Math 108 - Introduction to Abstract Mathematics
Supplementary Lecture Notes [1]

1 Mathematical rigor vs. scientific rigor

Did the sun rise today? Did it rise every day of your life so far? Your answers to these two questions are probably a resounding 'yes', because we tend to trust what we observe. This kind of *empiricism* - basing our notion of what is true on what we observe in the natural world - is the basis for most of modern science.

Scientists would go one step further, and ask: Will the sun rise tomorrow? They might say yes, based on the fact that it has risen every morning so far. However, this does not provide a convincing argument that it will rise tomorrow. On the day before your 20th birthday, you will have been less than 20 years old on every day before that, and yet that will no longer be true on the next day!

In order to provide more conclusive proof of their claims, scientists are forced to bring in mathematics, which is the only setting in which we can be 100% certain of an answer or claim, given agreed-upon underlying assumptions. We can come up with simple mathematical models (such as Newton's laws of gravitation) that simultaneously explain many of our observations, and then use these models to predict that the sun will indeed rise again tomorrow.

In this class, we will be focusing on the part that comes after making the mathematical models. That is, given a mathematical system, what methods can we use to draw conclusions with 100% certainty? This is the art of *mathematical proof*.

2 What is a proof?

Simply stated

> A *proof* is an **explanation** of why a statement is **objectively correct**.

Thus we have two goals for our proofs: first we want to verify that it is **objectively correct**, and second we want to be able to most effectively and elegantly **explain** that to our audience.

The Proof Spectrum: Rigor vs. Elegance

These two goals are sometimes in conflict. In order to be absolutely certain our proof is correct, we need to be exceedingly careful and rigorous. In order to be clear in our exposition we need to be succinct and elegant. It's difficult to do both simultaneously!

On the one hand mathematical proofs need to be rigorous - we naturally want our proofs to be correct. One way to ensure they are correct is to have them checked by a computer. (Note that checking to see if a proof is correct is much easier for a computer to do than finding a proof in the first place.) Proofs that can be checked by a computer are often called **formal proofs**.

On the other hand, most mathematicians are attracted to mathematics because of its intrinsic beauty. A proof that communicates the key ideas of a proof to the reader in a succinct and beautiful way is very effective for its expository properties, even if it is not as rigorous as a formal proof. Such a proof is called a **traditional proof**. The legendary mathematician Paul Erdös always spoke of "The Book", an imaginary book in which God had written down the best and most elegant proofs for mathematical theorems. When he saw any particularly inspiring proof he would exclaim "That proof is from 'The Book'!"

We will strive for both rigor and elegance in our proofs by studying both highly rigorous formal proofs and more elegant traditional proofs as we proceed through the course.

[1] These notes are adapted primarily from materials created for *Prove it! Math Academy*, an advanced summer math program for high school students focused on the transition to abstract mathematics and proof writing.

Formal Proof Systems

Definition 1. A **Formal Proof System** (or **Formal Axiom System**) consists of

1. A set of expressions $\mathcal{S}$, called the **statements**.
2. A set of rules $\mathcal{R}$, called the **rules of inference**.

Each rule of inference has zero or more inputs called **premises** and one or more outputs called **conclusions**. Most premises and all conclusions of a rule of inference are statements in the system.[2] There also may be conditions on when a particular rule of inference can be used.

Definition 2. An **axiom** is a conclusion of a rule of inference that has no premises.

Definition 3. A statement Q in a formal axiom system is **provable from** premises $Q_1, \ldots, Q_n$ if either:

1. Q is one of the premises $Q_1, \ldots, Q_n$, or
2. Q is a conclusion of a rule of inference whose premises are provable from $Q_1, \ldots, Q_n$.

In particular, if Q is an axiom, then Q is provable from no premises at all!

Definition 4. If Q follows from no premises in a formal axiom system, we say that Q is **provable** in the system. A provable statement is called a **theorem**.

Definition 5. A **proof** of a statement in a formal axiom system is a sequence of applications of the rules of inference that show that the statement is a theorem in that system.

