variable.

EXAMPLE

p is true only if *q* is true.
q is true if *p* is true.

REMARKS

The reason $p \longrightarrow q$ is true when both *p* and *q* are false is that the implication (conditional) mathematical construct **does not** attempt to mimic **English** [O'Donnell, Hall, Page, 2006, Section 6.2.1]. Indeed, mathematically, *p* should imply *q* whenever it is possible to derive *q* from *p* by using **valid** arguments.

EXAMPLE

It can be shown that **"if 0 = 1, then 2 = 7"** is true mathematically and by the truth table.

The following is a **mathematical proof**:

(a) 0 = 1 (This is by assumption.)
(b) 1 = 2 (This is by adding 1 to both sides.)
(c) 2 = 4 (This is by multiplying both sides by 2.)
(d) 0 = 3 (This is by multiplying (a) by 3.)
(e) 2 = 7 (This is by adding (c) and (d).)

EXAMPLE

The implication can be used for construct **inheritance hierarchies** [Gonzalez-Perez, 2018, Chapter 4].

For instance, the following propositions could be used to construct a UML Class Diagram, showing concepts as classes and the relationships between them as generalization relationships:

If *x* is a shantytown house, then *x* is also a house.
If *x* is an inhabited house, then *x* is also a home.
If *x* is an inhabited house, then *x* is also a house.
If *x* is a building, then *x* is also a structure.
If *x* is a house, then *x* is also a building.

Board Time!
Construct a UML Class Diagram for the collection of statements given previously.

12.1. A COMPARISON OF "IMPLICATION" IN LOGIC, ENGLISH, AND PROGRAMMING

The proposition $p \longrightarrow q$ has a **distant similarity** to "implies" in English and "if, then" in programming.

In English, a **correlation** between **the antecedent and the consequent** is **relevant**. For example, the antecedent **"Einstein ate apple pie"** is **not relevant** to the consequent **"Pankaj likes accessibility engineering"**.

In programming, a conditional statement "if *p*, then *S*" means a proposition *p* and a source code fragment *S* is executed). Upon the execution of a program that contains such a conditional statement, *S* is executed if *p* is true, but *S* is **not** executed if *p* is false.

However, in logic, conditional statements are **more general** than in English and in programming. The **mathematical conditional statement** is **independent** of the **cause-and-effect relationship** between the antecedent and the consequent [O'Donnell, Hall, Page, 2006, Section 6.2.5].

12.2. USES OF PROPOSITIONS FOR (SOFTWARE) REQUIREMENTS ENGINEERING

There are two common concerns in (software) requirements that can be addressed by logic, namely **formality and consistency**.

12.2.1. FORMALITY

> I spent part of my time writing software specifications using first-order predicate logic. [...] It brought **precision** where there was **potential ambiguity and rigor** where there was some hand-waving.
> — Ali Almossawi

Translating sentences in natural language (such as English) into logical expressions is part of **specifying software systems**.

The idea is to take software requirements expressed in natural language and to produce mathematical specifications that are precise and unambiguous [O'Donnell, Hall, Page, 2006, Section 6.1; Lover, 2008].

EXAMPLE

Translate the following English statement into a mathematical statement:

> "The airline reservation system will not allow access if the traveler is a guest user unless the traveler is a frequent flyer."

Solution.

Let p, q, and r denote

> "The airline reservation system will allow access",
>
> "The traveler is a guest user", and
>
> "The traveler is a frequent flyer", respectively.

Then, an equivalent mathematical statement is

$$(q \land \neg r) \longrightarrow \neg p.$$

EXAMPLE

Express the following software requirement using logical connectives:

> "The automated reply cannot be sent when the file system is full."

Solution. Let

> p denote "The automated reply can be sent"

and

> q denote "The file system is full."

Then $\neg p$ represents "It is not the case that the automated reply can be sent," which can also be expressed as "The automated reply cannot be sent."

Thus, the software requirement can be represented by the conditional statement

$$q \longrightarrow \neg p.$$

12.2.2. CONSISTENCY

It is crucial that a **software requirements specification** be **consistent** [Gervasi, Zowghi, 2005; ISO/IEC/IEEE, 2011; ISO/IEC/IEEE, 2018; Laplante, 2018, Chapter 6], that is, it does not contain **conflicting** software requirements that could be used to derive a **contradiction**.

From the perspective of propositional logic, this means the following: given any collection of software requirements in the software requirements specification, **each** software requirement in that collection must be **true**. (In other words, software requirements specification should form a **satisfiable theory**. A **theory** is a set of sentences. A theory is **satisfiable** when it is possible to present an interpretation in which **all of its sentences are true**.)

An **obsolete software requirement** is a software requirement (implemented or not) that is no longer required for the current release or future releases and has little or no business value for the potential customers or users of a software product [Wnuka, Gorschek, Zahda, 2013]. It has been pointed out in an empirical study that **change in software requirements can be unusually expensive**, and that **inconsistent software requirements are among those that are prone to change** the most [Wnuka, Gorschek, Zahda, 2013].

EXAMPLE

Determine whether the following software requirements that belong to a software requirements specification are consistent:

> "The diagnostic message is stored in the buffer or it is retransmitted."
> "The diagnostic message is not stored in the buffer."
> "If the diagnostic message is stored in the buffer, then it is retransmitted."

Solution. Let

p denote "The diagnostic message is stored in the buffer"

and

q denote "The diagnostic message is retransmitted."

The software requirements can then be expressed as the logical expressions

$p \vee q$,
$\neg p$, and
$p \longrightarrow q$, respectively.

Then, an assignment of truth values that makes all three requirements true must have p false to make $\neg p$ true. Now, as the aim is to have $p \vee q$ to be true but p must be false, q must be true. Furthermore, as $p \longrightarrow q$ is true when p is false and q is true, it can be concluded that these requirements are **consistent**, because they are all true when p is false and q is true.

EXAMPLE

Determine whether the following software requirements that belong to a software requirements specification are consistent:

"The router can send packets to the edge system only if it supports the new address space."
"For the router to support the new address space it is necessary that the latest software release be installed."
"The router can send packets to the edge system if the latest software release is installed."
"The router does not support the new address space."

Solution. Let

p: "The router can send packets to the edge system."
q: "The router supports the new address space."
r: "The latest software release is installed."

The software requirements can then be expressed as the logical expressions

$p \longrightarrow q$,
$q \longrightarrow r$,
$r \longrightarrow p$, and
$\neg q$, respectively.

It is given that q is false. Therefore, from the first expression, p must be false. From this and the third expression, it follows that r must be false. Thus, **if** in fact all three propositions are false, then all four software requirements are true, and so they are consistent.

REMARKS

- In the previous example, the term **'supports'**, unless defined, is vague. Therefore, the previous example shows a **difference between vagueness and consistency**.

- It is possible for multiple requirements to be **vague individually**, but be **consistent collectively**. However, it may be difficult to prove consistency of requirements that are vague.

EXAMPLE

Determine whether the following software requirements that belong to a software requirements specification are consistent:

"The system is in multiuser state if and only if it is operating normally."
"If the system is operating normally, the kernel is functioning."
"The kernel is not functioning or the system is in interrupt mode."
"If the system is not in multiuser state, then it is in interrupt mode."
"The system is not in interrupt mode."

Solution. Let

p: "The system is in multiuser state."
q: "The system is operating normally."
r: "The system is in interrupt mode."
s: "The kernel is functioning."

The software requirements can then be expressed as the logical expressions

$p \leftrightarrow q$,
$q \longrightarrow s$,
$\neg s \vee r$,
$\neg p \longrightarrow r$, and
$\neg r$, respectively.

It is given that r is false. Therefore:

- It follows from the fourth expression that $\neg p$ must be false, that is, p must be true. From this and the first expression, it follows that q must be true.

- It follows from the third expression, namely $s \longrightarrow r$, that s must be false. From this and the second expression, it follows that q must be false.

This is a **contradiction**, and so the software requirements are **not** consistent.

REMARKS

- The previous example shows a **difference between correctness and consistency**. It is possible for multiple requirements to be **correct individually**, but be **inconsistent collectively**.

- In the previous example, the term **'normally'**, unless defined, is vague. Therefore, the previous example shows a difference between vagueness and inconsistency.

- It could be noted that checking for consistency is a **problem of scale**. As the number of logical statements increase, the number of combinations to check also increases, and increases **exponentially**, making it **intractable for computers, sooner or later**. If there are n logical statements, then the problem belongs to $O(2^n)$. This is the **Boolean Satisfiability Problem**, an **NP-Complete Problem**.

13. CONVERSE OF IMPLICATION

The converse of implication is the converse of $p \longrightarrow q$, that is, $q \longrightarrow p$.

The truth table for converse of implication:

p	q	$q \longrightarrow p$
T	T	T
T	F	T
F	T	F
F	F	T

OBSERVATIONS

- If p is true, then the truth value of q does not affect the truth value of $q \longrightarrow p$.
- If q is false, then the truth value of p does not affect the truth value of $q \longrightarrow p$.
- The implication and the converse of implication **cannot both be false simultaneously**.

This can be shown by a comparison of their truth tables:

p	q	$p \longrightarrow q$	$q \longrightarrow p$
T	T	T	T
T	F	F	T
F	T	T	F
F	F	T	T

EXAMPLE

The following propositions are given:

p: I am going to town.
q: It is raining.

Then,

$p \longrightarrow q$: If I am going to town, then it is raining.
$q \longrightarrow p$: If it is raining, then I am going to town.

14. CONTRAPOSITIVE OF IMPLICATION

The contrapositive of implication is the contrapositive of $p \longrightarrow q$, that is, $\neg q \longrightarrow \neg p$.

The truth table for contrapositive of implication:

p	q	$\neg q$	$\neg p$	$\neg q \longrightarrow \neg p$
T	T	F	F	T
T	F	T	F	F
F	T	F	T	T
F	F	T	T	T

REMARKS

The contrapositive of implication has exactly the **same** truth table as implication.

EXAMPLE

The following propositions are given:

> *p*: I am going to town.
> *q*: It is raining.

Then,

> $p \longrightarrow q$: If I am going to town, then it is raining.
> $\neg q \longrightarrow \neg p$: If it is not raining, then I am not going to town.

EXAMPLE

This is an instance of the so-called **Wason Selection Task**, a puzzle in **deductive reasoning** [Roman, 2018, Section 2.3.4], shown in Figure 11.

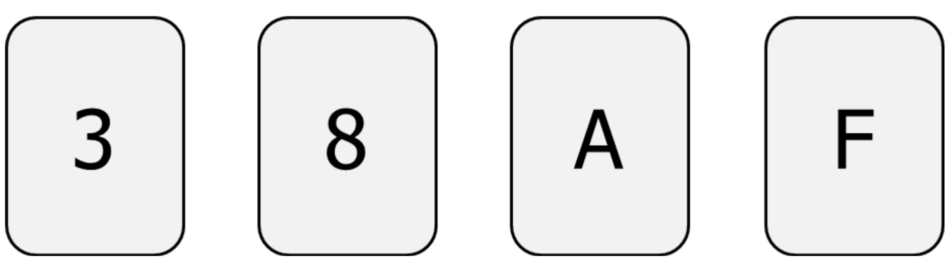

Figure 11. The four cards in the Wason Selection Task.

There are four cards placed on a table, each of which had a number on one side and a letter on the other side. Let there be the following proposition: **if a card shows an even number on one face, then there is a vowel on the other face**. The question is which cards must be **necessarily** turned over in order to test the truth value of the proposition.

The contrapositive of implication could be used to solve the Wason Selection Task.

The faces of card could be called face 1 and face 2.

Let

p: There is an even number on face 1 of card.
q: There is a vowel on face 2 of card.

Therefore,

$\neg p$: There is not an even number on face 1 of card.
$\neg q$: There is not a vowel on face 2 of card.

It could be observed from the truth table for contrapositive of implication that

$\neg q \longrightarrow \neg p$ is **always** T if $\neg q$ is F or if $\neg p$ is T.

Therefore, one does not need turn over the card labelled "A" and the card labelled "3". In other words, the **only** cards that need to be turned over are the ones labelled "8" and "F".

15. INVERSE OF IMPLICATION

The inverse of implication is the inverse of $p \longrightarrow q$, that is, $\neg p \longrightarrow \neg q$.

The truth table for inverse of implication:

p	q	$\neg p$	$\neg q$	$\neg p \longrightarrow \neg q$
T	T	F	F	T
T	F	F	T	T
F	T	T	F	F
F	F	T	T	T

OBSERVATIONS

- If p is true, then the truth value of q does not affect the truth value of $\neg p \longrightarrow \neg q$.
- If q is false, then the truth value of p does not affect the truth value of $\neg p \longrightarrow \neg q$.

EXAMPLE

The following propositions are given:

p: I am going to town.
q: It is raining.

Then,

$p \longrightarrow q$: If I am going to town, then it is raining.
$\neg p \longrightarrow \neg q$: If I am not going to town, then it is not raining.

EXAMPLE

Give the converse, the contrapositive, and the inverse of these conditional statements.

(a) If $|x| = x$, then $x \geq 0$.
(b) If n is greater than 3, then n^2 is greater than 9.

Solution.

(a)

The converse is "If $x \geq 0$, then $|x| = x$."
The contrapositive is "If $x < 0$, then $|x| \neq x$."
The inverse is "If $|x| \neq x$, then $x < 0$."

(b)

The converse is "If n^2 is greater than 9, then n is greater than 3."
The contrapositive is "If n^2 is not greater than 9, then n is not greater than 3."
The inverse is "If n is not greater than 3, then n^2 is not greater than 9."

16. BICONDITIONAL

The proposition $p \leftrightarrow q$ is true, when p and q have the **same** truth value; otherwise, $p \leftrightarrow q$ is false.

The truth table of biconditional:

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

OBSERVATIONS

$\leftrightarrow$ has exactly the **same** truth table as $(p \longrightarrow q) \wedge (q \longrightarrow p)$, and exactly the **opposite** truth table of $\oplus$.

NOTATION

There are **different ways** to express the biconditional statement $p \leftrightarrow q$. They all have the **same** connotation.

p if and only if q.
p iff q.
p* is necessary and sufficient for *q.

EXAMPLE

Construct a truth table for $[p \wedge (\neg(\neg p \vee q))] \vee (p \wedge q)$.

Solution.

p	q	$\neg p$	$\neg p \vee q$	$\neg(\neg p \vee q)$	$p \wedge (\neg(\neg p \vee q))$	$p \wedge q$	$[p \wedge (\neg(\neg p \vee q))] \vee (p \wedge q)$
T	T	F	T	F	F	T	T
T	F	F	F	T	T	F	T
F	T	T	T	F	F	F	F
F	F	T	T	F	F	F	F
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

OBSERVATIONS

- To a certain extent, the **precedence** of the operator dictates the **order** in which the columns are created. (The last row, which is included only for explanation and is not part of the truth table, gives the order in which the columns were created.)

