1 Logic

In everyday life the word "logical" is often used in a rather loose sense as meaning sensible or obvious. The scientific (or rather philosophical) meaning of the word is slightly different: logic is the study of the validity of reasonings or arguments. Invalid arguments may lead to wrong conclusions, so it is important to avoid them, especially when we aim for exactness as in mathematics, computing or science. Concepts from logic are also used in commands for querying a search engine or a database, in computer programming and in the building of circuits on computer chips.

In this section we will consider several ideas and techniques from logic. Logic tells us which arguments are valid, not by looking at the contents of the argument, but rather at its form. The strategy is to break the argument down into small steps, each of which should by themselves be a valid argument.

1.1 Logical Arguments

An argument can be thought of as a sequence of statements, where each next statement is a consequence of the previous one(s). The first statement is called the *premise* (or assumption), or premises if the argument starts with multiple statements. The final statement is the *conclusion*, or conclusions if there are more than one. Whether the premise or the conclusion is true or not depends on the contents of the specific statements, which is not the concern of logic. However, logic does tell us something else: if the argument is valid, i.e. the sequence of statements follows the rules of logic, then truth is preserved, i.e. the conclusion is true if the premise is true.

Example 1.1 *The following logical argument is valid:*

> *If Homer Simpson has a yellow skin, then Homer Simpson has jaundice.*
> *Homer Simpson has a yellow skin.*
> *Therefore, Homer Simpson has jaundice.*

The following argument is not valid[1]*:*

> *Sylvia and Robert are both hard-working students.*
> *The first year university exams are not too difficult.*
> *Therefore, Sylvia and Robert will pass their first year exams.*

What makes the first argument in Example 1.1 logically valid is not so much the contents of the statements, but the structure of the argument. Here is another argument with the same structure:

> If the moon is made of green cheese, then the moon is edible.
> The moon is made of green cheese.
> Therefore, the moon is edible.

Here the conclusion is clearly not true, but this is not because the argument is invalid, but rather because one of the premises is not true (the moon is not made of green cheese).

To clarify the structure of the first argument in Example 1.1 we first introduce the labels p and q for the following two statements:

> p: Homer Simpson has a yellow skin.
> q: Homer Simpson has jaundice.

[1] Any resemblance to actual students is entirely coincidental.

The structure of the argument can then be written more abstractly as follows:

> If p is true, then q is true.
> p is true.
> Therefore, q is true.

The structure of this example is called *modus ponens*, or the *asserting method*. Any argument with this structure is valid, i.e. the conclusion is true if both of the premises are true. Modus ponens is one of the most common and basic structures of a valid logical argument.

Note that the structure of modus ponens is completely independent of the statements that we substitute for p and q, as long as they are statements that can be true or not. We might just as well take

> p: The moon is made of green cheese.
> q: The moon is edible.

In order to discuss the abstract logical structure rather than one specific logical argument, it is useful to treat p and q as *statement variables*, which denote some statement that can be specified later. All we need to know is that the statements that p or q stand for can be true or not. (This is in analogy to mathematics, where we use x to denote a number, without specifying which number. This allows us to do computations with x without knowing which number it is.)

Another structure of a valid logical argument is called *modus tollens*, or the *denying method*. It's structure is as follows:

> If p is true, then q is true.
> q is not true.
> Therefore, p is not true.

Just as for modus ponens, this structure can be made into a specific logical argument by substituting appropriate statements for the statement variables p and q.

Example 1.2 *The following logical argument is valid:*

> *If Homer Simpson is sunburnt, then Homer Simpson has a red skin.*
> *Homer Simpson does not have a red skin.*
> *Therefore, Homer Simpson is not sunburnt.*

The following argument, however, is not valid:

> *If I cut my finger, then I am in pain.*
> *I did not cut my finger.*
> *Therefore I am not in pain.*

I could still be in pain for other reasons, e.g. because I hurt my foot, so this argument must be invalid. Let us clarify the structure of this argument:

> *p: I cut my finger.*
> *q: I am in pain.*
> *If p is true, then q is true.*
> *p is not true.*
> *Therefore q is not true.*

Note that the structure of this argument is neither modus ponens nor modus tollens.

Warning: Make sure that you do not swap the roles of p and q in the second and third lines of modus ponens or modus tollens. This leads to an argument with an invalid structure, as in the second argument in Example 1.2.

Some terminology that you may come across when reading books on propositional logic: a *syllogism* is a argument consisting of two permises and a conclusion. In modus ponens and modus tollens the first premise is called the *major premise* ("If p is true, then q is true.") and the second is called the *minor premise*.

1.2 Propositional Logic

The most basic branch of logic describes arguments in terms of propositions and is called propositional logic. Modus ponens and modus tollens are examples of this, where the statement variables p and q stand for statements that are called propositions. Propositional logic will get us a long way to avoid invalid arguments and for computing applications.

1.2.1 Propositions

A *proposition* is a statement that is either true or false, but not both. We all make such statements regularly, but not everything we say is a proposition.

Example 1.3 *The following statements are propositions:*

1. *Aristotle was born in 384 BCE.*
2. *Boris Johnson is a fictional character.*
3. $2^3 = 8$.