Notation. If Q is provable from premises $P_1, \ldots, P_n$ in a formal system we can denote this symbolically as

$$P_1, \ldots, P_n \vdash Q$$

It is also commonplace to refer to such an expression as a theorem. To prove such a theorem is to give a proof of Q in the same formal system where additionally the premises are 'Given' as axioms.

Toy Proofs

There are several examples of simple Formal Proof Systems available online at

`proveitmath.org/toyproofs`

Scrambler is a formal proof system where the statements are finite sequences of colors. The Rules of Inference are permutations of these sequences (and so have one premise and one conclusion each). The goal is to apply the Rules to show that a given sequence of colors is provable from another given sequence of colors.

Trix Game is a formal proof system where the statements are positive integers. There are only two Rules of Inference, both of which take a single positive integer as a premise, and return a single positive integer as their conclusion. This system illustrates a rule that has a condition on when you can use it. The goal is to show that a given positive integer is provable from the premise 1 in the system.

Circle-Dot is a formal proof system where the statements are just finite sequences of one or more circles and dots. This system has many of the features of actual mathematical formal axiom systems. There are five rules of inference, two of which are axioms. The goal is to prove various circle-dot strings in the system.

3 The Language of Mathematics

The toy proof systems above were just warmups - it is now time to build up the axiom systems on which mathematics is founded! In particular, we need to understand what a valid mathematical **statement** (or **proposition**) is in the language of mathematics. The building blocks of this language are described by the following terms.

[2] Other common premises are variable declarations, constant declarations, and subproofs.

Expressions and Statements

As in any language, each topic in mathematics defines certain arrangements of symbols to be valid *expressions* about that topic. Each expression can optionally have an associated property called its *type*, which describes the kinds of mathematical objects that the expression might represent.

Example 6. For example, the expression "*Prove it! Math is awesome.*" is a valid expression in the English language. On the other hand "`ls -a`" is not a valid expression in the English language, but is a valid expression in some programming languages.

Example 7. Similarly, in a high school math class "$\frac{x^2-1}{x^2+1}$" might be a valid expression whose type is 'real number', while the expression "cos" might be a valid expression whose type is 'function'. On the other hand, an expression like "$5 \underset{\neq}{\uparrow} \overset{a}{\ominus}$" probably isn't defined to be a valid expression in any mainstream topic in mathematics.

In general, a **statement** is any valid expression (in any language) whose type is 'statement'. Thus, one of the first tasks in defining (resp. learning) a new topic in mathematics is to define (resp. learn) what expressions are designated as having type 'statement' in that language.

Example 8. In English, statements are expressions which can be either true or false. For example, "*Math 108 is awesome.*" is either true or false[3] and is therefore a statement. Expressions like "*Oh, my!*" or "*Do you like ice cream?*" are valid English expressions but are neither true nor false and therefore are not statements in English.

Example 9. In a typical math course, expressions that are either true or false, like "$1 + 1 = 2$", "91 *is prime*", and "$x^2 = \cos(x)$", are also considered to be statements. Note that we don't need to know if such a statement is true or false, just that it is either true or false. Similarly, expressions like "42", "$\{1, 2, 3\}$", and "$x^3 + 1$" are neither true nor false and therefore are not statements.

Constants and Variables

An **identifier** is a name or label for something. Several identifiers can also be combined to form larger expressions called **compound expressions** (but identifiers are not compound expressions themselves). An identifier that refers to a unique, specific object is called a **constant**. An identifier that refers to a single, but unspecified, object is called a **variable**.

Example 10. Proper names are usually constants in the English language. For example, "*Paris*", "*Professor Gillespie*", and "*Groot*" are all intended to refer to a unique, specific thing. On the other hand, when filling out a form with blanks containing "FIRST NAME GOES HERE" or "CELL", those identifiers can be thought of as variables that can represent any first name or any phone number that might be entered into the form.

Example 11. In standard mathematics, identifiers like "π", "1024", "ln", "$+$" and "$\cap$" usually represent constants, whereas identifiers like x, n, P, α, and a_0 frequently represent variables.

