6 Tools from logic [7]

The fundamental goal of mathematics is to determine which mathematical assertions are true and which are false. If we have an assertion that we believe to be true, our aim is to give a *proof*, which is a convincing argument that it is true As we'll see, the notion of proof in mathematics is more demanding that in most other fields. While "proofs" in everyday life, or even in science, may leave room for doubt or controversy, a correct mathematical proof should leave no room for doubt.

Mathematical proofs are built by a series of *deductions*. In doing a mathematical proof, we always have a mathematical scenario, that is, some active objects and active assumptions about those objects. Deduction is the process of using the active assumptions together with universal principles to draw conclusions that most hold whenever the active assumptions hold.

The field of *mathematical logic* treats proofs themselves as mathematical ojbects. This turns out to be quite a difficult subject, with many subtle and complicated issues. Fortunately, the majority of problems that mathematicians study do not require deep understanding of the subtleties of mathematical logic, but they do require knowledge of some of the elementary tools of mathematical logic. Section 4 covered some of this basic knowledge: definite and indefinite assertions, and universal and existential quantification. In this section we cover some additional essentials in mathematical logic.

- Modeling complex assertions by logical expressions. By representing basic assertions by single letters, we can represent the structure of a complex assertion by an expression involving variables and logical connectives and quantifiers.
- Decomposing a complex assertion. Given a complex assertion, we need to be able to decompose it into its simpler parts. In particular, every complex assertion has something we call its *top-level structure* and you need to be able to identify this.
- Understanding the difference between logical deductions and mathematical deductions
- Making logical deductions: how to identify that one assertion can be correctly deduced from others.
- Logical equivalence. Two assertions are logically equivalent if each can be logically deduced from teh other.

Logical expressions. In elementary algebra, we work with *algebraic expressions* such as

$$5 \times a \times (b + c \times ((a - b) - 4) + 7 \times d.$$

These expressions are built by starting with variables, each representing a real number, and combining them with each other, or with constants by using addition, multiplication and subtraction.

[7]Version 9-26-2016(7PM).

In logic, one deals with *logical expressions*. In such expressions. the variables, called *propositional variables* represent assertions, which are combined or modified using $\neg$, $\wedge$, $\vee$, $\implies$ and $\iff$, $\forall$ and $\exists$.

Here's an example of a logical expression:

$$\exists x, A(x) \wedge \forall z (B(z) \implies C(z))$$

In the logical expression above, A, B and C represent assertions. We write $A(x)$ instead of A to emphasize that x appears as a free variable in A, and similarly z appears as a free variable in B and C

Remark 6.1. **Use of parentheses**. As with algebraic expressions, there can be ambiguity in the meaning of expressions depending on the order that we apply the logical connectives. Thus $A \implies B \wedge C$ is ambiguous, and we use parentheses to make clear whether we mean $A \implies (B \wedge C)$ or $(A \implies B) \wedge C$.

Modeling assertions by logical expressions Logical expressions allow us to study the way assertions are put together from certain starting assertions, without worrying what the starting assertions are. In the above example the expression is built from starting assertions A, B and C.

Example 6.1. Suppose we are studying the assertion "For every integer n, if n is prime and $n-1$ is divisible by 4, then there are integers a and b such that $n = a^2 + b^2$." We can represent this by logical expressions in various ways, depending on how we assign our variables:

- We could simply assign a single variable A to the entire assertion, and our logical representation is simply A.
- We could use $B(n)$ to represent the indefinite assertion "if n is prime and $n-1$ is divisible by 4 then there are integers a and b such that $n = a^2 + b^2$" and then our entire sentence is represented by "$\forall n B(n)$".
- We could let $C(n)$ be the indefinite assertion "n is prime", let $D(n)$ be the indefinite assertion "$n - 1$ is divisible by 4", and $E(n)$ be the indefinite assertion (with dummy variables a, b) "There are integers a and b such that $n = a^2 + b^2$". In this case our sentence will have the representation "$\forall n((C(n) \wedge D(n)) \implies E(n))$".
- We can let $F(n, a, b)$ denote the indefinite assertion with free variables a, b, n given by "$n = a^2 + b^2$". We can then rewrite our sentence as $\forall n((C(n) \wedge D(n)) \implies \exists a \exists b F(n, a, b)$.