- In practice, it is **not necessary** to include **all** the columns **all** the time. Indeed, as one becomes increasing familiar with the behavior of operators, certain columns could be omitted. For example, (3) is not necessary if one can create (4) directly.

- It could be noted that the columns for $[p \wedge (\neg(\neg p \vee q))] \vee (p \wedge q)$ and p are **identical**. This means the two expressions are **logically equivalent**. It also means that, in certain cases, **structurally complex logical expressions** can be **simplified**.

  This **simplification has consequences** for **hardware** (simplified expressions decrease the number of logic gates on an electronic circuit, an electronic circuit with fewer gates is smaller, consumes less power, has a lower defect rate, and is cheaper to manufacture) and **software** (simplified expressions are easier to read and understand, and, in some cases, faster to process) [Lehman, Leighton, Meyer, 2012, Section 3.2].

REMARKS

Let p and q be two propositions.

$p \leftrightarrow q$ says: if p, then q, and if q, then p. This means p and q are true or false simultaneously. It, however, does **not** mean that p and q have the **same meaning**.

For example, p could be "The triangle ABC has two equal sides", and q could be "The triangle ABC has two equal angles". For another example, p could be "A triangle has six sides", and q could be "A square has two unequal sides".

NECESSITY VERSUS SUFFICIENCY

Let p and q be two propositions.

It is possible to have $p \longrightarrow q$ but not $q \longrightarrow p$, and it is possible to have $q \longrightarrow p$ but not $p \longrightarrow q$.

For example, **being a mammal** is necessary, but not sufficient for **being human** (as there are mammals that are not human), and that **a number *x* is rational** is sufficient, but not necessary for ***x* being a real number** (as there are real numbers that are not rational).

For another example, consider the Venn Diagram in Figure 12.

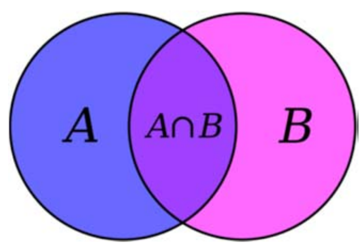

Figure 12. An example to illustrate the difference between necessity and sufficiency. (Source: Wikipedia.)

It follow that **being in the purple region** is sufficient for **being in *A***, but not necessary; **being in *A*** is necessary for **being in the purple region**, but not sufficient; and **being in *A*** and **being in *B*** is necessary and sufficient for **being in the purple region**.

17. MINIMAL LIST OF OPERATORS

There are some operators that can be defined in terms of **other operators**. There are **similarities** between the truth tables of biconditional and exclusive-or, and between the truth tables of implication and contrapositive of implication.

This may seem to imply that new symbols are **redundant**. Indeed, extra operators are for **convenience and succinctness**. (In fact, **negation** and **disjunction** are the only logical operators really needed to define all the others.)

EXAMPLE

Let p and q be the propositions:

p : I bought a lottery ticket this week.
q : I won the million dollar jackpot on Friday.

Express $\neg p \vee (p \wedge \neg q)$ as an English sentence.

Solution.

$\neg p$: I did not buy a lottery ticket this week.
$p \wedge \neg q$: I bought a lottery ticket this week and I did not win the million dollar jackpot on Friday.

Therefore,

$\neg p \vee (p \wedge \neg q)$: Either I did not buy a lottery ticket this week, or I bought a lottery ticket this week but I did not win the million dollar jackpot on Friday.

EXAMPLE

Let p, q, and r be the propositions:

p : You have the flu.
q : You miss the final examination.
r : You pass the course.

Express $(p \wedge q) \vee (\neg q \wedge r)$ as an English sentence.

Solution.

$p \wedge q$: You have the flu and miss the final examination.
$\neg q$: You do not miss the final examination.
$\neg q \wedge r$: You do not miss the final examination and you pass the course.

Therefore,

$(p \wedge q) \vee (\neg q \wedge r)$: Either you have the flu and miss the final examination, or you do not miss the final examination and do pass the course.

EXAMPLE

Explain, without the help of any construction (say, using a truth table or mathematical proof), why $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ is true when p, q, and r have the same truth value and is false otherwise.

Solution. Improvise. (This leads to two cases, and requires an understanding of the nature of $\vee$ and $\wedge$.)

18. REPRESENTING PROPOSITIONS AND LOGICAL OPERATORS AS FUNCTIONS

It is possible to represent **propositions** and **logical operators as functions**.

Let $\mathbf{B} = \{\mathrm{T}, \mathrm{F}\}$. Let $\mathrm{P}(p, q, r, \ldots)$ be a proposition. Then,

$$\mathrm{P} : \mathbf{B} \times \mathbf{B} \times \cdots \times \mathbf{B} \longrightarrow \mathbf{B}$$

is a function.

For example,

$$\mathrm{P}_1 := [(p \longrightarrow q) \wedge \neg q] \longrightarrow \neg p : \mathbf{B} \times \mathbf{B} \longrightarrow \mathbf{B}.$$

For another example,

$$\mathrm{P}_2 := (p \longrightarrow q) \longrightarrow r : \mathbf{B} \times \mathbf{B} \times \mathbf{B} \longrightarrow \mathbf{B}.$$

It is possible to represent **logical operators as functions**. Let o be a logical operator. Then,

$$\mathrm{o} : \mathbf{B} \times \mathbf{B} \times \cdots \times \mathbf{B} \longrightarrow \mathbf{B}$$

is a function.

For example, for unary and binary operators,

$$\neg : \mathbf{B} \longrightarrow \mathbf{B} \qquad \vee : \mathbf{B} \times \mathbf{B} \longrightarrow \mathbf{B} \qquad \wedge : \mathbf{B} \times \mathbf{B} \longrightarrow \mathbf{B}$$

19. BIT STRINGS

The electronic (digital) computers achieve their calculations inside **semi-conductors**. For reliability, only two stable voltage states are used. In these computers, most fundamental operations are carried out by **switching voltages** between these **two stable states**.

Logic is based upon only two truth values, namely true and false. Propositional logic can therefore be used for **modeling** computer activities.

High voltage values are modeled by **true** (or 1); **low** voltage values are modeled by **false** (or 0).

Voltage memory stored in a computer can be **represented** by a sequence of 0's and 1's. For example, 01 1011 0010 is such a representation.

Each number in the sequence is called a **bit**. The entire sequence of bits is called a **bit string**.

The **analogs** of the logical operations can be carried out inside the computer, one bit at a time. The results of the operations can then be transferred to entire bit strings.

EXAMPLE

The following is an example of exclusive-or of two bit strings:

$$\begin{array}{rl} & 01\ 1011\ 0010 \\ \oplus & \underline{10\ 0010\ 1111} \\ & 11\ 1001\ 1101 \end{array}$$

20. PROPOSITIONAL EQUIVALENCES

There is a need to **compare** propositions. The result of comparison can lead to further operations. For example, "if *p* is equivalent to *q*, then do something".

To do that, a **methodology** for **equivalences of propositions** is required.

There are **three types** of propositional equivalences:

(1) Tautology.
(2) Contradiction.
(3) Contingency.

(1) and (2) are **extremes**, (3) is somewhere in the **'middle'**. As an analogy, if (1) is white and (2) is black, then (3) is grey. In the real-world, (3) is most common.

If a proposition is a tautology or a contradiction, then it implies that the proposition under study is a **constant function**.

20.1. TAUTOLOGY

A tautology is a proposition that is **always true**.

The last column of the truth table of a tautology has **all** true values.

EXAMPLE

$p \vee \neg p$.

p	$\neg p$	$p \vee \neg p$
T	F	T
F	T	T

The proposition $p \vee \neg p$ is known as the **Complement Law** (or, equivalently, Negation Law).

EXAMPLE

Show, using a truth table, that $[(p \longrightarrow q) \wedge p] \longrightarrow q$ is a tautology.

Solution.

p	q	$p \longrightarrow q$	$[(p \longrightarrow q) \wedge p]$	$[(p \longrightarrow q) \wedge p] \longrightarrow q$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

The proposition $[(p \longrightarrow q) \wedge p] \longrightarrow q$ is known as **Modus Ponens**.

EXAMPLE

$(p \oplus q) \vee (p \oplus \neg q)$.

EXAMPLE

Show, using a truth table, that $[(p \longrightarrow q) \wedge \neg q] \longrightarrow \neg p$ is a tautology.

Solution. Improvise. The proposition $[(p \longrightarrow q) \wedge \neg q] \longrightarrow \neg p$ is known as **Modus Tollens**.

EXAMPLE

Show, using a truth table, that $[(p \vee q) \wedge (p \longrightarrow r) \wedge (q \longrightarrow r)] \longrightarrow r$ is a tautology.

Solution.

This solution is due to Xiabing Cui.

$[(p \vee q) \wedge (p \longrightarrow r) \wedge (q \longrightarrow r)] \longrightarrow r$
$\Leftrightarrow \neg[(p \vee q) \wedge (\neg p \vee r) \wedge (\neg q \vee r)] \vee r$
$\Leftrightarrow \neg(p \vee q) \vee \neg(\neg p \vee r) \vee \neg(\neg q \vee r) \vee r$
$\Leftrightarrow [\neg(p \vee q) \vee r] \vee [\neg(\neg p \vee r) \vee \neg(\neg q \vee r)]$
$\Leftrightarrow [\neg(p \vee q) \vee r] \vee \neg[(\neg p \vee r) \wedge (\neg q \vee r)]$
$\Leftrightarrow [\neg(p \vee q) \vee r] \vee \neg[(\neg p \wedge \neg q) \vee r)]$
$\Leftrightarrow [\neg(p \vee q) \vee r] \vee \neg[\neg(p \vee q) \vee r]$
$\Leftrightarrow$ T.

EXAMPLE

Determine whether $[\neg p \wedge (p \longrightarrow q)] \longrightarrow \neg q$ is a tautology.

Solution.

This is a conditional, which can only be false when $[\neg p \wedge (p \longrightarrow q)]$ is true and is $\neg q$ false. Therefore, $\neg q$ must remain true, that is, q must be false. For $[\neg p \wedge (p \longrightarrow q)]$ to be true $\neg p$ must be true, that is, p must be false.

In other words, for $[\neg p \wedge (p \longrightarrow q)] \longrightarrow \neg q$ to be true, both p and q must be false. However, a tautology is a proposition that is always true, regardless of the truth values of its constituent propositions.

Therefore, it is **not** a tautology.

EXAMPLE

Determine whether $[\neg p \wedge (p \vee q)] \longrightarrow q$ is a tautology.

Solution.

$[\neg p \wedge (p \vee q)] \longrightarrow q$
$\Leftrightarrow \neg\, [\neg p \wedge (p \vee q)] \vee q$
$\Leftrightarrow [p \vee \neg\, (p \vee q)] \vee q$
$\Leftrightarrow [(p \vee q) \vee \neg(p \vee q)]$
$\Leftrightarrow$ T.

Board Time!
Determine whether $[\neg p \wedge (p \vee q)] \longrightarrow p$ is a tautology.

20.2. CONTRADICTION

The only true wisdom is in knowing you know nothing.
— Socrates

I used to be indecisive, but now I'm not so sure.
— Boscoe Pertwee

A contradiction is a proposition that is **always false**.

The last column of the truth table of a contradiction has **all** false values.

EXAMPLE

$p \wedge \neg p$.

The truth table for this contradiction:

p	$\neg p$	$p \wedge \neg p$
T	F	F
F	T	F

EXAMPLE

$(p \oplus q) \wedge (p \oplus \neg q)$.

"IT'S ALL CATS AND DOGS"

Figure 13 shows a classical (quantum mechanical and comical) examples of contradiction.

(a) The Schrödinger's Cat, a thought experiment from quantum mechanics.

(b) An excerpt from 'The Adventures of Tintin' series of comic albums.

Figure 13. A collection of contradictions in the not necessarily mathematical world. (Source: Google Images.)

It is possible to come across contradictions in the real world, as shown in Figure 14.

Figure 14. A collection of contradictions in the real world. (Source: Google Images.)

20.3. CONTINGENCY

A contingency is a proposition that is **neither** a tautology, nor a contradiction. A contingency is **sometimes true and sometimes false**.

The **last column** of the truth table of a contingency has **some true and some false** values.

EXAMPLE

$p \longrightarrow \neg p$.

The truth table for this contingency:

p	$\neg p$	$p \longrightarrow \neg p$
T	F	F
F	T	T

EXAMPLE

$(p \oplus q) \longrightarrow (p \oplus \neg q)$.

The truth table for this contingency:

p	q	$\neg q$	$p \oplus q$	$p \oplus \neg q$	$(p \oplus q) \longrightarrow (p \oplus \neg q)$
T	T	F	F	T	T
T	F	T	T	F	F
F	T	F	T	F	F
F	F	T	F	T	T

EXAMPLE

$p \vee q \longrightarrow \neg r$.

If p, q, and r are all T, then $p \vee q \longrightarrow \neg r$ is F.
If p, q, and r are all F, then $p \vee q \longrightarrow \neg r$ is T.

EXAMPLE

I eat pasta on the weekend.

20.4. APPLICATIONS OF PROPOSITIONAL EQUIVALENCES

There are applications of propositional equivalences in **programming style** [Kernighan, Plauger, 1978; Kernighan, Pike, 1999], including the following:

- **Redundancy:** There should **never** be any **redundancies**[2] in source code, such as having part of the corresponding program that is never executed under any circumstances. (This type of redundancy is called **unreachable code**.)

- **Overflow:** There should **never** be any part of source code that leads to **buffer overflows**, such as, during adding an **infinite series of numbers**.

The presence of **tautologies and contradictions** in source code usually correspond to **poor** programming.

EXAMPLE

```
while(x <= 1 || x > 1) // Redundant
  x++; // Ad Infinitum
```

EXAMPLE

```
if(x > y)
  if(x == y)
    return; // Never Executed
```

[2] It should be noted that a repetition is **not** automatically a redundancy.

21. LOGICAL EQUIVALENCES

Two propositions p and q are logically equivalent if their **biconditional** $p \leftrightarrow q$ is a **tautology**.

The **last columns** of truth tables of logically equivalent propositions are **identical**. Conversely, if the last columns of truth tables of two propositions are identical, then they are **logically equivalent**.

If p and q are logically equivalent, then this fact is denoted by $p \Leftrightarrow q$ (or, equivalently, by $p \equiv q$). (If p and q are **not** logically equivalent, then this fact is denoted by $p \not\equiv q$.)

EXAMPLE

Show, using truth tables, that $p \longrightarrow q$ and $\neg p \vee q$ are logically equivalent.

Solution.

p	q	$\neg p$	$p \longrightarrow q$	$\neg p \vee q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

EXAMPLE

Show, using truth tables, that $p \longrightarrow q$ and $q \longrightarrow p$ are **not** logically equivalent.