Example 12. In the Circle-Dot system the only statements are circle-dot strings like "$\bullet\circ\bullet\bullet$" or "$\bullet\bullet\circ$". All of those expressions are constants.

Both constants and variables can have a type, and in that case represent an object or expression of that type.

Example 13. In Example 10 the variable "CELL" might have type 'phone number' and someone filling out the form might find that it rejects any expression that cannot be interpreted as a phone number.

Example 14. In the Rules of Inference for the Circle-Dot system the expressions W and V are variables of type 'circle dot string' as they are placeholders in the rule for an unspecified circle-dot string.

[3] In this case it is obviously true.

Substitution and Lambda Expressions

We can prefix an expression E to form the expression "$\lambda x, E$" (or "$x \mapsto E$") to indicate that all occurrences[4] of x in E are a variable that represents the same unspecified object of the same type as x. These prefixed expressions are called *lambda expressions* (or *anonymous functions*).

Such expressions can be *applied* to an expression a having the same type as x to form a new expression, $(\lambda x, E)(a)$ which has the same type as E. These can be further simplified to the expression obtained by replacing all occurrences[5] of x in E with a. If we give a name to a lambda expression, e.g., define f to be $\lambda x, E$ then the expression $(\lambda x, E)(a)$ is just the usual notation for function application $f(a)$.[6]

Example 15. In high school algebra, if x is a variable of type integer then $(\lambda x, x^2 + x + 1)(3)$ simplifies to $3^2 + 3 + 1$. Similarly, $(\lambda x, x + y)(3)$ simplifies to $3 + y$ while $(\lambda y, x + y)(3)$ simplifies to $x + 3$.

Two lambda expressions are said to be *equivalent* if they simplify to the same or equivalent things when applied to any argument. Renaming all occurrences of x in $\lambda x, E$ with a new identifier always produces a lambda expression that is equivalent to the original.

Example 16. In the previous example the lambda expression $(\lambda x, x^2+x+1)$ is equivalent to $(\lambda y, y^2+y+1)$. They both simplify to the same expression when applied to the same argument. For example, $(\lambda x, x^2 + x + 1)(3)$ and $(\lambda y, y^2 + y + 1)(3)$ both simplify to $3^2 + 3 + 1$.

Another common situation where we can simplify a lambda expression $\lambda x, E$ is when the expression E does not contain x. In this situation $(\lambda x, E)(a)$ simplifies to just E for every a, and thus we can say that $\lambda x, E$ simplifies to just E in that case.

Example 17. The expression $(\lambda x, \cos(x))(a)$ simplifies to $\cos(a)$ for every argument a. Thus the expression cos can be though of as the lambda expression $(\lambda x, \cos(x))$ since they both simplify to the same expression when applied to an argument a.

4 Rules of Inference in Mathematics

In the Circle-Dot system Axiom A is a rule of inference that says from no premises we can conclude $\bigcirc\bullet$. Rule 1, however, is technically not a rule of inference, but rather an infinite family of rules of inference, one for each choice of circle-dot strings we can substitute for the variables W and V. From this perspective we can think of Rule 1 as the lambda expression,

$$\lambda W, \lambda V, (WV, VW \vdash W)$$

So that, for example, substituting $\bigcirc$ for W and $\bullet$ for V produces the rule

$$\begin{aligned}(\lambda W, \lambda V, (WV, VW \vdash W))(\bigcirc)(\bullet) &= (\lambda V, (\bigcirc V, V\bigcirc \vdash \bigcirc))(\bullet) \\ &= (\bigcirc\bullet, \bullet\bigcirc \vdash \bigcirc)\end{aligned}$$

which allows us to conclude $\bigcirc$ from the premises $\bullet\bigcirc$ and $\bigcirc\bullet$.

Most rules of inference in mathematics are more similar to Rule 1 than to Axiom A in this sense – they are really lambda expressions which generate an entire family of specific rules of inference, one for each choice of variable in the statement of the rule. Because this is so common, we usually omit the lambda prefixes, and use the convention that any variable W that appears in the premises or conclusion of a rule of inference can be replaced with an expression of the same type to form a particular instance of that rule of inference.