All of these logical expressions represent the original assertion; the later ones provide more detail on the structure of the assertion by breaking it down further.

An assertion is called *atomic* if it does not use any quantifier "there exists" , "for all" or "there exists a unique", and uses no logical connectives "and", "or", "implies (if-then)" or "if and only if". In the previous example all of the assertions $C(n)$, $D(n)$ and $F(n, a, b)$ are atomic expresions. Here are other examples of atomic assertions: "$x^2 \neq 7$" or "$f(x) \leq f(z)$, and "$S \subseteq T$" Note that if A is atomic, then we also consider its negation $\neg(A)$ to be atomic.

Remark 6.2. The assertion "$A \subseteq B$" is atomic, but if we use the definition of "$\subseteq$", we can rewrite the sentence as: "for all $x \in A$, we have $x \in B$". This is not an atomic assertion since it has a universal quantifier. Thus it's possible that two sentences can have the same meaning where one is atomic and the other iis not.

An assertion that is not atomic is called a *compound assertion.* Every compound assertion is of one of the following forms:

decomposable An assertion of the form "A and B", "A or B", "A implies B" or "A if and only if B", where A and B are assertions, is said to be *decomposable*, because it can be decomposed into two assertiions. Note that A and B might themselves be compound assertions.

quantified An assertion of the form "for all x, $A(x)$" or "there exists x such that $A(x)$" is a *quantified assertion.* Here $A(x)$ might itself be a compound assertion.

negative compound assertion An assertion of the form "it is not true that A" where A is a compound assertion. is a negative compound assertion.

Representation of compound assertions using a logical expression will be very useful to us when we start doing proofs. The following three concepts will be especially important.

- The *top-level* structure of a compound assertion
- *Logical equivalence* of two compound assertions
- One compound assertion is a *logical consequence* of another.

Top-level structure of a compound assertion Every compound assertion can be written in exactly one of the following 7 forms, by specifying the assertions A and/or B appropriately. For a given compound assertion, the form that fits it is called the *top-level structure* of the assertion.

Form of assertion	Top-level structure
$A \wedge B$	**AND**
$A \vee B$	**OR**
$A \implies B$	**IMPLIES** or **IF-THEN**
$A \iff B$	**IF AND ONLY IF**
$\neg A$	**NEGATION**
$\exists x, A(x)$	**THERE EXISTS**
$\forall x, A(x)$	**FOR ALL**
$\exists! x, A(x)$	**UNIQUE EXISTENCE**'

Example 6.2. The sentence "For all x, $f(x) \leq f(y)$ implies $x \leq y$, and for all z, $z \leq f(z)$" has top-level sturcture **AND**. This assertion has two free-variables f (standing for a function) and y standing for a real number. The form of the sentence is $(\forall x, (C(x, y, f) \implies D(x, y))) \wedge (\forall z, E(z, f))$ where $C(x, y, f)$ is the assertion "$f(x) \leq f(y)$" and $D(x, y)$ is the

assertion "$x \leq y$" and $E(z, f)$ is the assertion"$z \leq f(z)$". Notice that the way this assertion is put together: $C(x, y, f)$ is combined with $D(x, y)$ using $\implies$ and then the combined assertion is given a universal quantifier in x, to produce an assertion $G(y, f)$. Separately, $E(z, f)$ is given a universal quantifier in z to produce an assertion $H(f)$. The entire assertion is then constructed as $G(y, f) \wedge H(f)$, and this final step in building the assertion provids the top-level structure

Remark 6.3. There is some possible ambiguity in the meaning of the original assertion because we need to know how the parts of the sentence are grouped. A different interpretation would be "For all x, $(C(x, y, f) \implies D(x, y) \wedge \forall z, E(z, f))$. In this interpretation, the top-level structure is $\exists x, J(x)$, where $J(x)$ represents the inner assertion above. How do we know which is meant? There are subtle reasons to prefer the first interpretation, but the ambiguity means that the wording of the sentence is dangerous.

> *A sentence that does not have a single clear interpretation (within its context) is unacceptable in mathematical communication.*

To be safe we might add commas, and separating words, such as "also" and write "For all x, $f(x) \leq f(y)$ implies $x \leq y$), and also, for all z, $z \leq f(z)$."