Solution.

p	q	$p \longrightarrow q$	$q \longrightarrow p$	$(p \longrightarrow q) \leftrightarrow (q \longrightarrow p)$
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

EXAMPLE

(a) Show, using truth tables, that $p \oplus q$ and $[(p \vee q) \wedge \neg(p \wedge q)]$ are logically equivalent.

(b) Show, using truth tables, that $p \leftrightarrow q$ and $[(p \longrightarrow q) \wedge (q \longrightarrow p)]$ are logically equivalent.

(c) Show, using truth tables, that $\neg[p \vee [q \longrightarrow (p \wedge q)]] \wedge [q \vee (p \wedge q)]$ and $\neg p \wedge q$ are logically equivalent.

Solution. Improvise.

EXAMPLE

Show, without using truth tables, that $(p \longrightarrow q) \longrightarrow r$ and $p \longrightarrow (q \longrightarrow r)$ are **not** logically equivalent.

Solution.

If p, q, and r are all F, then $(p \longrightarrow q) \longrightarrow r$ is F, but $p \longrightarrow (q \longrightarrow r)$ is T. Therefore, they cannot be logically equivalent. This shows that the $\longrightarrow$ operator is **not associative**.

21.1. "LAWS" OF LOGIC: EXAMPLES OF IMPORTANT LOGICAL EQUIVALENCES

There are certain important logical equivalences that are often **(re)used**. There are also certain logical equivalences that generalize to **arbitrary number** of propositions.

These logical equivalences, due to their significance, are called **"laws"**. These "laws" can be helpful in **devising new logical statements from old logical statements**, and **deriving logical inferences** [Hammack, 2013, Section 2.12].

Table 2 includes the specifics of some basic "laws" of logic. The name of a law serves as a **mnemonic** as the mathematical expression it represents are related.

Identity Laws	$p \wedge \mathrm{T} \Leftrightarrow p$ $p \vee \mathrm{F} \Leftrightarrow p$
Domination Laws	$p \wedge \mathrm{F} \Leftrightarrow \mathrm{F}$ $p \vee \mathrm{T} \Leftrightarrow \mathrm{T}$
Negation Laws	$p \vee \neg p \Leftrightarrow \mathrm{T}$ $p \wedge \neg p \Leftrightarrow \mathrm{F}$
Idempotent Laws	$p \wedge p \Leftrightarrow p$ $p \vee p \Leftrightarrow p$
Double Negation	$\neg(\neg p) \Leftrightarrow p$
Commutative Laws	$p \wedge q \Leftrightarrow q \wedge p$ $p \vee q \Leftrightarrow q \vee p$
Absorption Laws	$p \vee (p \wedge q) \Leftrightarrow p$ $p \wedge (p \vee q) \Leftrightarrow p$
Associative Laws	$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$ $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$
Distributive Laws	$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$ $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$
De Morgan's Laws	[Conjunctional Negation] $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$ [Conjunctional Negation: Generalized Version] $\neg(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \Leftrightarrow (\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n)$ [Disjunctional Negation] $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$ [Disjunctional Negation: Generalized Version] $\neg(p_1 \vee p_2 \vee \ldots \vee p_n) \Leftrightarrow (\neg p_1 \wedge \neg p_2 \wedge \ldots \wedge \neg p_n)$

Table 2. A collection of some basic logical equivalences.

EXAMPLE

(This example is attributed to [Sundstrom, 2018].)

Let a be a real number and let f be a real-valued function defined on an interval containing $x = a$. The following conditional statement is given:

If f is differentiable at $x = a$, then f is continuous at $x = a$.

From the following statements point out those that (1) have the same meaning as the given conditional statement and (2) the ones that are negations of the given conditional statement:

(a) If f is continuous at $x = a$, then f is differentiable at $x = a$.
(b) If f is not differentiable at $x = a$, then f is not continuous at $x = a$.
(c) If f is not continuous at $x = a$, then f is not differentiable at $x = a$.
(d) f is not differentiable at $x = a$ or f is continuous at $x = a$.
(e) f is not continuous at $x = a$ or f is differentiable at $x = a$.
(f) f is differentiable at $x = a$ and f is not continuous at $x = a$.

Solution.

(The problem is **not** asking which statements are true and which are false. It is asking which statements are logically equivalent to the given statement. It might be helpful to let p represent the antecedent of the given statement, q represent the consequent, and then determine a symbolic representation for each statement.)

(1) (c) and (d).
(2) (f).

22. LIMITATIONS OF TRUTH TABLES

The ability to perform a **context-independent calculation** is an advantage as well as a limitation of truth tables. There is no need to understand the reason why a given proposition is true (or false, or sometimes true) [O'Donnell, Hall, Page, 2006, Section 6.4.2].

The use of **two-dimensional space** is yet another advantage as well as a limitation of truth tables.

EXAMPLE

Give the number of rows required to construct the truth table of the following expression:

$$[(q \leftrightarrow (p \longrightarrow r)] \wedge [\neg(s \wedge r) \vee \neg t)] \longrightarrow (\neg q \longrightarrow r).$$

Solution. 32 (= 2^5) rows. (Indeed, each additional variable **doubles** the number of rows.)

As the number of atomic components in a compound proposition increases, the size of the truth table for that compound proposition increases **exponentially** [Reba, Shier, 2015, Chapter 8].

For n variables, there will be 2^n rows.

It can be **tedious and error-prone** to construct truth tables for large, compound, propositions. Therefore, an alternative approach to finding the truth values of propositions and checking for propositional equivalences is needed. This motivates **derivational (descriptive) proof techniques**.

23. DERIVATIONAL PROOFS

A **proof** is a **logically structured argument**, usually consisting of one or more statements, which **demonstrates** that a certain proposition is true.

The derivational (descriptive) proof techniques for propositional equivalences can be of two types:

1. **Natural Language-Based.**
   (a) Use of the definitions and truth values of the propositions is involved.
   (b) Usually useful for **simple** cases.

2. **Mathematics-Based.**
   (a) Use of logical equivalence and laws of logic is made.
   (b) Useful for any type of (**simple or complex**) cases.

EXAMPLE

Show, using natural language proof, that $p \longrightarrow q$ and $\neg q \longrightarrow \neg p$ are logically equivalent.

Solution.

By definition of logically equivalence, it needs to be shown that the truth values of $p \longrightarrow q$ and $\neg q \longrightarrow \neg p$ are identical. Now, $\neg q \longrightarrow \neg p$ is false only when $\neg p$ is false and $\neg q$ is true, that is, when p is true and q is false. This is the same for $p \longrightarrow q$. Therefore, the two are logically equivalent.

EXAMPLE

Show, using natural language proof, that $(p \longrightarrow q) \wedge (p \longrightarrow r)$ and $p \longrightarrow (q \wedge r)$ are logically equivalent.

Solution. There are two cases.

Case 1: Let $(p \longrightarrow q) \wedge (p \longrightarrow r)$ be true. To show that $p \longrightarrow (q \wedge r)$ is also true means showing that $(q \wedge r)$ is true whenever p is true. So, let p be true. Now, $(p \longrightarrow q) \wedge (p \longrightarrow r)$ is true means that both q and r must be true. Therefore, $(q \wedge r)$ is true.

Case 2: Let $p \longrightarrow (q \wedge r)$ be true. To show that $(p \longrightarrow q)$ is true and $(p \longrightarrow r)$ is true means showing that if p is true, then both q and r must also be true. However, this follows from $p \longrightarrow (q \wedge r)$ being true.

EXAMPLE

Show, (1) using a truth table, and (2) using a mathematical proof, that $[\neg p \wedge (p \vee q)] \longrightarrow q$ is a tautology.

Solution (1). In the truth table, all rows in the last column must contain a T.

p	q	$\neg p$	$p \vee q$	$\neg p \wedge (p \vee q)$	$[\neg p \wedge (p \vee q)] \longrightarrow q$
T	T	F	T	F	T
T	F	F	T	F	T
F	T	T	T	T	T
F	F	T	F	F	T

Solution (2). In a mathematical proof, the goal is to obtain a T at the end of the steps of the proof.

$[\neg p \wedge (p \vee q)] \longrightarrow q$	
$\Leftrightarrow [(\neg p \wedge p) \vee (\neg p \wedge q)] \longrightarrow q$	(Distributive Law)
$\Leftrightarrow [\mathrm{F} \vee (\neg p \wedge q)] \longrightarrow q$	(Negation Law)
$\Leftrightarrow [\neg p \wedge q] \longrightarrow q$	(Identity Law)
$\Leftrightarrow \neg\, [\neg p \wedge q] \vee q$	
$\Leftrightarrow [\neg\, (\neg p) \vee \neg q] \vee \mathrm{q}$	(De Morgan's Law)
$\Leftrightarrow [p \vee \neg q] \vee q$	(Double Negation Law)
$\Leftrightarrow p \vee [\neg q \vee q]$	(Associative Law)
$\Leftrightarrow p \vee [q \vee \neg q]$	(Commutative Law)
$\Leftrightarrow p \vee \mathrm{T}$	
$\Leftrightarrow \mathrm{T}.$	(Domination Law)

EXAMPLE

Show that $(p \wedge q) \longrightarrow r$ and $(p \longrightarrow r) \wedge (q \longrightarrow r)$ are **not** logically equivalent.

Solution.

The idea is to find an assignment of truth values that makes one of the given propositions true and the other false. Let p be true, and let q and r be false. Then, $(p \wedge q) \longrightarrow r$ is $\mathrm{F} \longrightarrow \mathrm{F}$, which is true, but $(p \longrightarrow r) \wedge (q \longrightarrow r)$ is $\mathrm{F} \wedge \mathrm{T}$, which is false.

EXAMPLE

Show, using a mathematical proof, that $\neg[p \vee (\neg p \wedge q)]$ and $\neg p \wedge \neg q$ are logically equivalent.

Solution.

$\neg[p \vee (\neg p \wedge q)]$	
$\Leftrightarrow \neg p \wedge \neg(\neg p \wedge q)$	(De Morgan's Law)
$\Leftrightarrow \neg p \wedge [\neg(\neg p) \vee \neg q]$	(De Morgan's Law)
$\Leftrightarrow \neg p \wedge (p \vee \neg q)$	(Double Negation Law)
$\Leftrightarrow (\neg p \wedge p) \vee (\neg p \wedge \neg q)$	(Distributive Law)
$\Leftrightarrow \mathrm{F} \vee (\neg p \wedge \neg q)$	
$\Leftrightarrow \neg p \wedge \neg q.$	(Identity Law)

REMARKS

- This example shows that certain expressions can be **simplified**, that is, a **longer expression** can be reduced to a **shorter expression**.

- The idea of **simplifying logical expressions** is similar to **simplifying algebraic expressions**. For example, $x^3 + 3x^2y + 3xy^2 + y^3 = (x + y)^3$.

EXAMPLE

Simplify

$$[p \wedge (\neg(\neg p \vee q))] \vee (p \wedge q).$$

Solution. There are multiple correct proofs.

Longer:

$[p \wedge (\neg(\neg p \vee q))] \vee (p \wedge q)$	
$\Leftrightarrow [p \wedge (p \wedge \neg q)] \vee (p \wedge q)$	(Double Negation Law, De Morgan's Law)
$\Leftrightarrow (p \wedge \neg q) \vee (p \wedge q)$	(Associative Law, Idempotent Law)
$\Leftrightarrow [(p \wedge \neg q) \vee p] \wedge [(p \wedge \neg q) \vee q]$	(Distributive Law)
$\Leftrightarrow [(p \vee p) \wedge (\neg q \vee p)] \wedge [(p \vee q) \wedge (\neg q \vee q)]$	(Distributive Law)
$\Leftrightarrow [(p) \wedge (\neg q \vee p)] \wedge [(p \vee q) \wedge (\mathrm{T})]$	(Negation Law)
$\Leftrightarrow p \wedge (p \vee q)$	(Absorption Law, Identity Law)
$\Leftrightarrow p.$	(Absorption Law)

Shorter:

$[p \wedge (\neg(\neg p \vee q))] \vee (p \wedge q)$	
$\Leftrightarrow [p \wedge (p \wedge \neg q)] \vee (p \wedge q)$	(Double Negation Law, De Morgan's Law)
$\Leftrightarrow (p \wedge \neg q) \vee (p \wedge q)$	(Associative Law, Idempotent Law)
$\Leftrightarrow p \wedge (\neg q \vee q)$	(Distributive Law)
$\Leftrightarrow p \wedge (\mathrm{T})$	(Negation Law)
$\Leftrightarrow p.$	(Identity Law)

REMARKS

The previous example shows that **multiple proofs** are possible, some shorter than others. Usually, a shorter proof is preferable to a longer proof, if only it occupies less space. However, as in other cases, including **source code**, there is **no direct correlation between length and readability, and between length and understandability**. A shorter proof may or may not be easier to read or understand.

EXAMPLE

Show, using a mathematical proof, that $(p \wedge q) \longrightarrow (p \vee q)$ is a tautology.

Solution. Improvise.

24. CONSIDERATIONS IN SOLVING PROPOSITIONAL LOGIC PROBLEMS

In solving propositional logic problems, there are three important considerations:

1. Completeness
2. Consistency
3. Patterns of Reuse

24.1. COMPLETENESS

The following recommendations are meant for making a proof **readable and understandable** to others:

- **Using Truth Tables.** It is important to include **columns for key steps**. This is evident when asked to prove that a proposition is a tautology or a contradiction.

- **Using Proof Techniques.** It is important (1) to include **key steps**, and (2) to indicate the **laws** used in a transition between steps. The law itself need not be repeated; stating its **name** is sufficient.

24.2. CONSISTENCY

The following recommendations are meant for making a proof **readable and understandable** to others:

- **Using Truth Tables.** It is important to be consistent in the **order** of column allocation in truth tables. It makes assigning truth values (T or F) **easier** and the result more **readable**. This is particularly important for **compound propositions**.

  For example, a column order of p first, q next, and so on, is a good trait.

- **Using Proof Techniques.** It is better to derive the **left-hand side (LHS)** and the **right-hand side (RHS)** separately.

24.3. PATTERNS OF REUSE

The following recommendations are meant for making a proof **concise**:

- The laws of logic are **frequently occurring** basic results proven to be true. They are **reusable**.

- The list of **basic results** can be **extended** to other, often used, results where the equivalence is close to a **definition**. This is particularly useful in proofs involving complex propositions. However, the use of those should be justifiable.

  For example, the fact that $p \longrightarrow q$ and $\neg p \vee q$ and $\neg q \longrightarrow \neg p$ and $p \leftrightarrow (p \wedge q)$ are logically equivalent can be **reused** as a **standard result**.

25. PROPOSITIONAL FUNCTIONS

A proposition **does not** have any inherent **semantics**. This can lead to **unsatisfactory answers**.