[4]These refer to free occurrences - see the quantifiers handout for details.

[5]See footnote 2. Also no free identifier in a should become bound as a result of the substitution.

[6]Indeed, in precalculus they usually write $f(x) = x^3$ instead of writing $f = (\lambda x, x^3)$, but the latter is usually what they mean.

Recipe Notation for Rules of Inference

Notation. A rule of inference having premises $P_1, \ldots, P_k$ and conclusions $Q_1, \ldots, Q_n$ can be expressed in *recipe notation* as

Show: P_1

$\vdots$

Show: P_k

Conclude: Q_1

$\vdots$

Conclude: Q_n

Some rules of inference have a premise of the form

$$(P_1, \ldots, P_k \vdash Q)$$

In that case, this premise is validated by including a *subproof* in that proof that Q can be proved from the given premises (which do not need to be justified by a rule of inference). We denote this in recipe notation as an indented 'assume-block' as illustrated below.

Exercise 18. Suppose we have a rule of inference that justifies the following.

$$P \text{ or } Q, (P \vdash R), (Q \vdash R) \vdash R$$

where P, Q, and R are any mathematical statements. Then we would express this rule in recipe notation as

Show: P or Q

  Assume P

  Show: R

  $\leftarrow$

  Assume Q

  Show: R

  $\leftarrow$

Conclude: R

In this, everything between an *Assume* and the following $\leftarrow$ (the 'end assumption' symbol) is a *subproof* that demonstrates the corresponding premise in the rule of inference. We indent such assumption blocks in our proofs. Subproofs can be nested, and the level of indentation corresponds to the level of nesting. Assumptions (lines that start with *Assume*) do not need to be justified by a rule of inference. We sometimes say that they are 'Given'.

Note that we do include the word "*Assume*" in the proof itself, but not the words "Show" or "Conclude" which are just instructions to the proof author (as opposed to the reader) for how to follow the recipe for this rule of inference.

5 Natural Deduction

We now turn our attention to a formal axiom system that is based on one first formulated by Gerhard Gentzen in 1934 as a formal system that closely imitates the way mathematicians actually reason when writing traditional expository proofs.

Propositional Logic

The Statements of Propositional Logic

The statements of propositional logic are expressions who have a *truth value* which is either true or false.

Definition 19. Let P, Q be statements. Then the five expressions "$\neg P$", "$P \wedge Q$", "$P \vee Q$", "$P \Rightarrow Q$", and "$P \Leftrightarrow Q$" are also statements whose truth values are completely determined by the truth values of P and Q as shown in the following table:

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

We can also write 'not' for $\neg$, 'if and only if' for $\Leftrightarrow$, 'implies' for $\Rightarrow$, ' and ' for $\wedge$, and ' or ' for $\vee$. A statement of the form '$P \Rightarrow Q$' is called a *conditional statement* or an *implication*, and can be written in English as 'P implies Q', 'if P then Q', 'Q follows from P', or 'Q, if P'.

Definition 20. The statements $\mathcal{S}$, of Propositional Logic consists of

1. Atomic Statements that do not contain any of the five logical operators, and
2. Compound Statements that are one of the five forms, $\neg P$, $P \wedge Q$, $P \vee Q$, $P \Rightarrow Q$, or $P \Leftrightarrow Q$ where P and Q are any elements of $\mathcal{S}$.