If necessary we might use parentheses as we do in symbolic expressions such as: "(For all x, $f(x) \leq f(y)$ implies $x \leq y$) and (for all z, $z \leq f(z)$)" although the use of parentheses in this way in mathematical English should be avoided if possible.

Example 6.3. The assertion "$x^2 \geq x$ implies that there is a number z such that $z^2 + 1 \leq 2xz$ and $z \geq 1$" has top-level structure $A(x) \implies B(x)$ where $A(x)$ is the sentence "$x^2 \geq x$" and $B(x)$ is the sentence " there is a number z such that $z^2 + 1 \leq 2xz$ and $z \geq 1$."

Since $B(x)$ is a compound assertion, we can take our analysis further to determine the top-level structure of $B(x)$. Here $B(x)$ has top-level structure $\exists z C(x, z)$ where $C(x, z)$ is the assertion "$z^2 + 1 \leq 2xz$ and $z \geq 1$". Also, $C(x, z)$ has top-level structure $D(x, z) \wedge E(z)$, where $D(x, z)$ is the assertion "$z^2 + 1 \leq 2xz$" and $E(z)$ is the assertion "$z \geq 1$".

Example 6.4. Consider the assertion "It is not the case that both n is prime and $n + 1$ is a square." (Note: A number is said to be a *square* if it is the square of some integer.) The top-level structure is $\neg C$ where $C = A(n) \wedge B(n))$, and $A(n)$ is "n is prime" and $B(n)$ is "$n+1$ is a square".

Making deductions. When we do mathematical proofs, we always work within a scenario that provides certain active objects, and active assumptions about the objects. A deduction is a conclusion about the objects made by applying universal principles to the active assumptions. There are two main types of deductions that are used in a mathematical proof, *mathematical deduction* and *logical deduction*. Let's start with a very simple illustration.

Example 6.5. Consider the scenario with:

Active objects: Real numbers x, y.

Active assumption: $3x + y \leq 4$

Here are two correct deductions:

- $3x + y \leq 4$ or $x \geq y$.
- $12x + 4y \leq 16$.

The first deduction uses a *universal principle of logic*: Whenever P and Q are assertions and P is true, then "P OR Q" is also true. Notice that this principle comes from the *logical structure* of the two assertions, and does not depend at all on what P and Q say.

The second deduction is obtained by applying a *universal principle of mathematics*: Whenever a,b,c are real numbers and $a \leq b$ and $c \geq 0$ we have $ac \leq bc$. We apply this principle with $a = 3x + y$, $b = 4$ and $c = 4$. Notice that, unlike the first deduction, here we are using the mathematical meaning of the assumption in order to make the deduction.

We refer to the first kind of deduction, which only depends on the logical structure of the assertions, as a *logical deduction*, and the second kind of deduction as a *mathematical deduction*.

For the rest of this section, we study logical deductions. As the above example shows, when we make a logical deduction we don't care about the meaning of the sentences involved, we only care about their logical structure.

Logical Equivalence If we have a logical expression such as $A \implies (B \wedge C)$ we can determine the truth or falsity of the expression by knowing which of A, B and C are true. We can do determine this by means of a truth table:

$TV(A)$	$TV(B)$	$TV(C)$	$TV(B \wedge C)$	$TV(A \implies (B \wedge C))$
T	T	T	T	T
T	T	F	F	F
T	F	T	F	F
T	F	F	F	F
F	T	T	T	T
F	T	F	F	T
F	F	T	F	T
F	F	F	F	T

The first three columns represent all possible truth values for A, B and C. The fourth column combines the B and C columns using the combination rule for $\wedge$, and the fifth column combines the A column and the $B \wedge C$ column using the combination rule for $\implies$. From this table we can determine the value of $A \implies (B \wedge C)$ for each possible truth value of A, B and C.

If we use the same process to analyze the expression $(B \vee \neg A) \wedge (C \vee \neg A)$, we get the table:

$TV(A)$	$TV(B)$	$TV(C)$	$TV(\neg A)$	$TV(B \vee \neg A)$	$TV(C \vee \neg A)$	$TV((B \vee \neg A) \wedge (C \vee \neg A))$
T	T	T	F	T	T	T
T	T	F	F	T	F	F
T	F	T	F	F	T	F
T	F	F	F	F	F	F
F	T	T	T	T	T	T
F	T	F	T	T	T	T
F	F	T	T	T	T	T
F	F	F	T	T	T	T

The key thing to notice is that the truth values of the two expressions $A \implies (B \wedge C)$ and $(B \vee \neg A) \wedge (C \vee \neg A)$ are the same for each setting of the truth values of A, B and C. This means that these two expressions are *logically equivalent*. We abbreviate this by writing $A \implies (B \wedge C) \equiv (B \vee \neg A) \wedge (B \vee \neg A)$.