The following expressions are not propositions:

4. *LOL!*
5. *27.*
6. *Happy birthday!*
7. *Would you like some coffee?*

These sentences are neither true nor false. A question can be answered with yes or no, but this is not quite the same as a proposition like "Angela Merkel likes coffee," which would be true or false.

Note that propositions do not need to be true. The *truth value* of a proposition is its truth or falsehood, denoted by T or F accordingly.

Example 1.4 *In Example 1.3 the truth values of the propositions are: 1. T, 2. F, 3. T.*

In this module we will also require that a proposition cannot assume any other truth value than true or false (there is no "maybe" or "probably" or "unknown"). This is called the *principle of the excluded third* or the *principle of the excluded middle*. Using this principle together with the requirement that a proposition is not both true and false (which is a kind of consistency condition), we can reformulate modus tollens as follows:

If p is true, then q is true.
q is false.
Therefore, p is false.

(Note that we simply replaced the words "not true" in the previous formulation of modus tollens by "false".)

Example 1.5 *Let us consider the following statement:*

Every even whole number greater than 2 is the sum of two prime numbers.

No one knows whether this is true or false. (This is Goldbach's conjecture, a famous open problem in mathematics.) Some authors would say this is not a proposition, because its truth value is "unknown", which is not allowed by the principle of the excluded middle. We will treat this as a proposition, though, claiming that it has to be either true or false and not both nor something else, even if we do not know (yet) what its truth value is.

In general it can be difficult to determine which statements are propositions and which ones are not. To avoid complications we do not allow statements that refer to themselves, or statements involving persons or objects whose identity is unclear. The second issue is illustrated by the following statement:

$x = 5$.

In order to assign a truth value to this statement we need to know more. What is x? In general the identity of x is unclear and we do not allow this as a proposition, but we will see later in Section 1.3 how to handle such expressions with other methods.

Remark 1.6 *As a remark on the side let us have a look at some other statements for which it is difficult to decide if they are propositions.*

1. *Tomorrow there will be rain.*

2. *The library is 100m away.*

3. *My water broke.*

4. *The tea bags are on my left.*

The truth values of these statements may depend on the time and location that the statement is made and on the person who makes the statement. Whether this is allowed, or to what extent, is an interesting question in the philosophy of language that will not concern us here, because the propositions used in mathematics or computing rarely have this character.
The statement

5. *Mariam is 25 years old*

is similar to $x = 5$ and depends on additional information: Mariam who? Because her identity is unclear, we will not allow this statement as a proposition. Note that this issue could have been raised earlier for Aristotle and for Boris Johnson, although their names are less likely to cause confusion. This issue illustrates that the decision whether something is a proposition or not can be difficult.

1.2.2 Compound Propositions

An important aspect of logic is that we can combine propositions to create new ones. The new proposition is called a *compound proposition*, whereas a proposition that is not built up out of other ones is called an *atomic proposition*. The different ways in which we can combine propositions are called *logical connectives* (or *logical operators*). We will list the most important ones, which mostly go by fancy names derived from latin.

Negation: The negation of a propostion p is the statement "not p". This statement is true when p is false and it is false when p is true. We write the negation in symbols as $\neg p$. (Some books prefer to write $\sim p$.) An example is:

p: Lois Lane is a journalist.
$\neg p$: Lois Lane is not a journalist.

The negation takes one proposition and turns it into anther one. We can also take two propositions and combine them into a single new one in several ways.

Conjunction: The conjuction of two propositions, p and q, is the statement "p and q". It is true when both p and q are true and it is false in all other cases. We write the conjuction in symbols as $p \wedge q$. (Some books prefer to write p **and** q.) An example is:

p: Lois Lane likes superman.
q: Lois Lane dislikes Clark Kent.
$p \wedge q$: Lois Lane likes superman and Lois Lane dislikes Clark Kent.

Disjunction: The disjuction of two propositions, p and q, is the statement "p or q". It is true when at least one of p and q is true (possibly both) and it is false when p and q are both false. We write the disjunction in symbols as $p \vee q$. (Some books prefer to write p **or** q.) An example is:

p: Lex Luthor had a bad childhood.
q: Lex Luthor is simply ruthless.
$p \vee q$: Lex Luthor had a bad childhood, or Lex Luthor is simply ruthless.

Implication: The implication between propositions p and q is the statement "If p then q", or sometimes "p is a sufficient condition for q" or "q is a necessary condition for p". The implication is true when p and q are both true and also when p is false (regardless of the truth value of q). It is only false when p is true and q is false. We write the implication in symbols as $p \Rightarrow q$. An example is:

p: Clark Kent is superman.
q: Clark Kent can fly.
$p \Rightarrow q$: If Clark Kent is superman, then Clark Kent can fly.

Bi-implication: The bi-implication (or equivalence) of two propositions p and q is the statement "p if and only if q", or "p is a necessary and sufficient condition for q". It is true when p and q have the same truth value, i.e. when both are true or when both are false. It is false when p and q have different truth values. We write the bi-implication in symbols as $p \Leftrightarrow q$. An example is:

p: Clark Kent is a happy man.
q: Lois Lane is a happy woman.
$p \Leftrightarrow q$: Clark Kent is a happy man if and only if Lois Lane is a happy woman.