A **propositional function** solves this problem. A propositional function **associates semantics** with a proposition via the use of **logical quantifiers**.

25.1. MOTIVATION FOR PROPOSITIONAL FUNCTIONS

NOTATION

The symbol := means “is, by definition, equal to”.

EXAMPLE

Give the truth value of the proposition the compound proposition C:

If Nazlie is a spider, then Nazlie has 8 legs.

Solution 1. C is composed of two atomic propositions:

p := "Nazlie is a spider"

and

q := "Nazlie has 8 legs".

C is represented by $p \longrightarrow q$.

C is a conditional that is **always true** when p is false.

Solution 1 is not satisfactory.

C should be true because of the fact that spiders have 8 legs, **not** because of some **non-semantic technicality** in the truth table of the implication.

However, propositional logic does **not** take **semantics** into account. Thus, there is **no way** that p could impact q, or affect the truth value of $p \longrightarrow q$.

Solution 2. The use of **logical quantifiers** can help resolve this issue.

Therefore, using logical quantifiers:

For all x, if x is a spider, then x has 8 legs.

Then, as a **special case** of x := Nazlie, it can be concluded that $p \longrightarrow q$ is indeed true as expected.

25.2. DEFINITION OF PROPOSITIONAL FUNCTION

The previous example motivates the following definition:

Definition [Propositional Function]. A statement (expression) that has a variable (or variables) that represent various possible subjects.

Let S be a set. Then, a propositional function P is defined by any one of the following ways:

$$P: S \longrightarrow \{\mathrm{T}, \mathrm{F}\}$$

or

$$P: S \times S \times \cdots \times S \longrightarrow \{\mathrm{T}, \mathrm{F}\}.$$

Let S_1 , S_2, …, S_n be sets. Then, a propositional function P is defined by

$$P: S_1 \times S_2 \times \cdots \times S_n \longrightarrow \{\mathrm{T}, \mathrm{F}\}.$$

OBSERVATIONS

- A propositional function is a **generalization** of a proposition.
- A propositional function has a **subject** and a **predicate**. By taking a subject variable, and applying a predicate, one obtains a propositional function.
- A **quantifier** is used to **bind** the variables and **create a proposition** with **embedded semantics**. When an object from the **universe of discourse** is placed into the variable(s), then the result is a truth value.

25.3. PROPOSITIONAL FUNCTION VERSUS PROPOSITION

It can be noted that a **propositional function**, by itself, is **not a proposition**. Indeed, a propositional function, when **assigned certain values**, becomes a proposition.

To be specific, a propositional function **becomes** a proposition **if** the following occurs:

- **Method 1:** The variable(s) contained is(are) assigned specific value(s).
- **Method 2:** The variable(s) contained is(are) quantified.

(Method 1 and Method 2 are indeed **different**.)

EXAMPLE

For all x, if x is a spider, then x has 8 legs.

This is composed of two **atomic** propositional functions:

$$P(x) = \text{“}x \text{ is a spider”}$$

and

$$Q(x) = \text{“}x \text{ has 8 legs”}$$

The conditional $P(x) \longrightarrow Q(x)$ is formed, and is bound by “For all x”.

A **universe of discourse** for x is set.

Finally, for a **special case** of x := Nazlie, the propositional function becomes a proposition, and its truth value can be concluded.

26. UNIVERSE OF DISCOURSE

The propositions such as “John is an actor”, “Kathy is 5 years old”, and “Andrew is in the house” have **no intrinsic meaning**. They are either true or false, but **no further inference** can be derived.

If logical propositions are to have **meaning**, they need to **describe something**.

(In general, in logic there is no **context**.) In order to equip propositions with **meaning**, there is need for **context**. This context is provided by the **universe of discourse**.

The universe of discourse is the specific **domain of the variable** in a proposition.

Definition [Universe of Discourse]. The collection of **subjects (or nouns)** about which the propositions relate.

In the domain of a universe of discourse, a property is true for **all values** of a variable.

The universe of discourse is often simply called **“universe”** and denoted by **“*U*”**.

In this document, the universe of discourse has a **finite number of elements**, unless otherwise stated.

EXAMPLE

There can be domains of natural language **characters**, such as the letters of English alphabet. (In this case, the universe of discourse has a finite number of elements.)

EXAMPLE

There can be domains of **numbers**, such as natural numbers, integers, and real numbers. (In this case, the universe of discourse has an infinite number of elements.)

EXAMPLE

Give a universe of discourse for the following propositions: “John is an actor”, “Kathy is 5 years old”, and “Andrew is in the house”.

Solution. There are **many** possibilities:

- John, Kathy, and Andrew. (This is the **smallest** correct answer.)
- Humans.
- Vertebrates.
- Animals.

27. PREDICATES

Definition [Predicate]. A **property** or description of **subjects** in a **universe of discourse**.

A **subject** is usually related to a **noun**; a **predicate** is usually related to an **adjective**.

EXAMPLE

Robert <u>**is tall**</u>.
The lake <u>**is deep**</u>.
11 <u>**is a prime number**</u>.
The Web Application <u>**is usable**</u>.

28. QUANTIFIERS

There are two important types of quantifiers:

1. **Universal Quantifier:** "$\forall$" reads "for all".

2. **Existential Quantifier:** "$\exists$" reads "there exists". ("$\exists!$" reads "there exists a unique". $\exists!$ is the **uniqueness quantifier**.)

A quantifier is **placed in front** of a propositional function, and **binds** it to obtain a proposition with **semantic value**.

MNEMONICS

I think part of the appeal of mathematical logic is that the formulas look mysterious – You write backward Es!
— Hilary Putnam

$\forall$: Upside-Down A signifies "For **A**ll".

$\exists$: Reverse E signifies "There **E**xists".

29. UNIVERSAL QUANTIFICATION

$P(x)$ is true **for every** x in the universe of discourse.

The common **synonyms** are "for every", "for all", and "for each".

Notation: $\forall x\, P(x)$.

Read as: "For every x, $P(x)$ is true", or "For all x, $P(x)$ is true", and so on.

Let the universe of discourse consist of $x_1, x_2, \ldots, x_n$. The **conjunction** is over **entire universe of discourse**:

$$\forall x\, P(x) \Leftrightarrow P(x_1) \wedge P(x_2) \wedge P(x_3) \wedge \cdots \wedge P(x_n).$$

The variable x is bound by the universal quantifier producing a proposition. "$\forall x\, P(x)$" is true when **every instance** of x makes $P(x)$ true when plugged in.

EXAMPLE

Universe: All students in computer science.
$P(x)$: x has an account on Alpha.

$\forall x\, P(x)$ means that "every student in computer science has an account on Alpha".

EXAMPLE

Universe = {L: A lion, M: A spider with 8 legs, N: A spider with 7 legs}.
Propositional Functions: $P(x)$ = "x is a spider" and $Q(x)$ = "x has 8 legs".

Find the truth value of $\forall x\, [\, P(x) \longrightarrow Q(x)\,]$.

Solution.

False. The proposition is equivalent to:

$$[\, P(\mathrm{L}) \longrightarrow Q(\mathrm{L})\,] \wedge [\, P(\mathrm{M}) \longrightarrow Q(\mathrm{M})\,] \wedge [\, P(\mathrm{N}) \longrightarrow Q(\mathrm{N})\,].$$

N is a **counterexample** that makes it false.

EXAMPLE

Universe: **R**.
$P(x)$: $|\, x + 1\, | > 0$.

Find the truth value of $\forall x\, P(x)$.

Solution.

False. $x = -1$ is the (only) **counterexample** that makes it false.

EXAMPLE

Universe: $\varnothing$.
$P(x)$: Any.

Find the truth value of $\forall x\, P(x)$.

Solution.

True. This makes sense because there is **nothing that could refute the proposition**.

In the real world, the above does not make sense [Grieser, 2018, Chapter 7]. If a person says "all my cars are environment-friendly", then people will assume that the person owns **at least one** car. However, mathematics sometimes does not mimic the real world.

30. EXISTENTIAL QUANTIFICATION

$P(x)$ is true **for some** x in the universe of discourse.

The common **synonyms** are "for some", "there exists", and "for at least one".

Notation: $\exists x\, P(x)$.

Read as: "For some x, $P(x)$ is true", "There exists an x such that $P(x)$ is true", and so on.

Let the universe of discourse consist of x_1, x_2, …, x_n. The **disjunction** is over **entire universe of discourse**:

$$\exists x\, P(x) \Leftrightarrow P(x_1) \vee P(x_2) \vee P(x_3) \vee \cdots \vee P(x_n).$$

The variable x is bound by the existential quantifier producing a proposition. $\exists x\, P(x)$ is true when **an instance can be found** which, when substituted for x, makes $P(x)$ true.

In case of the **uniqueness quantifier**, the disjunction is over entire universe of discourse, **one instance at a time**:

$$\begin{aligned}\exists! x\, P(x) \Leftrightarrow\ & [\ P(x_1) \vee \neg P(x_2) \vee \neg P(x_3) \vee \cdots \vee \neg P(x_n)\ ] \\ & \vee [\ \neg P(x_1) \vee P(x_2) \vee \neg P(x_3) \vee \cdots \vee \neg P(x_n)\ ] \\ & \vee \ldots \\ & \vee [\ \neg P(x_1) \vee \neg P(x_2) \vee \neg P(x_3) \vee \cdots \vee P(x_n)\ ]\end{aligned}$$

EXAMPLE

Universe: All students in computer science.
$P(x)$: x has an account on Alpha.

$\exists x\ P(x)$ means that "there is **a** student in computer science who has an account on Alpha".

EXAMPLE

Universe: {L: A lion, M: A spider with 8 legs, N: A spider with 7 legs}.
Propositional Functions: $P(x)$ = "x is a spider" and $Q(x)$ = "x has 8 legs".

Find the truth value of $\exists x\ [\ P(x) \longrightarrow Q(x)\ ]$.

Solution.

True. The proposition is equivalent to:

$$[\ P(\mathrm{L}) \longrightarrow Q(\mathrm{L})\ ] \vee [\ P(\mathrm{M}) \longrightarrow Q(\mathrm{M})\ ] \vee [\ P(\mathrm{N}) \longrightarrow Q(\mathrm{N})\ ].$$

L is an example that makes it true.

EXAMPLE

Universe: $\mathbf{Z}$.
$P(x)$: $|\ 2x - 1\ | > 1$.

Find the truth value of $\exists x\ P(x)$.

Solution.

True. For example, let $x = 2$.

EXAMPLE

Universe: {CSE students, ECE students, Special Needs students}.

$C(x)$: x is a CSE student.
$E(x)$: x is an ECE student.
$S(x)$: x is a special needs student.

In the following, certain predicates are given in English. Use mathematical notation to express each predicate.

(a) Everybody is a CSE student.
(b) Nobody is an ECE student.
(c) All CSE students are special needs students.
(d) Some CSE students are special needs students.
(e) No CSE student is an ECE student.
(f) There does not exist a CSE student who is also an ECE student.
(g) If an ECE student is a special needs student, then that student is also a CSE student.
(h) There are three CSE students who are also special needs students.

Solution.

(a) **Everybody is a CSE student.**

$\forall x\ C(x)$.

(b) **Nobody is an ECE student.**

(This could be restated directly as (1) "Everybody is not an ECE student." It could also be **restated indirectly** as (2) "(The negation of) There is a student who is an ECE student." or "(The negation of) Some student is an ECE student.".) It is often easier to work with (2) than with (1).

$\forall x\ \neg E(x)$

or, equivalently

$\neg \exists x\ E(x)$.

(c) **All CSE students are special needs students.**

(A universal quantifier ($\forall$) usually takes **implications**. In other words, if x is a CSE student, then x is a special needs student. The previous assertion applies to every x.)

$\forall x\ [\ C(x) \longrightarrow S(x)\ ].$

(d) **Some CSE students are special needs students.**

(An existential quantifier ($\exists$) usually takes **conjunctions**.)

$\exists x\ [\ C(x) \wedge S(x)\ ].$

(e) **No CSE student is an ECE student.**

(In other words, if x is a CSE student, then x is not an ECE student. The previous assertion applies to every x.)

$\forall x\ [\ C(x) \longrightarrow \neg E(x)\ ].$

(f) **There does not exist a CSE student who is also an ECE student.**

$\neg\exists x\ [\ C(x) \wedge E(x)\ ].$

(g) **If an ECE student is a special needs student, then that student is also a CSE student.**

(In other words, if x is an ECE student and x is a special needs student, then x is a CSE student. The previous assertion applies to every x.)

$\forall x\ [\ (E(x) \wedge S(x)) \longrightarrow C(x)\ ].$

(h) **There are three CSE students who are also special needs students.**

(This **assumes** that there are at least three different students in the universe.)

$\exists x_1\ \exists x_2\ \exists x_3$
$[((x_1 \neq x_2) \wedge (x_1 \neq x_3) \wedge (x_2 \neq x_3)) \wedge ((C(x_1) \wedge S(x_1)) \wedge (C(x_2) \wedge S(x_2)) \wedge (C(x_3) \wedge S(x_3)))].$

EXAMPLE

Determine the truth value of each of the following statements if the domain consists of real numbers:

(a) $\exists x\ (x^4 < x^2)$.
(b) $\forall x\ (2x > x)$.

Solution.

(a) True. For example, let $x = 1/2$. (In general, any x in the open interval (0, 1) or (–1, 0) would work.)

(b) False. For example, let $x = 0$. (In general, any non-positive x would work.)

EXAMPLE

Let $N(x)$ be the statement "x has visited New York City," where the domain consists of the students at a particular University. Express each of these quantifications in English.

(a) $\neg\forall x\ N(x)$.
(b) $\forall x\ \neg N(x)$.

Solution.

(a) It is not the case that every student in the school has visited New York City. (Or, alternatively, not all students in the University have visited New York City.)

(b) All students in the University have not visited New York City. (Or, alternatively, no student in the University has visited New York City.)

REMARKS

It is important to note that (a) and (b) are **not equivalent**.

EXAMPLE

Translate each of the following statements into logical expressions using predicates, quantifiers, and logical connectives. Let the domain consist of all people in the world.

(a) Everyone in your class has a mobile phone.
(b) Someone in your class has seen a foreign movie.
(c) Someone in your class cannot swim.

Solution.

Let $C(x)$ be the propositional function "x is in your class."
Let $P(x)$ be the propositional function "x has a mobile phone."
Let $M(x)$ be the propositional function "x has seen a foreign movie."
Let $S(x)$ be the propositional function "x can swim."

(a) $\forall x\ [C(x) \longrightarrow P(x)]$.
(b) $\exists x\ [C(x) \wedge M(x)]$.
(c) $\exists x\ [C(x) \wedge \neg S(x)]$.

REMARKS

It is important to note that the domain consists of **all** people in the world. Therefore, one needs to **make sure that a person is in the class** before making any other statements about that person.