Note: In compound statements we usually put parentheses around the statements P or Q involved. For instance if P is the statement 'P or Q' and Q is the statement 'R and S' then $P \Rightarrow Q$ should be written

$$(P \text{ or } Q) \Rightarrow (R \text{ and } S)$$

in order to avoid the confusion that 'P or $Q \Rightarrow R$ and S' might actually mean something like P or $(Q \Rightarrow (R$ and $S))$. In order to cut down on parentheses, we assign a **precedence** order for our operators, meaning we apply the operators in the following order (from highest to lowest):

Precedence of Notation
parentheses, brackets, $(), \{\}, []$ etc.
not
and , or
$\Rightarrow$
$\Leftrightarrow$

The Rules of Inference of Propositional Logic

Rules of Inference	
and +	and −
Show: W **Show:** V **Conclude:** W and V	**Show:** W and V **Conclude:** W **Conclude:** V
$\Rightarrow +$	$\Rightarrow -$ (modus ponens)
Assume W **Show:** V $\leftarrow$ **Conclude:** $W \Rightarrow V$	**Show:** W **Show:** $W \Rightarrow V$ **Conclude:** V
$\Leftrightarrow +$	$\Leftrightarrow -$
Show: $W \Rightarrow V$ **Show:** $V \Rightarrow W$ **Conclude:** $W \Leftrightarrow V$	**Show:** $W \Leftrightarrow V$ **Conclude:** $W \Rightarrow V$ **Conclude:** $V \Rightarrow W$
or +	or − (proof by cases)
Show: W **Conclude:** W or V **Conclude:** V or W	**Show:** W or V **Show:** $W \Rightarrow U$ **Show:** $V \Rightarrow U$ **Conclude:** U
not + (proof by contradiction)	not − (proof by contradiction)
Assume W **Show:** $\rightarrow\leftarrow$ $\leftarrow$ **Conclude:** not W	*Assume* not W **Show:** $\rightarrow\leftarrow$ $\leftarrow$ **Conclude:** W
$\rightarrow\leftarrow +$	copy
Show: V **Show:** not V **Conclude:** $\rightarrow\leftarrow$	**Show:** W **Conclude:** W

Remarks:

- The symbol $\leftarrow$ is an abbreviation for "end assumption".
- The symbol $\rightarrow\leftarrow$ is called "contradiction" and represents the logical constant FALSE.
- The italicized word *Assume* is actually entered as part of the proof itself, it is not just an instruction in the recipe like the words '**Show:** ' and '**Conclude:** '.
- The inputs *"Assume* -" and "$\leftarrow$" are not themselves statements that you prove or are given, but rather are inputs to rules of inference that may be inserted into a proof at any time. There is no reason however, to insert such statements unless you intend to use one of the rules of inference that requires them as inputs.
- The statement following an *Assume* is the same as any other statement in the proof and can be used as an input to a rule of inference.

- Statements in an *Assume*-$\leftarrow$ block can be used as inputs to rules of inference whose conclusion is also inside the same block only. Once a *Assume* is closed with a matching $\leftarrow$, only the entire block can be used as an input to a rule of inference. The individual statements within a block are no longer valid outside the block. We usually indent and *Assume*-$\leftarrow$ block to keep track of what statements are valid under which assumptions.

Definition 21. A compound statement of propositional logic is called a *tautology* if it is true regardless of the truth values the atomic statements that comprise it. (Its "truth table" contains only T's.)

It can be shown that a statement can be proved with Propositional Logic if and only if the statement is a tautology.

6 Formal Proof vs. Traditional Proof Style

One way to write down a formal proof of a theorem is called a **two column proof**. This style of proof consists of a sequence of numbered lines containing statements, reasons, and references to premises. Every line contains exactly one statement (or declaration - see below), and the reason given on that line is the name of a rule of inference for which the statement on that line is the conclusion. If the rule of inference has premises, the reason is followed by the line numbers containing the statements (or variable declarations) which are the premises that the rule is being applied to. References to premises can only refer to lines which appear earlier in the same proof which are not contained in a subproof that has been closed. Subproofs used as a premise are cited by listing the range of line numbers comprising the subproof.

Example 22. Let P and Q be statements. Prove the following case of DeMorgan's Law, namely that

$$\neg P \text{ or } \neg Q \Rightarrow \neg(P \text{ and } Q)$$

Proof.