Remark 1.7 *Most of the logical connectives are very similar to words that we use in everyday language, but there are some subtleties. This is because our natural language can be ambiguous and propositional logic needs to remove these ambiguities. An important example of this is the word "or", which has two distinct meanings. One of them corresponds to the logical operator $\vee$, which is sometimes called an inclusive disjunction, because $p \vee q$ is also true when both p and q are true. By contrast, the word "or" can also denote an exclusive disjunction, a logical connective that sometimes arises in computing and that is often written as p* **xor** *q. This compound proposition is true when p is true or q is true, but not both. The exclusive disjunction is often used when you are asked to choose between two options, e.g. "Do you want to go left or right"?*

We can combine the logical connectives to construct more complicated propositions. Here we use parentheses (i.e. brackets) to clarify the order in which the connectives need to be applied (similar to mathematical formulae in arithmetic and algebra). E.g., for propositions p, q and r the compound proposition $(p \vee q) \wedge r$ does not mean the same thing as $p \vee (q \wedge r)$. It is customary to omit the brackets for $\neg$ when it acts on the atomic proposition immediately following it, so $\neg p \vee q$ means $(\neg p) \vee q$ and not $\neg(p \vee q)$.

A *logical expression* is a formula that is built up of statement variables like p, q, etc., combined with the connectives $\neg$, $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$ as well as the parentheses (and). (We will only consider logical expressions that make sense, unlike e.g. $\vee pqr)) \wedge \neg(.)$ When we substitute specific propositions for p, q, etc., then the logical expression will turn into a proposition. (This is analogous to a mathematical expression like $x \cdot (x - 1) + 7$, which is not a number itself, but when we substitute a number for x, say $x = 5$, then the expression becomes a number, namely $5 \cdot (5 - 1) + 7 = 27$.)

Example 1.8 *The logical structure of modus ponens and modus tollens can be written entirely in terms of logical symbols if we write $\therefore$ to mean "therefore". We then have*

$$\begin{array}{l} p \Rightarrow q \\ p \\ \therefore q \end{array}$$

for modus ponens and

$$\begin{array}{l} p \Rightarrow q \\ \neg q \\ \therefore \neg p \end{array}$$

for modus tollens.

1.2.3 Truth Tables and Equivalences

The truth value of a compound proposition depends on the truth values of its constituents. A very useful way to see this dependence is by constructing a *truth table*.

Example 1.9 *The following truth table shows the dependence of the truth values of $p \wedge q$ on those of p and q:*

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

In general, reading from left to right, a truth table starts by listing the relevant atomic propositions and all combinations of their truth values. It then continues to list the truth values of compound statements, using one column for each logical connective. Here are the truth tables for all the logical connectives we have seen:

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

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

p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

p	q	$p \Leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

p	$\neg p$
T	F
F	T

Let us now look at a more complex compound proposition of the form $p \vee (\neg q)$. This logical expression has two connectives, so we need an extra column for each connective:

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

Some compound propositions can only be false, regardless of the truth values of the atomic propositions occuring in it. E.g. $p \wedge \neg p$ is false for any proposition p, because a proposition cannot be true and not true. A logical expression whose truth table only contains the value F is called a *contradiction*. A specific proposition which can only be false due to its logical structure is called a *contradictory statement*. Similarly, a logical expression whose truth table only contains the value T is called a *tautology* and a specific proposition which can only be true due to its logical structure is called a *tautological statement*. An example of a tautology is $p \vee \neg p$, which expresses the principle of the excluded third. It will be convenient to introduce the symbol $\top$ for a proposition which can only be true and the symbol $\bot$ for a proposition which can only be false.

We call two logical expressions p and q *equivalent* when $p \Leftrightarrow q$ is a tautological statement. Truth tables are a great tool to find out whether a logical expression is a tautology, or whether two compound propositions are equivalent. Note that a tautology must have the truth value T in each row and two propositions are equivalent precisely when they have the same truth value on each row.

Example 1.10 *Let us compare the logical expressions $p \Rightarrow q$ and $(\neg q) \Rightarrow (\neg p)$:*

p	q	$\neg q$	$\neg p$	$p \Rightarrow q$	$(\neg q) \Rightarrow (\neg p)$
T	T	F	F	T	T
T	F	T	F	F	F
F	T	F	T	T	T
F	F	T	T	T	T

Because the last two columns have the same truth values in each row, the logical expressions $p \Rightarrow q$ and $(\neg q) \Rightarrow (\neg p)$ are equivalent. Alternatively, one could add an extra column for the expression $(p \Rightarrow q) \Leftrightarrow ((\neg q) \Rightarrow (\neg p))$ and one finds that this has the truth value T in every row, so it is a tautology.

The equivalence in Example 1.10 is called the *contraposition*, or contrapositive.

Warning: The implication $p \Rightarrow q$ is not equivalent to the implication $q \Rightarrow p$, which is called its *converse*.