31. MULTIVARIATE PROPOSITIONAL FUNCTIONS

The **multivariable (or multivariate)** predicates, together with their variables create multivariable propositional functions.

EXAMPLE

x is faster than y.
a is smaller than one of b, c.
x is at least n inches taller than y.

32. MULTIVARIATE PREDICATES

The multivariable predicates generalize predicates to allow descriptions of **relationships** between subjects.

The subjects of multivariable predicates **may or may not be** in the same universe of discourse.

EXAMPLE

Give the universe of discourse in each of the following cases:

(a) John is taller than Derek.
(b) 13 is greater than one of 12, 15.
(c) John is at least 5 inches taller than Derek.

Solution. There are many correct answers, including the following:

(a) The universe of discourse of **both** variables is all **people** in the world.
(b) The universe of discourse of **all three** variables is **Z**, the set of integers.
(c) The universe of discourse of the **first and last** variables is **all people** in the world. The universe of discourse of the **second** variable is **R**, the set of real numbers.

33. MULTIVARIATE QUANTIFICATION

Quantification involving **single variable** is a collection of $\vee$'s, or a collection of $\wedge$'s.

However, quantification involving **multiple variables** is **different**. If multiple variables are involved, such that each variable is bound by a quantifier, two things are important:

1. The **order** of the binding.
2. The resultant **meaning**.

Quantification involving multiple variables leads to **nested quantifiers**.

33.1. QUANTIFICATION IN TWO VARIABLES

There are **four** notable possibilities that are considered in detail.

1. For all pair x, y, $P(x, y)$.

Notation:

$\forall x\ \forall y\ P(x, y)$
or
$\forall x\ (\forall y\ P(x, y))$
or
$\forall x, y\ P(x, y)$.

True: $P(x, y)$ is true for every pair x, y.
False: There is a pair x, y for which $P(x, y)$ is false.

2. For every x, there is a y, such that $P(x, y)$.

Notation:

$\forall x \; \exists y \; P(x, y)$

True: For every x there is a y for which $P(x, y)$ is true.
False: There is an x such that $P(x, y)$ is false for every y.

3. There is an x, such that $P(x, y)$ for all y.

Notation:

$\exists x \; \forall y \; P(x, y)$

True: There is an x for which $P(x, y)$ is true for every y.
False: For every x there is a y for which $P(x, y)$ is false.

4. There is a pair x, y, such that $P(x, y)$.

Notation:

$\exists x \; \exists y \; P(x, y)$
or
$\exists x \; (\exists y \; P(x, y))$
or
$\exists x, y \; P(x, y)$.

True: There is a pair x, y for which $P(x, y)$ is true.
False: $P(x, y)$ is false for every pair x, y.

EXAMPLE

Let P: $\mathbf{Z} \times \mathbf{Z} \longrightarrow \{T, F\}$, where $P(x, y)$ denotes "$x + y = 5$". Give the truth value of the following propositions:

(a) $\forall x\ \forall y\ P(x, y)$.
(b) $\forall x\ \exists y\ P(x, y)$.
(c) $\exists x\ \forall y\ P(x, y)$.
(d) $\exists x\ \exists y\ P(x, y)$.

Solution.

(a) $\forall x\ \forall y\ P(x, y)$.

False. For example, $x = 0$ and $y = 0$.

(b) $\forall x\ \exists y\ P(x, y)$.

True. For any x, it is always possible to find a y by setting $y = 5 - x$. Figure 15 illustrates the graph of the **"continuous version"** of $y = 5 - x$.

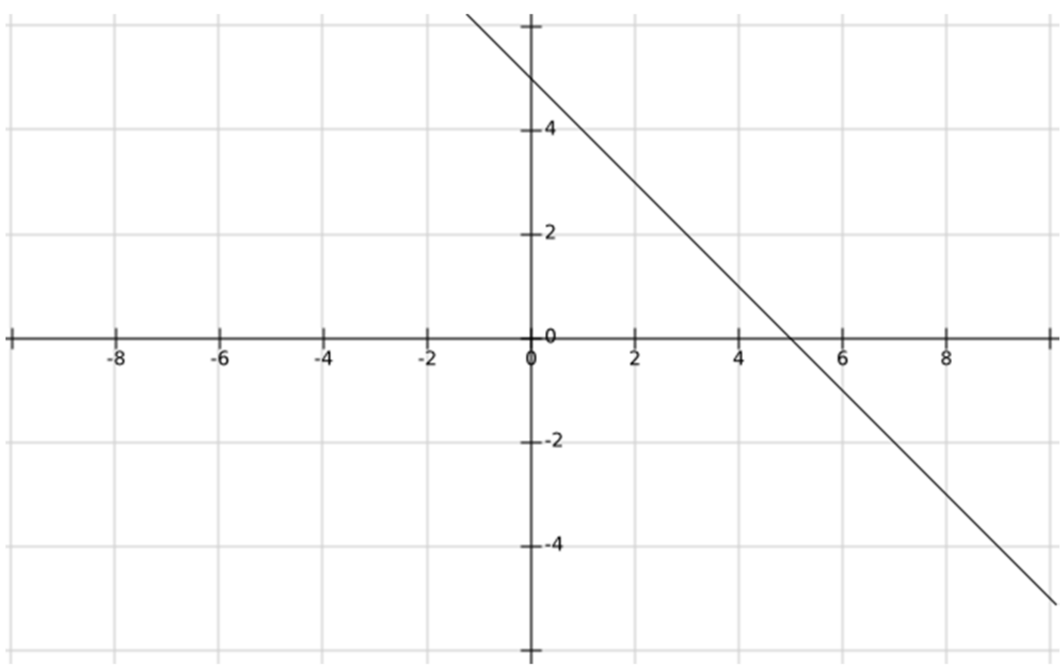

Figure 15. The graph of $y = 5 - x$.

(c) $\exists x\ \forall y\ P(x, y)$.

False. For example, $x = \text{n}$ and $y \neq 5 - \text{n}$.

(d) $\exists x\ \exists y\ P(x, y)$.

True. For example, $x = 2$ and $y = 3$.

EXAMPLE

Let $Q(x, y)$ be the statement "$x + y = x - y$." Let the domain for both x and y be the set of integers. Give the truth values of the following propositions:

(a) $\exists x\ Q(x, 2)$.
(b) $\exists x\ \exists y\ Q(x, y)$.
(c) $\forall x\ \exists y\ Q(x, y)$.
(d) $\exists y\ \forall x\ Q(x, y)$.

Solution.

(a) False. The equation $x + 2 = x - 2$ has no solution.

(b) True. For example, $x = 0$ and $y = 0$ satisfies the statement.

(c) True. Take $y = 0$ for each x.

(d) True. Take $y = 0$.

33.2. ORDER OF VARIABLES IN QUANTIFICATION

You can fool all the people some of the time, and some of the people all the time, but you cannot fool all the people all the time.
— Abraham Lincoln

A quantification in two (or more) variables leads to the possibilities of **different orders** of application of quantifiers.

In general, the order of the quantifiers matters, but **not** always:

- **Quantifiers of Same Type:** If the quantifiers are of the **same type**, then the order **does not** matter.

- **Quantifiers of Different Types:** If the quantifiers are of **different types**, then the **order is important**.

(In any case, a lack of understanding the **difference** is one of the major sources of errors.)

$\forall x\ \forall y$ is the **same** as $\forall y\ \forall x$

This is because these are both interpreted as "for every combination of x and y …".

$\exists x\ \exists y$ is the **same** as $\exists y\ \exists x$

This is because these are both interpreted as "there exists a pair x, y …".

The **order matters with quantifiers of different types** because the **meaning of the expression changes if the order is changed**. This fact is illustrated in a number of examples in this document.

EXAMPLE

Let $T(x, y)$: x trusts y; x, y are people.

Then,

$\forall x\ \forall y\ T(x, y)$	Everybody trusts everyone
$\forall y\ \forall x\ T(x, y)$	Everybody trusts everyone
$\forall x\ \exists y\ T(x, y)$	Everybody trusts someone
$\exists y\ \forall x\ T(x, y)$	Somebody is trusted by everyone
$\forall y\ \exists x\ T(x, y)$	Everybody is trusted by someone
$\exists x\ \forall y\ T(x, y)$	Somebody trusts everyone
$\exists x\ \exists y\ T(x, y)$	Somebody trusts someone
$\exists y\ \exists x\ T(x, y)$	Somebody trusts someone

(It can be noted that ***x* trusts *y*** and ***y* is trusted by *x*** are **inverse** of each other.)

It can be seen that the **order does not matter** when **quantifiers are of the same type**.

The following are the **same**:

$\forall x\ \forall y\ T(x, y)$	Everybody trusts everyone
$\forall y\ \forall x\ T(x, y)$	Everybody trusts everyone

It can be seen that **order does matter** when **quantifiers are not of the same type**.

The following are **not the same**:

$\forall x\ \exists y\ T(x, y)$	Everybody trusts someone
$\exists y\ \forall x\ T(x, y)$	Somebody is trusted by everyone

$\forall x\ \exists y\ T(x, y)$ means "For every x there is a y for which $T(x, y)$ is true". Therefore, selecting a person x_1 should give a person y_1 such that x_1 trusts y_1, selecting a person x_2 should give a person y_2 such that x_2 trusts y_2, and so on. It can be noted that y_1 and y_2 need **not** be the same.

$\exists y\ \forall x\ T(x, y)$ means "There is a y for which $T(x, y)$ is true for every x". Therefore, it should be possible to find a y, such that, for $x_1, x_2, x_3, \ldots$, x_1 trusts y, x_2 trusts y, x_3 trusts y, and so on.

(In other words, $\forall$ and $\exists$ are **not commutative** [Houston, 2009, Chapter 10]).

EXAMPLE

Universe: **N**.
$R(x, y) := x < y$.

State the meaning and truth values of (1) $\forall x\ \exists y\ R(x, y)$, and (2) $\exists y\ \forall x\ R(x, y)$. Explain.

Solution.

(1) $\forall x\ \exists y\ R(x, y)$:

Meaning: All numbers x admit a bigger number y.

Truth Value: True.

Rationale: Set $y = x + 1$.

(2) $\exists y \, \forall x \, R(x, y)$:

Meaning: Some number y is bigger than all numbers x.

Truth Value: False.

Rationale: y is never bigger than itself; so setting $x = y$ is a **counterexample**.

EXAMPLE

Universe = **Z**.
$P(x, y) := x = y$.

State truth values of (1) $\forall x \, \exists y \, P(x, y)$, and (2) $\exists y \, \forall x \, P(x, y)$. Explain.

Solution. Improvise.

33.3. PARSING MULTIVARIATE QUANTIFICATION

"Parsing", in the sense used here, means when evaluating a logical expression involving multivariate quantification (by hand or programmatically), the proposition is transliterated in the **same order** to English.

EXAMPLE

Logical Expression: $\exists x \, \forall y \, \exists z \, P(x, y, z)$.

English Equivalent: There is an x such that for all y there is a z such that $P(x, y, z)$ holds.

EXAMPLE

Logical Expression: $\exists x \, \forall y \, \exists z \, P(x, y, z)$, where $P(x, y, z) = y - x \geq z$.

English Equivalent 1 (Inelegant): There is an x such that for all y there is a z such that $y - x \geq z$.

English Equivalent 2 (Elegant): There is some number x which when subtracted from any number y results in a number bigger than some number z.

EXAMPLE

Let the universe of discourse for x, y, and z be the set of **non-negative integers**, that is, $\{0, 1, 2, \ldots\}$.

Find the truth value of

$$\exists x \; \forall y \; \exists z \; P(x, y, z), \text{ where } P(x, y, z) = y - x \geq z.$$

Solution.

True. For instance, set $x := 0$ and $z := 0$ (or $z := y$).

It is **not** possible to satisfy

$$\exists x \; \forall y \; \exists z \; P(x, y, z), \text{ where } P(x, y, z) = y - x \geq z,$$

by setting $x := y$ and $z := 0$ because the **scope** of x is **higher** than the scope of y. (The existence of x comes **before** one knows about y.) Indeed, with respect to y, x is like a **constant** and **cannot affect** y.

EXAMPLE

Let the universe of discourse for x and y be the set of integers.

Let $P(x, y)$ denote the statement

$$x + y = 10.$$

Find the truth value in each of the following cases:

(a) $\forall x \; \forall y \; P(x, y)$.
(b) $\exists x \; \exists y \; P(x, y)$.
(c) $\forall x \; \exists y \; P(x, y)$.
(d) $\forall x \; \exists! y \; P(x, y)$.
(e) $\exists x \; \forall y \; P(x, y)$.

Solution.

(a) $\forall x\ \forall y\ P(x, y)$.

False. For example, $x = 1$, $y = 1$.

(b) $\exists x\ \exists y\ P(x, y)$.

True. For example, $x = 1$, $y = 9$.

(c) $\forall x\ \exists y\ P(x, y)$.

True. For example, $y = 10 - x$.

(d) $\forall x\ \exists! y\ P(x, y)$.

False. It follows from (c) that y (depends on x and therefore) is not unique.

(e) $\exists x\ \forall y\ P(x, y)$.

False. For example, $x = \mathrm{n}$ and $y \neq 10 - \mathrm{n}$.

EXAMPLE

Let the universe of discourse for x, y, and z be the set of integers.

Let $P(x, y, z)$ denote the statement

$$x^2y = z.$$

Find the truth value in each of the following cases:

(a) $\forall x\ \forall y\ \forall z\ P(x, y, z)$.
(b) $\exists x\ \exists y\ \exists z\ P(x, y, z)$.
(c) $\forall y\ \forall z\ \exists x\ P(x, y, z)$.
(d) $\exists! y\ \exists! z\ \forall x\ P(x, y, z)$.
(e) $\forall x\ \forall y\ \exists z\ P(x, y, z)$.
(f) $\forall x\ \exists y\ \exists z\ P(x, y, z)$.

Solution.

(a) $\forall x \, \forall y \, \forall z \, P(x, y, z)$.

False. For example, $x = 1$, $y = 2$, $z = 3$.

(b) $\exists x \, \exists y \, \exists z \, P(x, y, z)$.

True. For example, $x = 1$, $y = 1$, $z = 1$.

(c) $\forall y \, \forall z \, \exists x \, P(x, y, z)$.

False. For example, $y = 2$, $z = 3$.

(d) $\exists! y \, \exists! z \, \forall x \, P(x, y, z)$.

True. For example, $y = 0$, $z = 0$.

(e) $\forall x \, \forall y \, \exists z \, P(x, y, z)$.

True. For example, $z = x^2 y$.

(f) $\forall x \, \exists y \, \exists z \, P(x, y, z)$.

True. For example, $y = 1$, $z = x^2$.

EXAMPLE

Let the universe of discourse for x, y, and z be the set of positive integers.