1.	Assume $\neg P$ or $\neg Q$	-
2.	Assume $\neg P$	-
3.	Assume P and Q	-
4.	P	by and $-$; 3
5.	$\rightarrow\leftarrow$	by $\rightarrow\leftarrow +$; 2,4
6.	$\leftarrow$	-
7.	$\neg(P \text{ and } Q)$	by not+; 3,5,6
8.	$\leftarrow$	-
9.	$\neg P \Rightarrow \neg(P \text{ and } Q)$	by $\Rightarrow +$; 2,7,8
10.	Assume $\neg Q$	-
11.	Assume P and Q	-
12.	Q	by and $-$; 11
13.	$\rightarrow\leftarrow$	by $\rightarrow\leftarrow +$; 10,12
14.	$\leftarrow$	-
15.	$\neg(P \text{ and } Q)$	by not+; 11, 13, 14
16.	$\leftarrow$	-
17.	$\neg Q \Rightarrow \neg(P \text{ and } Q)$	by $\Rightarrow +$; 10,15,16
18.	$\neg(P \text{ and } Q)$	by or $-$; 1,9,17
19.	$\leftarrow$	-
20.	$\neg P$ or $\neg Q \Rightarrow \neg(P \text{ and } Q)$	by $\Rightarrow +$; 1,18

□

Notice that when a rule of inference has a subproof for a premise, we indicate this by citing the line numbers for the assumption, the conclusion, and the end of assumption block indicator ($\leftarrow$) e.g., as shown in line 7 above.

A **traditional proof** or **informal proof** would be more explanatory, like an essay, describing the steps and reasons in word-wrapped form. The above proof could be written in a traditional way as follows.

Theorem 23. *Let P and Q be statements. Then* ($\neg P$ or $\neg Q$) $\Rightarrow \neg(P$ and $Q)$.

Proof. We wish to show that if at least one of 'not P' or 'not Q' is true, then it is not the case that both P and Q are true.

So, assume $\neg P$ or $\neg Q$. We wish to show $\neg(P$ and $Q)$. As a shorthand, define R to be the statement $\neg(P$ and $Q)$. We will prove R, using the "or $-$" rule of inference starting from $\neg P$ or $\neg Q$. In other words, we want to show that both $\neg P$ and $\neg Q$ imply R.

First, suppose $\neg P$ is true. Assume for contradiction that P and Q. Then by the and $-$ rule we know P is true, contradicting our assumption of $\neg P$. Thus by $\neg+$ (or proof by contradiction), we can conclude $\neg(P$ and $Q)$. Therefore:

$$\neg P \Rightarrow \neg(P \text{ and } Q).$$

Next, suppose $\neg Q$ is true. A similar argument to the previous paragraph shows that $\neg(P$ and $Q)$ is true in this case as well, so

$$\neg Q \Rightarrow \neg(P \text{ and } Q).$$

By the or $-$ rule, our proof is complete. □

Tips for Informal Proof Writing: There is not a single correct way to word wrap a formal proof into a traditional proof style, but we generally try to use the following guidelines to make our proofs more readable:

- Tell the reader what your proof strategy is up front. Notice in the proof above we say up front that we will be using the 'or $-$' rule of inference (which we could have also called "proof by cases"). This allows the reader to more easily follow the remainder of the argument.
- Never start a sentence with a math expression. We would never start the sentence with $\neg P$, for instance.
- Use "we" rather than "I", as in "We now show that..." rather than "I will now show that..."
- Put important mathematical expressions in the proof in **display mode** - centered and on a line by themselves - in order to draw attention to them. You can also label such lines with a number on the right if you need to refer to them later.
- Skip the last few lines of the formal proof if they are clear to anyone familiar with the rules of inference.
- Skip details that will be clear to the reader, but don't skip any important details! (In general, which details you can skip will depend somewhat on your audience.)
- Put an end-of-proof symbol, usually either □ or "QED", to mark the end of your proof!

Exercise 24. Give a formal proof for the reverse case of DeMorgan's Law, namely that

$$\neg(P \text{ and } Q) \Rightarrow \neg P \text{ or } \neg Q$$

Exercise 25. Give a formal proof for yet another case of DeMorgan's Law, namely that

$$\neg(P \text{ or } Q) \Leftrightarrow \neg P \text{ and } \neg Q$$

Famous Tautologies

Try to prove the following tautologies using formal proofs. In all of the following P, Q, R, S are statements.