Now consider the two assertions:

- n is prime implies that both 4 is a divisor of $n - 1$ and 5 is a divisor of $n - 2$
- 4 is a divisor of $n - 1$ or n is not prime, and 5 is a divisor of $n - 2$ or n is not prime.

If we let A be "n is prime", and B be "4 is a divisor of $n - 1$" and C be "5 is a divisor of $n - 2$", then the first sentence is represented by $A \implies (B \wedge C)$ and the second is $(B \vee \neg A) \wedge (C \vee \neg A)$. Since these two expressions are logically equivalent, we conclude that the these two mathematical assertions are equivalent. In this example, the two assertions have a free variable n, and we know that for each setting of n, either both assertions are **T**or both are **F**.

One always use the truth table method to check whether two logical expressions are logically equivalent, but there are methods that are easier to use. We start with certain basic equivalence rules that allow us to say that two logical expressions are equivalent, and by using these rules we can generate all logical equivalences.

The first set of rules concerns equivalences between simple combinations of two assertions:

Equivalence			Rule Name
$A \vee B$	$\equiv$	$\neg A \implies B$	**OR** to **IMPLIES**
$A \implies B$	$\equiv$	$B \vee \neg A$	**IMPLIES** to **OR**
$A \implies B$	$\equiv$	$\neg B \implies \neg A$	Contrapositive
$A \iff B$	$\equiv$	$(A \implies B) \wedge (B \implies A)$	Definition of $\iff$

Remark 6.4. It is important to remember that $A \implies B$ and $B \implies A$ are *not equivalent.*

The next set of rules involves combining three assertions. The first two are reminiscent of the distributive law of multiplication over addition in algebra.

Equivalence			Rule Name
$(A \wedge B) \vee C$	$\equiv$	$(A \vee C) \wedge (B \vee C)$	logical distributive law
$(A \vee B) \wedge C$	$\equiv$	$(A \wedge C) \vee (B \wedge C)$	logical distributive law
$A \implies (B \implies C)$	$\equiv$	$(A \wedge B) \implies C$	Nested Implication

The next group of equivalences are called *negation rules*. They replace an assertion whose high-level structure is "negation" to one whose high-level structure is something else.

Equivalence			Rule name
$\neg(A \wedge B)$	$\equiv$	$\neg A \vee \neg B$	Negation of $\wedge$
$\neg(A \vee B)$	$\equiv$	$\neg A \wedge \neg B$	Negation of $\vee$
$\neg(A \implies B)$	$\equiv$	$A \wedge \neg B$	Negation of $\implies$
$\neg(\neg A)$	$\equiv$	A	Double negation

We also have equivalence rules involving quantifiers. The first two rules, which are negation rules for quantifiers are the most important.

Equivalence			Rule name
$\neg\forall x, B(x)$	$\equiv$	$\exists x, \neg B(x)$	Negation of $\forall$
$\neg\exists x B(x)$	$\equiv$	$\forall x, \neg B(x)$	Negation of $\exists$
$\forall x \forall y, A(x,y)$	$\equiv$	$\forall y \forall x, A(x,y)$	Interchanging $\forall$
$\exists x \exists y, A(x,y)$	$\equiv$	$\exists y \exists x, A(x,y)$	interchanging $\exists$
$\exists! x, A(x)$	$\equiv$	$(\exists x A(x)) \wedge \forall x \forall y (A(x) \wedge A(y)) \implies (x = y)$	Definition of $\exists!$

Remark 6.5. The final equivalence is the definition of the unique existential quantifier. As discussed in Section 4 the assertion $\exists! x, A(x)$ is really the combination (AND) of two assertions. The first says that there is an x making $A(x)$ true, and the second says that if x and y both make A true then x must be the same as y.ers

Remark 6.6. The third and fourth equivalence say that if we form an expression of the form $\forall x \forall y A(x,y)$, or $\exists x \exists y A(x,y)$,so that x and y have the same quantifier, then we can switch the order of the quantifiers.