Let $P(x, y, z)$ denote the statement

$$x^2 + y^2 = z.$$

Find the truth value in each of the following cases:

(a) $\forall x\ \forall y\ \forall z\ P(x, y, z)$.
(b) $\exists x\ \exists y\ \exists z\ P(x, y, z)$.
(c) $\forall x\ \forall y\ \exists z\ P(x, y, z)$.
(d) $\forall x\ \forall z\ \exists y\ P(x, y, z)$.
(e) $\forall z\ \exists x\ \exists y\ P(x, y, z)$.

Solution.

(a) $\forall x\ \forall y\ \forall z\ P(x, y, z)$.

False. For example, $x = 1$, $y = 1$, $z = 3$.

(b) $\exists x\ \exists y\ \exists z\ P(x, y, z)$.

True. For example, $x = 1$, $y = 1$, $z = 2$.

(c) $\forall x\ \forall y\ \exists z\ P(x, y, z)$.

True. For given positive integers x and y, set $z = x^2 + y^2$.

(d) $\forall x\ \forall z\ \exists y\ P(x, y, z)$.

False. For example, there is no y for $x = 1$ and $z = 1$.

(e) $\forall z\ \exists x\ \exists y\ P(x, y, z)$.

False. For example, there is no x and y for $z = 1$.

33.4. DE MORGAN’S LAWS REVISITED

The De Morgan’s Laws **connect** conjunction and disjunction. These laws can be applied to quantifiers.

Indeed, **distributing a negation operator** across a quantifier **changes** a **universal to an existential**, and **vice versa**. This provides a **connection between universal and existential quantifications**.

Universal Negation:

$$\neg\forall x\, P(x) \Leftrightarrow \exists x\, \neg P(x)$$

Existential Negation:

$$\neg\exists x\, P(x) \Leftrightarrow \forall x\, \neg P(x)$$

EXAMPLE

Let $P(x)$ be the propositional function "x is a positive number". Let the domain consist of the set of integers. Translate each of the following statements into logical expressions, and give the truth value of each:

(a) There exists no integer that is a positive number.
(b) Not all integers are positive numbers.

Solution.

(a) $\neg\exists x\, P(x)$. False.
(b) $\neg\forall x\, P(x)$. True.

EXAMPLE

This shows the **process** of how a logical expression is **negated**.

$\neg\exists x\, \forall y\, \forall z\, P(x, y, z)$
$\Leftrightarrow \forall x\, \neg[\, \forall y\, \forall z\, P(x, y, z)\,]$
$\Leftrightarrow \forall x\, \exists y\, \neg[\, \forall z\, P(x, y, z)\,]$
$\Leftrightarrow \forall x\, \exists y\, \exists z\, \neg P(x, y, z)$.

EXAMPLE

In each of the following cases, express the negations of the statements so that all negation symbols immediately precede predicates:

(a) $\exists z\, \forall y\, \forall x\, T(x, y, z)$.
(b) $\exists x\, \exists y\, P(x, y) \wedge \forall x\, \forall y\, Q(x, y)$.
(c) $\exists x\, \exists y\, [\, Q(x, y) \leftrightarrow Q(y, x)\,]$.

Solution.

The use of the negation symbol affects the quantifiers. Indeed, as the negation symbol is moved towards the inside or outside, each quantifier it passes must change its type.

This can be done by using De Morgan's Laws, by using $\neg(p \longrightarrow q) \Leftrightarrow p \wedge \neg q$, and by using $\neg(p \leftrightarrow q) \Leftrightarrow \neg p \leftrightarrow \neg q$.

(a)

$\neg\exists z\ \forall y\ \forall x\ T(x, y, z)$
$\Leftrightarrow \forall z\ \exists y\ \exists x\ \neg T(x, y, z)$.

(b)

$\neg[\ \exists x\ \exists y\ P(x, y) \wedge \forall x\ \forall y\ Q(x, y)\ ]$
$\Leftrightarrow \forall x\ \forall y\ \neg P(x, y) \vee \exists x\ \exists y\ \neg Q(x, y)$.

(c)

$\neg[\ \exists x\ \exists y\ (Q(x, y) \leftrightarrow Q(y, x))\ ]$
$\Leftrightarrow \forall x\ \forall y\ \neg[\ Q(x, y) \leftrightarrow Q(y, x)\ ]$.

EXAMPLE

This shows the **process** of how a logical expression is **transformed**.

$\exists x\ [\ P(x) \longrightarrow Q(x)\ ]$
$\Leftrightarrow \exists x\ [\neg P(x) \vee \mathrm{Q}(x)\ ]$
$\Leftrightarrow [\ \exists x\ \neg P(x)\ ] \vee [\ \exists x\ Q(x)\ ]$, as $[\ \exists x\ P(x)\ ] \vee [\ \exists x\ Q(x)\ ] \Leftrightarrow \exists x\ [\ P(x) \vee Q(x)\ ]$
$\Leftrightarrow [\ \neg\forall x\ P(x)\ ] \vee [\ \exists x\ Q(x)\ ]$
$\Leftrightarrow [\ \forall x\ P(x)\ ] \longrightarrow [\ \exists x\ Q(x)\ ]$.

EXAMPLE

Simplify

$$\neg\forall x\ \exists y\ (x^2 \leq y).$$

Solution.

Note: $\neg\forall x\, \exists y\, (x^2 \le y)$ is the **same** as $\neg(\forall x\, \exists y\, (x^2 \le y))$.

English: Some x does not admit a y greater or equal to the square of x.

Algebraic: Use **both** De Morgan's Laws applied to quantifiers:

$\neg\forall x\, \exists y\, (x^2 \le y)$
$\Leftrightarrow \exists x\, \forall y\, \neg(x^2 \le y)$
$\Leftrightarrow \exists x\, \forall y\, (x^2 > y)$.

34. EQUIVALENCE IN PROPOSITIONAL FUNCTIONS

Two logical expressions involving propositional functions and quantifiers are said to be **logically equivalent** if the expressions always have the **same truth value**, **irrespective** of the **universe** and **propositional functions** used.

EXAMPLE

$\forall x\, \exists y\, Q(x, y)$ and $\forall y\, \exists x\, Q(y, x)$ are logically equivalent.

EXAMPLE

$\forall x\, \exists y\, Q(x, y)$ and $\exists y\, \forall x\, Q(x, y)$ are **not** logically equivalent.

EXAMPLE

Show that the following are logically equivalent:

$$[\ \forall x\, A(x)\ ] \lor [\ \forall x\, B(x)\ ] \text{ and } \forall x\, \forall y\, [\ A(x) \lor B(y)\ ].$$

(To induce reality into abstraction, it can be useful to substitute for A and B. For example, $A(x)$:= "x likes **A**nna", and $B(x)$:= "x likes **B**ritney".)

Solution.

To shows the logical equivalence, it is sufficient to show that the following is a **tautology**:

$$[\ \forall x\, A(x)\ ] \vee [\ \forall x\, B(x)\ ] \leftrightarrow \forall x\, \forall y\, [\ A(x) \vee B(y)\ ].$$

To do that, it is sufficient to show that **LHS and RHS must always have same truth values**. (In other words, if LHS is true, then RHS is true; if LHS is false, then RHS is false.)

Case I. Let LHS be true. The goal is to show that RHS is also true.

Now, either $\forall x\, A(x)$ is true, or $\forall x\, B(x)$ is true.

Case I(a). For all x, $A(x)$ is true. Now, (T $\vee$ Anything) = T. Thus, it is possible to set Anything = $B(y)$ and obtain: For all x and for all y, $A(x) \vee B(y)$. Therefore, RHS is true.

Case I(b). For all x, $B(x)$ is true. This is similar to Case I(a) by **symmetry**.

Case II. Let LHS be false. The goal is to show that RHS is also false.

Now, both $\forall x\, A(x)$ is false and $\forall x\, B(x)$ is false.

Therefore, there exists an x_1 for which $A(x_1)$ is false, and an x_2 for which $B(x_2)$ is false.

Now, $A(x_1) \vee B(x_2)$ is false, as F $\vee$ F = F.

However, setting $x = x_1$ and $y = x_2$, yields a **counterexample** to the RHS, that is, $\forall x\, \forall y\, (A(x) \vee B(y))$. Therefore, RHS is false.

EXAMPLE

Let S be a set. Let P: $S \longrightarrow \{T, F\}$ and Q: $S \longrightarrow \{T, F\}$. Show, for $S = \{1, 2\}$, that

$$\forall x\ [\ (P(x) \vee Q(x)\ ] \text{ and } \forall x\, \forall y\ [\ (P(x) \vee Q(y)\ ]$$

are **not** logically equivalent.

Solution.

This requires a counterexample.

Take

x	$P(x)$	$Q(x)$
1	T	F
2	F	T

Then,

$\forall x$ [$(P(x) \vee Q(x)$]
$\Leftrightarrow$ [$(P(1) \vee Q(1)$] $\wedge$ [$(P(2) \vee Q(2)$]
$\Leftrightarrow$ T $\wedge$ T
$\Leftrightarrow$ T.

$\forall x\ \forall y$ [$(P(x) \vee Q(y)$]
$\Leftrightarrow$ [$(P(1) \vee Q(1)$] $\wedge$ [$(P(1) \vee Q(2)$] $\wedge$ [$(P(2) \vee Q(1)$] $\wedge$ [$(P(2) \vee Q(2)$]
$\Leftrightarrow$ T $\wedge$ T $\wedge$ F $\wedge$ T
$\Leftrightarrow$ F.

REMARKS

The **moral** of the example is that $\forall x$ [$(P(x) \vee Q(x)$] and $\forall x\ \forall y$ [$(P(x) \vee Q(y)$] are **not** always equivalent.

EXAMPLE

Let S be a set. Let P: $S \longrightarrow \{\mathrm{T}, \mathrm{F}\}$ and Q: $S \longrightarrow \{\mathrm{T}, \mathrm{F}\}$. Show that

$$[\ \forall x\, P(x)\ ] \vee [\ \forall x\, Q(x)\ ] \text{ and } \forall x\ [\ P(x) \vee Q(x)\ ]$$

are **not** logically equivalent.

Solution. This requires a counterexample.

Let

$$P\colon \mathbf{Z}^{+} \longrightarrow \{\mathrm{T}, \mathrm{F}\} \text{ be defined by } P(x) := \text{“}x \text{ is even”},$$

and

$$Q\colon \mathbf{Z}^{+} \longrightarrow \{\mathrm{T}, \mathrm{F}\} \text{ be defined by } Q(x) := \text{“}x \text{ is odd”}.$$

Then, $[\ \forall x\ P(x)\ ] \vee [\ \forall x\ Q(x)\ ]$ is false, and $\forall x\ [\ P(x) \vee Q(x)\ ]$ is true.

EXAMPLE

Let S be a set. Let $P\colon S \longrightarrow \{\mathrm{T}, \mathrm{F}\}$ and $Q\colon S \longrightarrow \{\mathrm{T}, \mathrm{F}\}$. Determine whether

$$[\ \forall x\ P(x)\ ] \leftrightarrow [\ \forall x\ Q(x)\ ] \text{ and } \forall x\ [\ P(x) \leftrightarrow Q(x)\ ]$$

are logically equivalent.

Solution.

They are **not** logically equivalent.

P and Q need to be such that they are **sometimes, but not always, true**, so that the first biconditional is F $\leftrightarrow$ F and hence true, but such that there is an x making one true and the other false, so that the second biconditional has T $\leftrightarrow$ F and hence false.

Let S be $\mathbf{Z}^{+}$, the set of **positive integers**.

Let

$$P(x) := \text{“}x \text{ is a multiple of 2”},$$

and

$$Q(x) := \text{“}x \text{ is a multiple of 3”}.$$

Then,

$$[\ \forall x\ P(x)\ ] \leftrightarrow [\ \forall x\ Q(x)\ ] \text{ is true.}$$

However, for $x = 4$,

$$\forall x\ [\ P(x) \leftrightarrow Q(x)\ ] \text{ is false.}$$

EXAMPLE

Let S be a set. Let P: $S \longrightarrow \{\mathrm{T}, \mathrm{F}\}$ and Q: $S \longrightarrow \{\mathrm{T}, \mathrm{F}\}$. Show that

$$[\ \exists x\ P(x)\ ] \wedge [\ \exists x\ Q(x)\ ] \text{ and } \exists x\ [\ P(x) \wedge Q(x)\ ]$$

are **not** logically equivalent.

Solution.

Let

$$P\colon \mathbf{Z}^+ \longrightarrow \{\mathrm{T}, \mathrm{F}\} \text{ be defined by } P(x) := \text{“}x \text{ is even”},$$

and

$$Q\colon \mathbf{Z}^+ \longrightarrow \{\mathrm{T}, \mathrm{F}\} \text{ be defined by } Q(x) := \text{“}x \text{ is odd”}.$$

Then, $[\ \exists x\ P(x)\ ] \wedge [\ \exists x\ Q(x)\ ]$ is true, and $\exists x\ [\ P(x) \wedge Q(x)\ ]$ is false.

34.1. A COMPENDIUM OF LOGICAL EQUIVALENCES

Let S be a set. Let P: $S \longrightarrow \{\mathrm{T}, \mathrm{F}\}$ and Q: $S \longrightarrow \{\mathrm{T}, \mathrm{F}\}$. Table 3 lists commonly-used logical equivalences.

$$
\begin{aligned}
\neg[\,\exists x\, P(x)\,] \quad &\Leftrightarrow \quad \forall x\,[\,\neg P(x)\,] \\
\neg[\,\forall x\, P(x)\,] \quad &\Leftrightarrow \quad \exists x\,[\,\neg P(x)\,] \\
\\
[\,\forall x\, P(x)\,] \wedge [\,\forall x\, Q(x)\,] \quad &\Leftrightarrow \quad \forall x\,[\,P(x) \wedge Q(x)\,] \\
[\,\exists x\, P(x)\,] \vee [\,\exists x\, Q(x)\,] \quad &\Leftrightarrow \quad \exists x\,[\,P(x) \vee Q(x)\,] \\
\\
[\,\forall x\, P(x)\,] \wedge [\,\forall x\, Q(x)\,] \quad &\Leftrightarrow \quad \forall x\,\forall y\,[\,P(x) \wedge Q(y)\,] \\
[\,\exists x\, P(x)\,] \vee [\,\exists x\, Q(x)\,] \quad &\Leftrightarrow \quad \exists x\,\exists y\,[\,P(x) \vee Q(y)\,] \\
\\
[\,\forall x\, P(x)\,] \vee [\,\forall x\, Q(x)\,] \quad &\Leftrightarrow \quad \forall x\,\forall y\,[\,P(x) \vee Q(y)\,] \\
[\,\exists x\, P(x)\,] \wedge [\,\exists x\, Q(x)\,] \quad &\Leftrightarrow \quad \exists x\,\exists y\,[\,P(x) \wedge Q(y)\,]
\end{aligned}
$$

Table 3. A collection of some commonly-used logical equivalences.