1. (*Contradiction implies anything*) $\rightarrow\leftarrow \Rightarrow P$
2. (*Double negation*) $\neg\neg P \Leftrightarrow P$
3. (*Modus tollens*) $(P \Rightarrow Q)$ and $\neg Q) \Rightarrow \neg P$
4. (*Transitivity of* $\Rightarrow$) $(P \Rightarrow Q)$ and $(Q \Rightarrow R) \Rightarrow (P \Rightarrow R)$
5. (*Contrapositive*) $(P \Rightarrow Q) \Leftrightarrow (\neg Q \Rightarrow \neg P)$
6. (*No contradictions*) $\neg \rightarrow\leftarrow$
7. (*Excluded middle*) P or $\neg P$
8. (*Alternate* or $-$) $(P$ or $Q)$ and $\neg P \Rightarrow Q$
9. (*DeMorgan's Law*) $\neg(P$ or $Q) \Leftrightarrow (\neg P$ and $\neg Q)$
10. (*DeMorgan's Law*) $\neg(P$ and $Q) \Leftrightarrow (\neg P$ or $\neg Q)$

Commutativity

1. P and $Q \Leftrightarrow Q$ and P
2. P or $Q \Leftrightarrow Q$ or P
3. $(P \Leftrightarrow Q) \Leftrightarrow (Q \Leftrightarrow P)$
4. Give examples of statements, P and Q such that $(P \Rightarrow Q) \Rightarrow (Q \Rightarrow P)$ is false.

Associativity

1. $((P$ and $Q)$ and $R) \Leftrightarrow (P$ and $(Q$ and $R))$
2. $((P$ or $Q)$ or $R) \Leftrightarrow (P$ or $(Q$ or $R))$
3. $((P \Leftrightarrow Q) \Leftrightarrow R) \Leftrightarrow (P \Leftrightarrow (Q \Leftrightarrow R))$

Distributivity

1. P and $(Q$ or $R) \Leftrightarrow (P$ and $Q)$ or $(P$ and $R)$
2. P or $(Q$ and $R) \Leftrightarrow (P$ or $Q)$ and $(P$ or $R)$

Transitivity

1. $((P \Rightarrow Q)$ and $(Q \Rightarrow R)) \Rightarrow (P \Rightarrow R)$
2. $((P \Leftrightarrow Q)$ and $(Q \Leftrightarrow R)) \Rightarrow (P \Leftrightarrow R)$

7 Common proof shortcuts for "semi-formal" and informal proofs

So far we have only allowed axioms, rules of inference, assumptions, end assumptions, definition statements, and theorem statements in the Reasons column of our two column proofs. But in practice, to cut down on the number of steps of a proof, mathematicians often take sequences of steps that are routine and come up everywhere and skip them entirely, only stating the main reason that leads to the conclusion.

We'll introduce these shortcuts throughout the class, but here are three that we can start with:

1. **If and only if substitution:** We can often treat the '$\Leftrightarrow$' symbol as statement equality, even going so far as to use substitution. In particular if $P \Leftrightarrow Q$, then we can replace any occurrence of Q in any statement with P or vice versa. (Note that P and Q may themselves be compound statements here.)

   Note that if we only have $P \Rightarrow Q$, we can replace P's with Q's but not vice versa.

2. **Break down theorems or rules of inference into longer rules of inference:** We can "expand" rules of inference into recipes that tell us how to prove each of the inputs. Rather than saying we need both $P \Rightarrow Q$ and $Q \Rightarrow P$ to prove $Q \Leftrightarrow P$, we can expand the inputs into their required inputs:

   $$\begin{array}{l} \textit{Assume} \quad P \\ \textbf{Show: } Q \\ \leftarrow \\ \textit{Assume} \quad Q \\ \textbf{Show: } P \\ \leftarrow \\ \textbf{Conclude: } P \Leftrightarrow Q \end{array}$$

   Or, since showing that $\neg Q \Rightarrow \neg P$ is the same as showing $P \Rightarrow Q$, we can replace that sub-proof with the following:

   $$\begin{array}{l} \textit{Assume} \quad \neg Q \\ \textbf{Show: } \neg P \\ \leftarrow \\ \textit{Assume} \quad Q \\ \textbf{Show: } P \\ \leftarrow \\ \textbf{Conclude: } P \Leftrightarrow Q \end{array}$$

   Both of these are acceptable 'shortcuts' for the $\Leftrightarrow$ + rule.