Equivalences of logical expressions can often be used as rules to rewrite them. Such rewriting is often useful to simplify expressions, or, in computing, to make programs run more efficiently and to reduce the chance of error. In particular, the equivalences in Table 1 show that the logical connectives of Section 1.2.2 are not independent and that we can completely eliminate the logical connectives $\Leftrightarrow$, $\Rightarrow$ and $\wedge$ and rewrite them in terms of $\neg$ and $\vee$.

logical expression	equivalent expression	name of rule
$p \Leftrightarrow q$	$(p \Rightarrow q) \wedge (q \Rightarrow p)$	equivalence law
$p \Rightarrow q$	$(\neg p) \vee q$	implication law
$\neg(p \wedge q)$	$(\neg p) \vee (\neg q)$	De Morgan's law
$\neg(p \vee q)$	$(\neg p) \wedge (\neg q)$	De Morgan's law
$\neg(\neg p)$	p	double negative law
$p \wedge (q \vee r)$	$(p \wedge q) \vee (p \wedge r)$	distributive law of $\wedge$ over $\vee$
$p \vee (q \wedge r)$	$(p \vee q) \wedge (p \vee r)$	distributive law of $\vee$ over $\wedge$

Table 1: Equivalences of logical expressions. All these equivalences can be established using truth tables as in Example 1.10.

Example 1.11 *The logical expression $p \wedge q$ is equivalent to $\neg((\neg p) \vee (\neg q))$. This can be shown using truth tables as before, or it can be derived from the table of equivalences above as follows. De Morgan's law says that $(\neg p) \vee (\neg q)$ is equivalent to $\neg(p \wedge q)$. Applying a negation to both expressions preserves this equivalence. (This may sound obvious, but in principle it should be checked using truth tables.) Applying the negation shows that $\neg((\neg p) \vee (\neg q))$ is equivalent to $\neg\neg(p \wedge q)$, which in turn in equivalent to $p \wedge q$ by the double negation law.*

1.3 Predicate Logic

So far we have treated propositions as the smallest building blocks for logic, but propositions often have some internal structure that can also be very useful. In predicate logic we consider a particularly useful case of such internal structure, namely predicates, which can describe properties of things or relations between them. Such statements occur e.g. in computer programming, where instructions can contain conditions like "If $x < 5$" with a variable x that can take different values during the course of the programme.

1.3.1 Predicates

Consider the following propositions:

p: 2 is an even number.
q: 3 is an even number.
r: 4 is an even number.
s: 5 is an even number.

These propositions are strikingly similar and it is easy to construct many more of them. Indeed, given any natural number n we could make a corresponding proposition "n is a prime number", where we substitute the given number for n. E.g., if $n = 37$ we get "37 is an even number". (We would soon run out of letters to use as labels for these propositions, but this can be fixed by choosing the labels more cleverly, e.g. p_2, p_3, p_4, p_5, etc.)

To clarify the common structure of these propositions we can use predicates. A *predicate* is a property of a variable, or a relation between several variables. We write a predicate as $P(x)$, or $Q(x, y)$, where x and y are variables and the predicate symbol P denotes the property or relation. (We can think of variables like x as an individual thing or person whose identity is not known.)

Example 1.12 *The following sentences with variables are predicates:*

$P(x)$*: x is an even number.*
$Q(x, y)$*: x likes to play y.*
$R(x)$*: $x < 37$.*

The following sentence also contains a variable, but is not a predicate:

Let us consider the number x.

If we substitute a number for x, say 5, we get the sentence "Let us consider the number 5." This is not a proposition: it cannot be true nor false.

For $P(x)$ the variable x should be a (natural) number, $x = 1$, $x = 2$, $x = 3$, etc., but for $Q(x, y)$ the variables x and y should apparently be humans (or other living creatures who can be in love). This shows that the values that can be substituted for the variables can depend on the predicate. The collection of values over which x can range is called the *domain of discourse* of the predicate variable. It is often useful to identify this domain by giving it a name, say D. We can then indicate that x ranges over this domain by writing $x \in D$.

Remark 1.13 *It is sometimes helpful to think of a proposition as a kind of predicate which takes in zero variables. This can be particularly useful when discussing predicates with a varying number of variables in a systematic way. In this module, however, there is no need to think about propositions this way.*

We can use the logical operators of Section 1.2.2 also for predicates in an obvious way. For predicates $P(x)$ and $Q(x)$, we can treat $\neg P(x)$, $P(x) \wedge Q(x)$, $P(x) \vee Q(x)$, $P(x) \Rightarrow Q(x)$ and $P(x) \Leftrightarrow Q(x)$ as (compound) predicates, where we can substitute a value for x in the predicates $P(x)$ and $Q(x)$ to obtain propositions and then combine these propositions as before.

Example 1.14 *Consider the predicates*

$P(x)$*: x is an animal.*
$Q(x)$*: x is red.*

Then we can use logical operators to obtain the following predicates:

$\neg P(x)$*: x is not an animal.*
$P(x) \wedge Q(x)$*: x is an animal and x is red.*
$P(x) \vee Q(x)$*: x is an animal or x is red.*
$P(x) \Rightarrow Q(x)$*: If x is an animal, then x is red.*
$P(x) \Leftrightarrow Q(x)$*: x is an animal if, and only if, x is red.*

1.3.2 Quantifiers

The advantage of using predicates becomes much clearer when we consider propositions like the following two:

All humans are mortal.
There is a black swan.

Instead of treating these as atomic propositions, we can also inspect their internal structure using predicates as follows:

$P(x)$: x is mortal.
H: all humans.
$Q(x)$: x is black.
S: all swans.
For all $x \in H$ $P(x)$.
There is an $x \in S$ such that $Q(x)$.