However, if the quantifiers are different, then the order matters a great deal. The logical forms $\forall x, \exists y, A(x,y)$ and $\exists y, \forall x, A(x,y)$ have very different meaning, as was discussed in Section 4.

Also, in the case of the quantifier $\exists!$, we cannot interchange the order of $\exists! x, \exists! y$.

Reformulations of a universal assertion The same assertion can be reformulated in many different ways. Sometimes the reformulation is based on modifying words using the fact that in English there are different ways to say the same thing. Other times, the reformulation makes use of logical equivalences. Here we'll see how this works in a simple example.

- For every positive integer n if n is prime and $n > 2$ then n is odd.

This hypothesis of the universal assertion is the scenario:

Input. A positive integer n Assumption: n is prime and $n > 2$
and the conclusion is n is odd.

Here are different ways to express this universal assertion:

1. Every prime number greater than 2 is odd.
2. Each prime number greater than 2 is odd.
3. For all prime integers n that are greater than 2, n is odd.
4. For all prime integers n satisfying $n > 2$, n is odd.
5. For all integers n such that n is prime and $n > 2$, n is odd.
6. For all prime integers n, if $n > 2$ then n is odd.
7. For all integers n, if n is prime and $n > 2$ then n is odd.
8. There are no even primes larger than 2.
9. For all integers n, if n is even and $n > 2$ then n is not prime.

Look at each one of these carefully. All of them express the same universal proposition in different language. The words "every" and "for all" clue you in that this is a universal proposition. The words "satisfying" and "such that" after the "For all n" are clues that what follows is part of the assumption.

Assertion 8 does not start with "Every" or "For all", the statement starts "There are no ...". This is expressing the negation of the existential assertion "There is an even prime number larger than 2". We saw already that every universal assertion is the negation of an existential assertion.

Assertion 9 has the form of a universal assertion, but it seems to scramble the assumption and the conclusion. We can show that it means the same as the original assertion by using the logical equivalence rules presented earlier in this section.

The assertion we're considering is built from 3 atomic assertions: $A =$"n is prime", $B =$"$n > 2$" and $C =$"n is odd". The symbolic representation of Assertion 1 is "$\forall n, (A \wedge B) \implies C$", and the symbolic representation of Assertion 9 is "$\forall n, (\neg C \wedge B) \implies \neg A$". The fact that these two mean the same thing follows by showing that the two propositional expressions $(A \wedge B) \implies C$ and $(\neg C \wedge B) \implies \neg A$ are logically equivalent. For this we use the equivalence rules of Section 4.1. We use the abbrevation $\equiv$ for "is logically equivalent to".

	Equivalence		Rule Used
$(A \wedge B) \implies C$	$\equiv$	$A \implies (B \implies C)$	Nested Implication
	$\equiv$	$\neg(B \implies C) \implies \neg A$	Contrapositive
	$\equiv$	$(\neg C \wedge B) \implies \neg A$	Negation of Implication.

Other types of logical deductions As mentioned, when we do logical deductions, we start with a mathematical scenario, with certain active assumptions. We make deductions based on the active assumptions. So far, we've discussed deductions based on logical equivalence. If we have one active assertion A and another assertion B that is logically equivalent to A, we can add deduce B.

There are other types of important deductions, which are given below. As usual, A, B and C represent arbitrary assertions. The form of each deduction rule is: if we are given (as active assumptions) a particular assertion, or list of assertions, then we can deduce another assertion. The meaning is this, any substitution of A, B and C by assertions that make the assumptions true, must also make the deduction true. This is different from logical equivalence because there may be ways to make the deduction true that don't make all of the assumptions true.

The first three deduction rules are quite obvious, and the other two are also intuitively true. The last two are quite important. The name of the fourth is standard terminology coming from Latin.

Assumption(s)	Deductions	Rule Name
A	$A \vee B$	
$A \wedge B$	A	
A, B	$A \wedge B$	
A, $A \implies B$	B	Modus ponens
$A \implies B$, $B \implies C$	$A \implies C$	Chain of implications

Each of these can be demonstrated by a building a proof table for the assumptions and the deduction, and checking that whenever all of the assumptions are true, the deduction is also.