EXAMPLE

Let $P\colon S \longrightarrow \{\mathrm{T}, \mathrm{F}\}$ and $Q\colon S \longrightarrow \{\mathrm{T}, \mathrm{F}\}$. Show that

$$[\,\forall x\, P(x)\,] \wedge [\,\exists x\, Q(x)\,] \Leftrightarrow \forall x\,\exists y\,[\,P(x) \wedge Q(y)\,].$$

Solution.

A proof requires showing

$$\text{LHS} \Rightarrow \text{RHS}$$

and

$$\text{RHS} \Rightarrow \text{LHS (or, equivalently and contrapositively, } \neg\text{LHS} \Rightarrow \neg\text{RHS).}$$

(1) LHS ⇒ RHS.

If the LHS is true, then $P(x)$ is true for all x, and $Q(x_0)$ is true for some x, say, x_0. Then, choosing $y = x_0$, it can be seen that the RHS is also true.

(2) ¬LHS ⇒ ¬RHS.

To show this is the same as showing

$$[\,\exists x\, \neg P(x)\,] \vee [\,\forall x\, \neg Q(x)\,] \Rightarrow \exists x\,\forall y\,[\,\neg P(x) \vee \neg Q(y)\,].$$

As $\neg P$ and $\neg Q$ are arbitrary, they can be replaced by P and Q, respectively. Thus,

$$[\ \exists x\ P(x)\ ] \vee [\ \forall x\ Q(x)\ ] \Longrightarrow \exists x\ \forall y\ [\ P(x) \vee Q(y)\ ].$$

EXAMPLE

Give the truth value of the following:

$\forall x \in \mathbf{Z}^+$, $\exists y \in \mathbf{Z}^+$, and $\exists z \in \mathbf{Z}^+$, such that

$$yz^2 = x^3.$$

Solution.

True.

It could be noted that if x is a positive integer, then so is x^3. Next, given any x, one could set $y = x$ and $z = x$, so that LHS = RHS.

EXAMPLE

Give the truth value of the following:

$\forall z \in \mathbf{Z}$, $\exists x \in \mathbf{Z}^+$, and $\exists y \in \mathbf{Z}^+$, such that

$$x^2 + y = z^3.$$

Solution.

False.

Let $z = 0$. Then, RHS = 0, but LHS can never be 0.

EXAMPLE

Give the truth value of the following:

$\forall x \in \mathbf{Z}^+, \exists y \in \mathbf{R}$, such that

$$x = y\sqrt{3}.$$

Solution.

True.

For any positive integer x, it is possible to (find a real number y and) set $y = x/\sqrt{3}$.

EXAMPLE

Give the truth value of the following:

$\forall x \in \mathbf{R}, \exists y \in \mathbf{R}$, such that

$$xy = 1.$$

Solution.

True.

EXAMPLE

Let the domain be the set of integers. Give the truth value of the following:

(a) $\exists! x\ (x = x + 1)$.
(b) $\exists! x\ (x^2 = 1)$.
(c) $\exists! x\ (x > 1)$.

Solution.

(a) False.
(b) False.
(c) False.

(It could be noted that each is false for a **different reason in each case**.)

EXAMPLE

Express each of the following statements in predicate logic:

(a) "There is a unique x for which $P(x)$ is true."
(b) "There is no greatest integer."
(c) "Every person has exactly two parents."

Solution.

(a)

$\exists! x\ P(x)$.

(b)

Here is an **equivalent statement** that can lead towards a solution: "There is no integer that is greater than all other integers."

Universe = **Z**.
$\forall x\ \exists y\ (y \geq x) \equiv \neg\exists x\ \forall y\ (y < x)$.

(c)

Let $P(x, y)$ denote "x is a parent of y".

There are three people, each (pairwise) **different** from the others:

$$\exists x_1\ \exists x_2\ \exists y \text{ such that } (x_1 \neq x_2),\ (x_1 \neq y), \text{ and } (x_2 \neq y).$$

There is a child (person) and two parents:

$$P(x_1, y) \text{ and } P(x_2, y).$$

There are exactly two parents (and they are x_1 and x_2):

$$\forall z\ P(z, y) \longrightarrow ((z = x_1) \vee (z = x_2)).$$

Therefore,

$$\exists x_1 \, \exists x_2 \, \exists y$$
$$((x_1 \neq x_2) \wedge (x_1 \neq y) \wedge (x_2 \neq y)) \wedge P(x_1, y) \wedge P(x_2, y) \wedge [\forall z \, P(z, y) \longrightarrow ((z = x_1) \vee (z = x_2))].$$

EXAMPLE

Translate each of the following into equivalent English statements. The domain for each variable consists of real numbers.

(a) $\forall x \, \forall y \, [\, (x \geq 0) \wedge (y < 0) \longrightarrow (x - y) > 0 \,]$.
(b) $\forall x \, \forall y \, \exists z \, (x = y + z)$.

Solution.

(a) A nonnegative number minus a negative number is a positive number.

(b) For every real number x and real number y, there exists a real number z such that $x = y + z$.

EXAMPLE

Translate each of the following nested quantifications into an English statement that expresses a mathematical fact. The domain in each case consists of real numbers.

(a) $\exists x \, \forall y \, (x + y = y)$.
(b) $\exists x \, \exists y \, [\, (x \leq 0) \wedge (y \leq 0) \wedge (x - y) > 0 \,]$.
(c) $\forall x \, \forall y \, [\, (x \neq 0) \wedge (y \neq 0) \longleftrightarrow (xy \neq 0) \,]$.

Solution.

(a) There exists a number that when added to every number does not change its value. (Or, alternatively, there is an **additive identity** for the real numbers. In fact, 0 is such an identity.)

(b) The difference of two nonpositive numbers is not necessarily nonpositive. (Or, alternatively, the difference of two nonpositive numbers can be positive. In fact, let $x = 0$ and let $y = -1$.)

(c) The product of two numbers is nonzero if and only if both numbers are nonzero.

EXAMPLE

Determine the truth value of each of these statements if the domain of each variable consists of real numbers.

(a) $\forall x\, \exists y\, (x^2 = y)$.
(b) $\forall x\, \exists y\, (x = y^2)$.
(c) $\forall x\, \exists y\, [\, (x + y = 2) \wedge (2x - y = 1)\,]$.
(d) $\exists x\, \forall y\, (y \neq 0 \longrightarrow xy = 1)$.
(e) $\forall x\, \forall y\, \exists z\, (x + y = 2z)$.

Solution.

(a) True. (It is always possible find such a y by setting $y = x^2$.)

(b) False. (If x is a negative number, then no such y exists.)

(c) False. The (linear) system

$$x + y = 2, \text{ and } 2x - y = 1,$$

has a **unique solution**, $x = 1$ and $y = 1$. (This is because each equation represents a straight line that can intersect **only once** with the other, as shown in Figure 16.) Therefore, the statement is false for any other value of x.

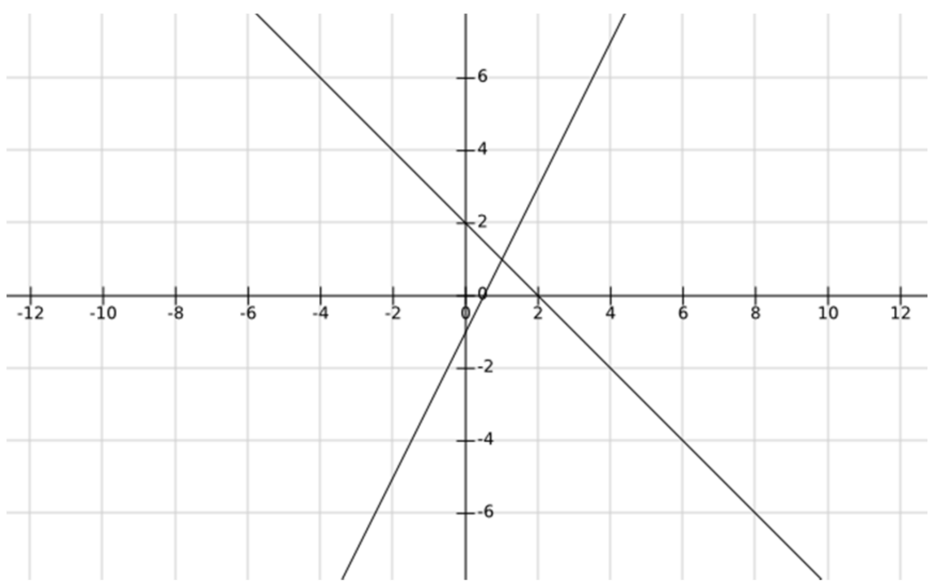

Figure 16. The graph of $y = 2 - x$ and $y = 2x - 1$.

(d) False. (The reciprocal of y depends on y. Therefore, there is no single x that works for all y.)

(e) True. (Let $z = (x + y)/2$. In fact, such a z always exists as it is simply the **arithmetic mean** of x and y.)

EXAMPLE

Find a counterexample, if possible, to these universally quantified statements, where the domain for all variables consists of all integers.

(a) $\forall x\, \exists y\, (y^2 - x < 100)$.
(b) $\forall x\, \forall y\, (x^2 \neq y^3)$.

Solution.

(a) $y^2 - x < 100$ means $y^2 < 100 + x$. For $x = -100$, this gives $y^2 < 0$, which is not possible, and serves as a counterexample.

(b) It could be noted that the sixth power of a number is both a square and a cube. Therefore, $x = 0$ and $y = 0$ (or $x = 1$ and $y = 1$, or $x = 8$ and $y = 4$) serves as a counterexample.

EXAMPLE

Let $F(x, y)$ be the statement "x can fool y". Let the domain consists of all people in the world. Use quantifiers to express each of these statements.

(a) There is no one who can fool everybody.
(b) Everyone can be fooled by somebody.
(c) No one can fool both Nazlie and James.
(d) Nancy can fool exactly two people.

Solution.

(a) $\exists x\, \forall y\, F(x, y)$ means that there is someone who can fool everybody. Therefore, the solution is its negation

$$\neg\exists x\, \forall y\, F(x, y).$$

(b) $\forall y \, \exists x \, F(x, y)$.

(c) $\exists x \, (F(x, \text{Nazlie}))$ means that there is an x who can fool Nazlie, and $\exists x \, (F(x, \text{James}))$ means that there is an x who can fool James. Therefore, $\exists x \, [\, F(x, \text{Nazlie}) \wedge (F(x, \text{James}) \,]$ means that there is an x who can fool both Nazlie and James.

Thus, the solution is its negation

$$\neg \exists x \, (F(x, \text{Nazlie}) \wedge (F(x, \text{James})).$$

(d) Let y_1 and y_2 be the two people who Nancy can fool. In other words,

$$\exists y_1 \, (F(\text{Nancy}, y_1) \,] \wedge \exists y_2 \, [\, F(\text{Nancy}, y_2))$$

and

$$y_1 \neq y_2.$$

Moreover, there cannot be other people whom Nancy can fool. In fact, if there are such people, then those people are either y_1 or y_2. In other words,

$$\forall y \, (F(\text{Nancy}, y) \longrightarrow (y = y_1) \vee (y = y_2)).$$

Therefore, the solution is the **combination** of all of the above, that is,

$$\exists y_1 \, \exists y_2$$
$$[(F(\text{Nancy}, y_1) \wedge F(\text{Nancy}, y_2) \wedge (y_1 \neq y_2)) \wedge \forall y \, (F(\text{Nancy}, y) \longrightarrow (y = y_1) \vee (y = y_2))].$$

EXAMPLE

Let $I(x)$ be the statement "x has an Internet connection" and $C(x, y)$ be the statement "x and y have chatted over the Internet". Let the domain for the variables x and y consists of all students in your class. Use quantifiers to express each of these statements.

(a) No one in the class has chatted with Mary.
(b) Steven has chatted with everyone except Peter.
(c) Someone in your class does not have an Internet connection.
(d) Not everyone in your class has an Internet connection.
(e) Exactly one student in your class has an Internet connection.

(f) Everyone in your class with an Internet connection has chatted over the Internet with at least one other student in your class.

Solution.

(a)

$$\neg \exists x\, C(x, \text{Mary}).$$

(b)

If Steven has chatted, then it is with everyone except Peter. In other words,

$$\forall y\, C(\text{Steven}, y) \longrightarrow (y \neq \text{Peter}).$$

If everyone but Peter have chatted, then it is with Steven. In other words,

$$(y \neq \text{Peter}) \longrightarrow \forall y\, C(\text{Steven}, y).$$

Therefore, the solution is the combination of all of the above, that is,

$$\forall y\, (C(\text{Steven}, y) \longleftrightarrow (y \neq \text{Peter})).$$

(c)

$$\exists x\, \neg I(x).$$

(d)

$$\neg \forall x\, I(x).$$

(Thus, (c) and (d) are **identical**.)

(e)

$\exists x\, I(x)$ means that there is a student x in the class who has an Internet connection. $\forall y\, I(y)$ means that every student in the class has an Internet connection. Now, to make sure that x is the **only** such student in the class, one can set $x = y$.

Therefore, the solution is the combination of all of the above, that is,

$$\exists x\ \forall y\ [\ I(y) \leftrightarrow (x = y)\ ].$$

(f)

It is important to note that **not every** student has an Internet connection, **only** those students who have an Internet connection can chat, and one **cannot** chat with himself/herself.

Therefore,

$$\forall x\ [\ I(x) \longrightarrow \exists y\ (\ (x \neq y) \wedge C(x, y)\ )\ ].$$

35. EXTENSIONS OF 'CONVENTIONAL' LOGIC

The 'conventional' or binary logic is limited by its assumption that the world is objective (in which a proposition about the state of the world must be either true or false). This limitation is overcome by **subjective logic** [Jøsang, 2016].

36. LOGICAL FALLACIES

[3]

There are several types of **irrational arguments** [Bunch, 1982; Pirie, 2006; Howard-Snyder, Howard-Snyder, Wasserman, 2009, Chapter 4; Reba, Shier, 2015, Section 7.4].

The presence of a fallacy can make an argument unfair [Swart, 2018, Chapter 10] and can contaminate communication [Hewitt, 2019, Chapter 7].

[3] URL: https://yourlogicalfallacyis.com/ .

36.1. CLASSIFICATION OF FALLACIES

In [Pirie, 2006], fallacies have been classified into the following categories:

1. Formal
2. Informal (Linguistic)
3. Informal (Relevance - Omission)
4. Informal (Relevance - Intrusion)
5. Informal (Relevance - Presumption)

In other words, fallacies have been classified broadly into formal fallacies and informal fallacies [Swart, 2018, Chapter 10].

A **logical fallacy** is a kind of formal fallacy.

A **logical fallacy** is "a pattern of reasoning rendered invalid by a flaw in its logical structure that can neatly be expressed in a standard logic system, for example propositional logic" [Wikipedia].