Here $P(x)$ and $Q(x)$ are predicates and H and S are domains of discourse. The last two lines represent the two propositions above written in predicate logic. Note that we did not substitute a particular value for the predicate variable x in order to turn the predicates into a proposition. Instead we used the words "For all x" and "There is an x such that". A statement starting with the words "For all x" is called a *universal* statement and a a statement starting with the words "There is an x such that" is called an *existential* statement.

The *universal quantifier* is the symbol $\forall$, which represents the words "For all". The *existential quantifier* is the symbol $\exists$, which represents the words "There exists... such that". Note that $\forall$ looks like a capital letter A turned upside down, which should remind you of the word "all", which starts with an a. Similarly, $\exists$ looks like a mirror image of the capital letter E (with left and right swapped), which should remind you of the word "exists", which starts with an e. The propositions above can then be written as a formula as follows:

$\forall x \in H$ $P(x)$.
$\exists x \in S$ $Q(x)$.

Here, $P(x)$ is a predicate with variable x, but the variable is said to be *bound* by the universal quantifier $\forall x \in H$, so the formula $\forall x \in H$ $P(x)$ is a proposition and not a predicate: it can no longer accept a variable as input. Similarly, $Q(x)$ is a predicate, but the variable x is bound by the existential quantifier $\exists x$, so the formula $\exists x \in S$ $Q(x)$ is a proposition and not a predicate: it can no longer accept a variable as input. A variable which is not bound by a quantifier is called a *free variable*.

The truth values of quantified statements are somewhat obvious. The proposition $\forall x \in D$ $P(x)$ is true exactly when $P(x)$ is true for all $x \in D$ and $\exists x \in D$ $P(x)$ is true exactly when there is at least one $x \in D$ for which $P(x)$ is true.

Remark 1.15 *If the domain of discourse D of a predicate $P(x)$ contains finitely many elements, say 37, then we can label them by, say, $a_1, a_2, \ldots, a_{37}$. We can then consider the 37 propositions $P(a_1)$, $P(a_2)$, $P(a_3)$, ..., $P(a_{37})$, each of which is either true or false. In this setting the proposition*

$\forall x \in D$ $P(x)$

is equivalent to $P(a_1) \wedge P(a_2) \wedge P(a_3) \wedge \ldots \wedge P(a_{37})$ and the proposition

$\exists x \in D$ $P(x)$

is equivalent to $P(a_1) \vee P(a_2) \vee P(a_3) \vee \ldots \vee P(a_{37})$. *This means that in this case the new concepts of predicate logic can be reduced to proopositional logic. However, this method breaks down when the domain of discourse* D *has infinitely many elements, which happens quite often in mathematics.*

Because predicate logic knows more about the internal structure of some propositions, it can also explain the validity of a wider range of logical arguments than propositional logic. As an example we consider the following logical argument:

All humans are mortal.
Socrates is human.
Therefore, Socrates is mortal.

This argument is valid, but this cannot be explained by propositional logic alone. Although the structure looks very close to modus ponens, it is slightly different:

p: All humans are mortal.
q: Socrates is human.
r: Socrates is mortal.
p
q
$\therefore r$.

To have modus ponens we would need to replace the first line by $q \Rightarrow r$, i.e. by "If Socrates is human, then Socrates is mortal". As it stands, the structure of this argument, given in the last three lines, is not valid in propositional logic. If the argument is valid, it must be due to the internal structure of the propositions p, q and r.

Note that the original argument relies on a particular property of Socrates, namely that he is human. To explain the validity of this argument we need to resort to predicate logic, which is about properties. This goes as follows:

$P(x)$: x is mortal.
H: all humans.
$\forall x \in H \ P(x)$.
Socrates $\in H$.
$\therefore P$(Socrates).

Here Socrates is not a variable, but a (constant) individual in H. Note that we have written the previous propositions p and r as $p : \forall x \in H \ P(x)$ and $r : P$(Socrates), respectively. The proposition q is replaced by Socrates $\in H$, which identifies Socrates as an individual in the domain of discourse of the predicate $P(x)$. (We may think of this as a proposition: "Socrates is in the domain of discourse H".) Next we will need to explain that the structure of this argument is of a special form, which is valid.

One of the most important kinds of a valid argument in predicate logic is the *universal instantiation*. (Its central role in predicate logic is a bit similar to the role of modus ponens in propositional logic). Its structure is as follows:

$\forall x \in D \ P(x)$
$a \in D$
$\therefore P(a)$

Here a is a constant individual in D, not a variable (similar to Socrates in H). This argument simply allows us to deduce the property P for a single individual a, if we know that it is true for the entire class of individuals to which a belongs. The argument about Socrates' mortality is exactly of the universal instantiation form.

It is also possible to have a universal statement as the conclusion of a valid logical argument. To deduce $\forall x \in D\ P(x)$ we first pick an arbitrary $x \in D$. This means we know nothing about x, except that it is in D. If it follows that $P(x)$ for this particular, but arbitrary, x, then we may conclude that $\forall x \in D\ P(x)$. This (rather tricky) type of argument is called *universal generalization* and it is a key argument in mathematics, where we often try to prove statements that are as general as possible. If we know a universal statement for a large domain of discourse, then we immediately know a lot of particular cases by universal instantiation.

Another valid argument structure in predicate logic which is not contained in propositional logic is the *existential introduction*. Its structure is:

$a \in D$
$P(a)$
$\therefore \exists x \in D\ P(x)$

Again a is a constant individual in D, not a variable. Simply put, if we know that the individual a is in the domain of discourse D and has the property P, then there is an individual in D with the property P.

1.3.3 Equivalences in Predicate Logic

There are several rules that govern the use of quantifiers. The most important ones are the logical equivalences in Table 2. Here D is the domain of discourse of the predicate $P(x)$. To obtain the negation of a statement involving a quantifier we see that we need to change the quantifier and negate the predicate. (Nothing happens to the domain of discourse.)

logical expression	equivalent expression	name of rule
$\neg\forall x \in D\ P(x)$	$\exists x \in D\ \neg P(x)$	negation of a universal statement
$\neg\exists x \in D\ P(x)$	$\forall x \in D\ \neg P(x)$	negation of an existential statement

Table 2: Some logical equivalences in predicate logic.

Note that we may treat the expression $x \in D$ as a proposition, which is true when x is in the domain of discourse D and false when it is not. Using this we can rewrite quantified statements in the following way, which is sometimes useful. Instead of $\forall x \in D\ P(x)$ we may write $\forall x\ (x \in D) \Rightarrow P(x)$ and instead of $\exists x \in D\ P(x)$ we may write $\exists x\ (x \in D) \wedge P(x)$.

An important issue arises when predicates involve two or more variables.

Example 1.16 *Let us consider the predicate*

$P(x, y)$*:* $x < y$

where the domain of discourse consists of all natural numbers, i.e. $x \in \mathbb{N}$ *and* $y \in \mathbb{N}$*. To turn this predicate into a proposition we can either substitute individual values for* x *and* y*, e.g.* $7 < 3$ *(which is false), of we can use quantifiers. Let us now consider the propositions:*

q*:* $\exists y \in \mathbb{N}\ \forall x \in \mathbb{N}\ P(x, y)$
r*:* $\forall x \in \mathbb{N}\ \exists y \in \mathbb{N}\ P(x, y)$

Writing these logical expressions out in words we notice that there is indeed a difference between them:

q: There is a natural number y such that $x < y$ for all natural numbers x.
r: For all natural numbers x there is a natural number y such that $x < y$.

The second statement is clearly true: given any x we may choose y to be larger, e.g. $y = x + 1$. The first statement, however, is false: there is no number y which is larger than all natural numbers x. (Note that "infinity" is not a natural number.)

Apparently the ordering of the quantifiers $\forall$ and $\exists$ matters. In general, in a statement of the form

$$\forall x \in D\ \exists y \in D'\ P(x, y)$$

the individual y that should make the proposition $P(x, y)$ true may depend on x, whereas in a statement of the form

$$\exists y \in D'\ \forall x \in D\ P(x, y)$$

the same y should make the proposition $P(x, y)$ true for all x.

When negating an expression with two quantifiers we also need to keep the ordering in mind. E.g., the following logical expressions are equivalent:

$$\neg(\forall x \in D\ \exists y \in D'\ P(x, y))$$
$$\exists x \in D\ \neg(\exists y \in D'\ P(x, y))$$
$$\exists x \in D\ \forall y \in D'\ \neg P(x, y)$$

A similar ordering issue does not arise if we use the same quantifier for both variables, so $\forall x\ \forall y\ P(x, y)$ is equivalent to $\forall y\ \forall x\ P(x, y)$ and similarly for two existential quantifiers.

1.4 Methods of Proof in Mathematics

Mathematical proofs are logically valid arguments that start with some assumptions about mathematical objects and end with a conclusion. Such proofs are also used to verify the correctness of computer algorithms. We will present several basic types of proof and the important method of mathematical induction. To illustrate these methods we will consider various statements about numbers and we introduce recursively defined sequences of numbers.

Although logic and mathematics are very precise and systematic, there is no fool proof recipe that tells you how to find a proof for a given statement. To find a proof often takes multiple attempts and, like a creative art, you get better at it with experience.

1.4.1 Basic proof methods

Suppose that p and q are propositions and we want to show that $p \Rightarrow q$ is true. Within propositional logic we can identify three different strategies to do this:

direct proof: We assume p and try to derive q to show directly that $p \Rightarrow q$ is true.

direct proof using contraposition: We assume $\neg q$ and ty to derive $\neg p$ to show that $(\neg q) \Rightarrow (\neg p)$ is true. This is equivalent to $p \Rightarrow q$ being true.

proof by contradiction: (Also known as *reductio ad absurdum.*) We assume p and $\neg q$ and derive a contradiction. This shows that $p \wedge (\neg q)$ is false. (More formally it shows that $(p \wedge (\neg q)) \Rightarrow \bot$ is true. Since $\bot$ is false this can only happen when $p \wedge (\neg q)$ is false.) By negation, $\neg(p \wedge (\neg q))$ is true. This is equivalent to $(\neg p) \vee q$ and hence to $p \Rightarrow q$ being true.