In other words, a logical fallacy is an **error in reasoning**.

36.2. EXAMPLES OF LOGICAL FALLACIES

There are several types of logical fallacies[4] [Bennett, 2012; Almossawi, 2013; Copi, Cohen, McMahon, 2014, Chapter 4; Swart, 2018, Chapter 10; Hewitt, 2019, Chapter 7]. The following is a selective subset:

- **Ad Hominem:** In this case, the person making the argument is attacked, rather than the argument itself, when the attack on the person is completely irrelevant to the argument the person is making.

- **Appeal to Authority:** In this case, an authority is used as evidence in a person's argument when the authority is not really an authority on the facts relevant to the argument.

- **Appeal to Flattery:** In this case, an attempt is made to win support for an argument not by the strength of the argument, but by using flattery on those whom a person wants to accept his or her argument.

[4] URL: http://www.logicalfallacies.info/ .

- **Appeal to Novelty:** In this case, a claim is made that something that is new or modern is superior to the status quo, based exclusively on its newness. (Logical Form: X has been around for years now. Y is new. Therefore, Y is better than X.)

- **Appeal to Popularity:** In this case, popularity of a premise or proposition is used as evidence for its goodness or truthfulness. However, adopting it ignores the possibility that today clever marketing, social and political weight, and money can 'buy' popularity. Indeed, the use of this logical fallacy is not uncommon in advertisements to **persuade people** [Lieto, Vernero, 2013].

- **Argument by Repetition:** In this case, an argument or a premise is repeated over and over in place of more supporting evidence. (Logical Form: X is true. X is true. X is true. …)

- **Cherry Picking:** In this case, only select evidence is presented in order to persuade the audience to accept a position, and evidence that would go against the position is withheld.

- **Circular Reasoning:** In this case, a proposition is supported by the premises, which is supported by the proposition, creating a circle in reasoning where no useful information is being shared.

The Top Ten Logical Fallacies And How People Misuse Them[5] sheds light on how some of the common logical fallacies occur in routine activities, and are misunderstood and misapplied.

FALLACY FILES

The Fallacy Files[6] provides a taxonomy, a glossary, and a regularly updated blog on logical fallacies.

[5] URL: https://www.youtube.com/watch?v=cCl0Ua2_itc .
[6] URL: http://www.fallacyfiles.org/ .

37. LIMITATIONS OF LOGIC

Logic will get you from A to B. Imagination will take you everywhere.
— Albert Einstein

A computer scientist or a software engineer must be aware of the limitations of logic so that logic is not misapplied.

It is not possible to model all problems mathematically and to explain all phenomena logically. For example, this is the case with the so-called **'wicked problems'** [Rittel, 1972].

ACKNOWLEDGEMENT

This document has benefited from the Lecture Notes on Discrete Mathematics by Eusebius Doedel, especially from the examples therein, and from the remarks by David Ford. The inclusion of images from external sources is only for non-commercial educational purposes, and their use is hereby acknowledged.

REFERENCES

[Almossawi, 2013] An Illustrated Book of Bad Arguments. By A. Almossawi. JasperCollins Publishers. 2013.

[Ammann, Offutt, 2008] Introduction to Software Testing. By P. Ammann, J. Offutt. Cambridge University Press. 2008.

[Baldwin, Walker, Henderson, 2013] The Roles of Mathematics in Computer Science. By D. Baldwin, H. M. Walker, P. B. Henderson. ACM Inroads. Volume 4. Issue 4. 2013. Pages 74-80.

[Bassham, Irwin, Nardone, Wallace, 2011] Critical Thinking: A Student’s Introduction. By G. Bassham, W. Irwin, H. Nardone, J. M. Wallace. Fourth Edition. McGraw-Hill. 2011.

[Belcastro, 2012] Discrete Mathematics with Ducks. By S.-M. Belcastro. CRC Press. 2012.

[Ben-Ari, 2012] Mathematical Logic for Computer Science. By M. Ben-Ari. Third Edition. Springer-Verlag. 2012.

[Bennett, 2012] Logically Fallacious: The Ultimate Collection of Over 300 Logical Fallacies. By B. Bennett. eBookIt. 2012.

[Berry, Kamsties, 2005] The Syntactically Dangerous All and Plural in Specifications. By D. M. Berry, E. Kamsties. IEEE Software. Volume 22. Issue 1. 2005. Pages 55-57.

[Bramanti, Travaglini, 2018] Studying Mathematics: The Beauty, the Toil and the Method. By M. Bramanti, G. Travaglini. Springer International Publishing. 2018.

[Bunch, 1982] Mathematical Fallacies and Paradoxes. By B. Bunch. Dover Publications. 1982.

[Clark, 2012] Paradoxes from A to Z. By M. Clark. Third Edition. Routledge. 2012.

[Cook, 2013] Paradoxes. By R. T. Cook. Polity Press. 2013.

[Copi, Cohen, McMahon, 2014] Introduction to Logic. By I. M. Copi, C. Cohen, K. McMahon. Fourteenth Edition. Pearson Education. 2014.

[Cormen, Leiserson, Rivest, Stein, 2009] Introduction to Algorithms. By T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Third Edition. The MIT Press. 2009.

[Cunningham, 2012] A Logical Introduction to Proof. By D. W. Cunningham. Springer Science+Business Media. 2012.

[Dooley, 2011] Software Development and Professional Practice. By J. Dooley. Apress. 2011.

[Genesereth, Kao, 2013] Introduction to Logic. By M. Genesereth, E. Kao. Second Edition. Morgan and Claypool. 2013.

[Genesereth, Kao, 2017] Introduction to Logic. By M. Genesereth, E. Kao. Third Edition. Morgan and Claypool. 2017.

[Gerstein, 2012] Introduction to Mathematical Structures and Proofs. By L. J. Gerstein. Springer Science+Business Media. 2012.

[Gervasi, Zowghi, 2005] Reasoning About Inconsistencies in Natural Language Requirements. By V. Gervasi, D. Zowghi. ACM Transactions on Software Engineering and Methodology. Volume 14. Number 3. 2005. Pages 277-330.

[Gonzalez-Perez, 2018] Information Modelling for Archaeology and Anthropology: Software Engineering Principles for Cultural Heritage. By C. Gonzalez-Perez. Springer International Publishing. 2018.

[Govier, 2014] A Practical Study of Argument. By T. Govier. Enhanced Seventh Edition. Cengage Learning. 2014.

[Grieser, 2018] Exploring Mathematics: Problem-Solving and Proof. By D. Grieser. Springer International Publishing. 2018.

[Haan, Koppelaars, 2007] Applied Mathematics for Database Professionals. By L. de Haan, T. Koppelaars. Apress. 2007.

[Hammack, 2013] Book of Proof. By R. Hammack. Second Edition. Virginia Commonwealth University. Richmond, U.S.A. 2013.

[Hart, Sandefur, 2018] Teaching and Learning Discrete Mathematics Worldwide: Curriculum and Research. By E. W. Hart, J. Sandefur (Editors). Springer International Publishing. 2018.

[Hewitt, 2019] Technology Strategy Patterns: Architecture as Strategy. By E. Hewitt. O'Reilly Media. 2019.

[Houston, 2009] How to Think Like a Mathematician: A Companion to Undergraduate Mathematics. By K. Houston. Cambridge University Press. 2009.

[Howard-Snyder, Howard-Snyder, Wasserman, 2009] The Power of Logic. By F. Howard-Snyder, D. Howard-Snyder, R. Wasserman. Fourth Edition. McGraw-Hill. 2009.

[Huth, Ryan, 2004] Logic in Computer Science: Modelling and Reasoning about Systems. By M. Huth, M. Ryan. Second Edition. Cambridge University Press. 2004.

[IEEE, 2014] Guide to the Software Engineering Body of Knowledge (SWEBOK) Version 3.0. The Institute of Electrical and Electronics Engineers (IEEE) Computer Society. 2014.

[ISO/IEC/IEEE, 2011] ISO/IEC/IEEE 29148:2011. Systems and Software Engineering -- Life Cycle Processes -- Requirements Engineering. The International Organization for Standardization (ISO)/The International Electrotechnical Commission (IEC)/The Institute of Electrical and Electronics Engineers (IEEE) Computer Society. 2011.

[ISO/IEC/IEEE, 2018] ISO/IEC/IEEE 29148:2018. Systems and Software Engineering -- Life Cycle Processes -- Requirements Engineering. The International Organization for Standardization (ISO)/The International Electrotechnical Commission (IEC)/The Institute of Electrical and Electronics Engineers (IEEE) Computer Society. 2018.

[Jorgensen, 2014] Software Testing: A Craftsman's Approach. By P. C. Jorgensen. Fourth Edition. CRC Press. 2014.

[Jøsang, 2016] Subjective Logic: A Formalism for Reasoning Under Uncertainty. By A. Jøsang. Springer International Publishing. 2016.

[Kernighan, Pike, 1999] The Practice of Programming. By B. Kernighan, R. Pike. Addison-Wesley. 1999.

[Kernighan, Plauger, 1978] The Elements of Programming Style. By B. Kernighan, P. J. Plauger. Second Edition. McGraw-Hill. 1978.

[Krantz, 1998] A Primer of Mathematical Writing: Being a Disquisition on Having Your Ideas Recorded, Typeset, Published, Read, and Appreciated. By S. G. Krantz. The American Mathematical Society. 1998.

[Kurgalin, Borzunov, 2018] The Discrete Math Workbook: A Companion Manual for Practical Study. By S. Kurgalin, S. Borzunov. Springer International Publishing. 2018.

[Laplante, 2018] Requirements Engineering for Software and Systems. By P. A. Laplante. Third Edition. CRC Press. 2018.

[Lehman, Leighton, Meyer, 2012] Mathematics for Computer Science. By E. Lehman, F. T. Leighton, A. R. Meyer. January 4, 2012.

[Lieto, Vernero, 2013] Unveiling the Link Between Logical Fallacies and Web Persuasion. By A. Lieto, F. Vernero. The Fifth ACM Web Science Conference (WebSci 2013). Paris, France. May 1-5, 2013.

[Levin, 2016] Discrete Mathematics: An Open Introduction. By O. Levin. Second Edition. University of Northern Colorado. Greely, U.S.A. 2016.

[Lover, 2008] Elementary Logic: For Software Development. By R. Lover. Springer-Verlag. 2008.

[Łukowski, 2011] Paradoxes. By P. Łukowski. Springer Science+Business Media. 2011.

[Makinson, 2012] Sets, Logic and Maths for Computing. By D. Makinson. Second Edition. Springer-Verlag. 2012.

[Mclnerny, 2004] Being Logical: A Guide to Good Thinking. By D. Q. Mclnerny. Random House. 2004.

[Mendelson, 2015] Introduction to Mathematical Logic. By E. Mendelson. Sixth Edition. CRC Press. 2015.

[Moller, Struth, 2013] Modelling Computing Systems: Mathematics for Computer Science. By F. Moller, G. Struth. Springer-Verlag. 2013.

[Naik, Tripathy, 2008] Software Testing and Quality Assurance: Theory and Practice. By S. Naik, P. Tripathy. John Wiley and Sons. 2008.

[O'Donnell, Hall, Page, 2006] Discrete Mathematics Using a Computer. By J. O'Donnell, C. Hall, R. Page. Second Edition. Springer-Verlag. 2006.

[O'Regan, 2013] Mathematics in Computing: An Accessible Guide to Historical, Foundational and Application Contexts. By G. O'Regan. Springer-Verlag. 2013.

[O'Regan, 2016] Guide to Discrete Mathematics: An Accessible Introduction to the History, Theory, Logic and Applications. By G. O'Regan. Springer International Publishing. 2016.

[Pace, 2012] Mathematics of Discrete Structures for Computer Science. By G. J. Pace. Springer-Verlag. 2012.

[Parnas, Chik-Parnas, 2005] Goals for Software Engineering Student Education. By D. Parnas, L. Chik-Parnas. ACM SIGSOFT Software Engineering Notes. Volume 30. Number 4. 2005. Page 6-7.

[Posamentier, Thaller, 2015] Numbers: Their Tales, Types, and Treasures. By A. S. Posamentier, B. Thaller. Prometheus Books. 2015.

[Pratt, 2010] Logic: Inquiry, Argument, and Order. By S. L. Pratt. John Wiley and Sons. 2010.

[Pirie, 2006] How to Win Any Argument: The Use and Abuse of Logic. By M. Pirie. Continuum International Publishing. 2006.

[Reba, Shier, 2015] Puzzles, Paradoxes, and Problem Solving: An Introduction to Mathematical Thinking. By M. A. Reba, D. R. Shier. CRC Press. 2015.

[Rittel, 1972] On the Planning Crisis: Systems Analysis of the First and Second Generation. By H. W. J. Rittel. Bedriftsøkonomen. Number 8. 1972. Pages 390-396.

[Roberts, 2010] Introduction to Mathematical Proofs: A Transition. By C. E. Roberts, Jr. CRC Press. 2010.

[Roman, 2018] Thinking-Driven Testing: The Most Reasonable Approach to Quality Control. By A. Roman. Springer International Publishing. 2018.

[Rosen, 2012] Discrete Mathematics and Its Applications. By K. H. Rosen. Seventh Edition. McGraw-Hill. 2012.

[Scheinerman, 2013] Mathematics: A Discrete Introduction. By E. R. Scheinerman. Third Edition. Brooks/Cole. 2013.

[Sebesta, 2012] Concepts of Programming Languages. By R. W. Sebesta. Tenth Edition. Addison-Wesley. 2012.

[Seidl, Scholz, Huemer, Kappel, 2015] UML @ Classroom: An Introduction to Object-Oriented Modeling. By M. Seidl, M. Scholz, C. Huemer, G. Kappel. Springer International Publishing. 2015.

[Smullyan, 2009] Logical Labyrinths. By R. M. Smullyan. A K Peters. 2009.

[Sundstrom, 2018] Mathematical Reasoning: Writing and Proof. By T. Sundstrom. Version 2.1. Grand Valley State University. Allendale, U.S.A. October 16, 2018.

[Swart, 2018] Philosophical and Mathematical Logic. By H. de Swart. Springer Nature. 2018.

[Vatsa, Vatsa, 2009] Discrete Mathematics. By B. S. Vatsa, S. Vatsa. Fourth Edition. New Age International. 2009.

[Vince, 2015] Foundation Mathematics for Computer Science: A Visual Approach. By J. Vince. Springer International Publishing. 2015.

[Wallwork, 2014] User Guides, Manuals, and Technical Writing: A Guide to Professional English. By A. Wallwork. Springer. 2014.

[Wnuka, Gorschek, Zahda, 2013] Obsolete Software Requirements. By K. Wnuka, T. Gorschek, S. Zahda. Information and Software Technology. Volume 55. 2013. Pages 921-940.