Because more general results contain more specific cases we are often interested in proving a statement of the form $\forall x \in D\ P(x) \Rightarrow Q(x)$. To do this we can generalise the three strategies above. When D contains a finite number of individuals, we can prove $P(x) \Rightarrow Q(x)$ for each individual separately using one of the three strategies. This is called the *method of exhaustion*. For more general domains of discourse we can combine the three strategies above with universal generalization. This is illustrated in the following examples.

Example 1.17 *Recall that the natural numbers are* $1, 2, 3, \ldots$. *A natural number is even if and only if it equals 2 times some natural number. A natural number is odd if and only if it equals an even natural number minus 1. We consider the following three statements:*

1. *The sum of two odd natural numbers is even.*

2. *Every natural number is either even or odd.*

3. *If a natural number has an odd square, then it is odd.*

To clarify the proofs of these statements it helps to translate them into a logical formula. Consider the predicates

$E(x)$: *x is an even number*
$O(x)$: *x is an odd number*

then the statements can be written as

1. $\forall m \in \mathbb{N}\ \forall n \in \mathbb{N}\ (O(m) \wedge O(n)) \Rightarrow E(m+n)$

2. $\forall m \in \mathbb{N}\ \exists n \in \mathbb{N}\ n > m$

3. $\forall n \in \mathbb{N}\ E(n) \vee O(n)$

4. $\forall n \in \mathbb{N}\ O(n^2) \Rightarrow O(n)$

Let us now consider proofs of these statements:

1. ***direct proof:*** *Let $m \in \mathbb{N}$ and $n \in \mathbb{N}$ be two arbitrary numbers. Assume that m and n are odd, so $m = 2k-1$ and $n = 2l-1$ for some $k \in \mathbb{N}$ and some $l \in \mathbb{N}$. Then $m+n = (2k-1)+(2l-1) = 2k+2l-2 = 2(k+l-1)$. Since $k+l-1$ is also a natural number, this shows that $m+n$ is even. Since m and n were arbitrary, the conclusion holds for all odd natural numbers, as was to be shown.*

2. ***proof by contradiction:*** *Let $n \in \mathbb{N}$ be an arbitrary number. Suppose that n is neither even nor odd. Then $n > 1$, because 1 is odd. Now consider $n-1$, which is again a natural number. If $n-1$ were even, then $n-1 = 2k$ for some natural number k and hence $n = 2k+1 = 2(k+1)-1$, so n would be odd, which contradicts our assumption on n. If $n-1$ were odd, then n would be even, which again contradicts our assumptions on n. Therefore, $n-1$ can be neither even nor odd. Repeating this argument we find that $n-2$ can be neither even nor odd, $n-3$ can be neither even nor odd, etc., until we find that 1 can be neither even nor odd. However, this gives a contradiction, because 1 is odd. Because the assumption that n is neither even nor odd leads to a contradiction, it must be false, i.e. n must be either even or odd. Since n was arbtirary, the conclusion holds for all natural numbers, as was to be shown.*

3. ***direct proof using contraposition:*** *Let $n \in \mathbb{N}$ be an arbitrary number. Suppose that n is not odd. Then n is even and $n = 2k$ for some natural number $k \in \mathbb{N}$. Therefore, $n^2 = (2k)^2 = 4k^2 = 2(2k^2)$. Since $2k^2$ is also a natural number we see that n^2 is an even number and in particular it is not an odd number. Since n was arbtirary, the conclusion holds for all natural numbers, as was to be shown.*

Instead of the final words "as was to be shown" mathematicians often write Q.E.D., which is an abbreviation of the Latin words "quod erat demonstrandum", meaning "which was to be shown".

All statements in Example 1.17 are universal statements. Sometimes it is useful to prove that a universal statement is false. Using the equivalence of $\neg(\forall x \in D\ P(x))$ and $\exists x \in D\ \neg P(x)$ this can be done by proving an existential statement.

A general strategy to prove an existential statement $\exists x \in D\ Q(x)$ is a *constructive proof*, where one tries to find an explicit example of a specific individual $a \in D$ for which $Q(a)$ is true. To find or construct such an example a often requires much creativity and ingenuity. When we want to disprove a universal statement, then $Q(x) = \neg P(x)$ and a must be an example for which the negative statement $\neg P(a)$ is true, so $P(a)$ is false. For this reason we call a a *counter-example* in that case.

Finally it is sometimes useful to prove that a solution to a mathematical problem is unique. This can be cast into a logical form as follows. Consider a particular mathematical problem and let $P(x)$ be the predicate "x is a solution". What we mean by uniqueness is that there can be at most one solution (there might be none). This is captured by the following logicla expression:

$$\forall x \in D\ \forall y \in D\ (P(x) \wedge P(y)) \Rightarrow x = y$$

1.4.2 Proof by mathematical induction

There is a special strategy for proving statements of the form $\forall n \in \mathbb{N}\ P(n)$ where the domain of discourse is the set of natural numbers, $\mathbb{N} = \{1, 2, 3, \ldots\}$:

Mathematical induction: If $P(x)$ is a predicate with domain of discourse $\mathbb{N}$ and if

1. $P(1)$ is true (base case)

2. $\forall n \in \mathbb{N}\ P(n) \Rightarrow P(n+1)$ is true (induction step)

then $\forall n \in \mathbb{N}\ P(n)$ is true.

The principle of *mathematical induction* works like dominoes:

1. $P(1)$ is true by assumption.

2. From the induction step we get $P(1) \Rightarrow P(2)$ (by universal instantiation). From this, $P(1)$ and modus ponens we find that $P(2)$ is true.

3. From the induction step we get $P(2) \Rightarrow P(3)$ (by universal instantiation). From this, $P(2)$ and modus ponens we find that $P(3)$ is true.

4. etc.

For any $n \in \mathbb{N}$ we find that $P(n)$ is true by repeating the argument above a finite number of times.

Example 1.18 *For every $n \in \mathbb{N}$ it is true that $1 + 2 + \ldots + n = \frac{1}{2}n(n+1)$.*
Proof by induction: *For $n \in \mathbb{N}$ let $P(n)$ be the statement that $1 + 2 + \ldots + n = \frac{1}{2}n(n+1)$. We first prove the base case $P(1)$. When $n = 1$ the left hand side is 1 and the right-hand side is $\frac{1}{2} \cdot 1 \cdot (1+1) = \frac{1}{2} \cdot 2 = 1$. Since $1 = 1$ we have equality, i.e. $P(1)$ is true.*
Next we prove the induction step. Let $n \in \mathbb{N}$ be arbitrary. We assume $P(n)$ (the induction hypothesis*) and want to prove $P(n+1)$. For this we compute*

$$\begin{aligned} 1 + 2 + \ldots + n + (n+1) &= \frac{1}{2}n(n+1) + (n+1) \\ &= \frac{1}{2}(n+1)n + \frac{1}{2}(n+1) \cdot 2 \\ &= \frac{1}{2}(n+1)(n+2)\,, \end{aligned}$$

where we used the induction hypothesis in the first equality. Comparing the first and last expressions we see that $P(n+1)$ holds (since $(n+1)+1 = n+2$). Since $n \in \mathbb{N}$ was arbitrary we conclude that $\forall n \in \mathbb{N}\ P(n) \Rightarrow P(n+1)$. From the principle of mathematical induction it then follows that for all $n \in \mathbb{N}$ $P(n)$ is true, which is what was to be shown.

The numbers $1 + 2 + \ldots + n$ in Example 1.18 can be described in a more precise way, without "…", if we define them *recursively*, i.e. if each next number in the sequence is defined in terms of the previous one(s) by a formula, which is called a *recurrence relation*. This works as follows. First we want to give a name to every number in the sequence. We cannot use x, y, z, because we soon run out of letters in the alphabet. For this reason we call the numbers x_1, x_2, x_3, etc., where the index k on a number x_k indicates its place in the sequence. (x_1 is the first number, x_2 is the second number, etc.). We can now define $x_1 := 1$ and, for every $n \in \mathbb{N}$

$$x_{n+1} := x_n + (n+1)\,.$$

By substituting increasing values of n into this formula we can compute

$$\begin{aligned} x_1 &= 1 \\ x_2 &= x_1 + 2 = 1 + 2 \\ x_3 &= x_2 + 3 = 1 + 2 + 3 \end{aligned}$$

etc. This reproduces the desired expression $x_n = 1 + 2 + \ldots + n$.

Warning: Note that x_{k+1}, with $+1$ in the subscript, is generally not the same as $x_k + 1$! Whereas x_{k+1} is the next number in the sequence after x_k, $x_k + 1 = 1 + x_k$ is the number x_k plus one.

When a sequence of numbers is defined recursively, it is often useful to find out if it can also be described in a non-recursive way, using a direct formula in terms of k which does not depend on other numbers in the sequence. Example 1.18 finds such a direct formula using a proof by induction. Here is another example.

Example 1.19 *Let a be a real number with $a \neq 1$. Define the sequence of numbers x_k recursively by*

$$\begin{aligned} x_1 &:= 1 \\ x_{k+1} &:= x_k + a^k\,. \end{aligned}$$

Then $x_k = \frac{1-a^k}{1-a}$ for all $k \in \mathbb{N}$. (Note that for $a = 1$ the quotient on the right-hand side is not defined, because we cannot divide by zero.)

Proof: *We give a proof by induction. For $k \in \mathbb{N}$ let $P(k)$ be the statement $x_k = \frac{1-a^k}{1-a}$.*
Base case: $x_1 = 1$ by definition and $\frac{1-a^1}{1-a} = 1$, because $a^1 = a$. Hence, $x_1 = \frac{1-a^1}{1-a}$ and $P(1)$ is true.
Induction step: Let $k \in \mathbb{N}$ be arbitrary and assume $P(k)$. Using the recursive definition of the sequence we compute

$$\begin{aligned} x_{k+1} &= x_k + a^k \\ &= \frac{1-a^k}{1-a} + a^k \\ &= \frac{1-a^k}{1-a} + \frac{1-a}{1-a}a^k \\ &= \frac{1-a^k}{1-a} + \frac{(1-a)a^k}{1-a} \\ &= \frac{1-a^k}{1-a} + \frac{a^k - a^{k+1}}{1-a} \\ &= \frac{1-a^k + a^k - a^{k+1}}{1-a} \\ &= \frac{1-a^{k+1}}{1-a} \end{aligned}$$

so $P(k+1)$ is true.
It now follows from the principle of mathematical induction that for all $k \in \mathbb{N}$ $x_k = \frac{1-a^k}{1-a}$.