3. **Transitive chains for $\Leftrightarrow$ or $\Rightarrow$:** To prove $P \Leftrightarrow Q_n$, if we can find a sequence of statements $Q_1, \ldots, Q_n$ for which $P \Leftrightarrow Q_1$, $Q_1 \Leftrightarrow Q_2, \ldots, Q_{n-1} \Leftrightarrow Q_n$ then we can write the following transitive chain of statements:

   $$\begin{array}{rcl} P & \Leftrightarrow & Q_1 \\ & \Leftrightarrow & Q_2 \\ & \Leftrightarrow & Q_3 \\ & \vdots & \\ & \Leftrightarrow & Q_n \end{array}$$

4. **Skip the last line of the proof.**

We can use these in the following example:

Theorem 26. $\neg((\neg P \text{ or } Q) \text{ and } R) \Leftrightarrow ((P \text{ and } \neg Q) \text{ or } \neg R))$

Proof.

1.	Assume $\neg((\neg P \text{ or } Q) \text{ and } R)$	-
2.	$\neg(\neg P \text{ or } Q) \text{ or } \neg R$	DeMorgan's Law; 1
3.	$(\neg\neg P \text{ and } \neg Q) \text{ or } \neg R$	DeMorgan's Law; 2
4.	$(P \text{ and } \neg Q) \text{ or } \neg R$	$\neg\neg P \Leftrightarrow P$; 3
5.	$\leftarrow$	-
6.	Assume $(P \text{ and } \neg Q) \text{ or } \neg R$	-
7.	$(\neg\neg P \text{ and } \neg Q) \text{ or } \neg R$	$\neg\neg P \Leftrightarrow P$; 5
8.	$\neg(\neg P \text{ or } Q) \text{ or } \neg R$	DeMorgan's Law; 6
9.	$\neg((\neg P \text{ or } Q) \text{ and } R)$	DeMorgan's Law; 7
10.	$\leftarrow$	-

□

Or, as a proof by transitive chain, we can prove it this way:

Proof. We have:

$$\begin{aligned} \neg((\neg P \text{ or } Q) \text{ and } R) &\Leftrightarrow \neg(\neg P \text{ or } Q) \text{ or } \neg R \\ &\Leftrightarrow (\neg\neg P \text{ and } \neg Q) \text{ or } \neg R \\ &\Leftrightarrow (P \text{ and } \neg Q) \text{ or } \neg R \end{aligned}$$

□

8 Predicate Logic

We can extend Propositional Logic by adding more statements and rules of inference to those we already have in our formal system. This extended formal system is called **Predicate Logic**.

8.1 Quantifiers

The symbol λ in the lambda expression $(\lambda x, E)$ is an example of a **quantifier**. The thing that all quantifiers have in common is that they **bind variables**. If W is an expression that does not contain any quantifiers, then every occurrence of every identifier that appears in the expression is said to be a **free** occurrence of that identifier.

If a quantifier appears in an expression, there are one or more variables that it binds. All occurrences of the variables that are in the scope of the quantifier (usually everything to the right of it until a scope delimiter for that quantifier is encountered) are called **bound variables**.

Predicate logic extends propositional logic by defining two additional quantifiers.

Definition 27. The symbols $\forall$ and $\exists$ are *quantifiers*. The symbol $\forall$ is called "for all", "for every", or "for each". The symbol $\exists$ is called "for some" or "there exists".

Exercise 28. Translate the following English sentences into statements with quantifiers.

1. Every integer is greater than -4 or less than 6.
2. Every integer is greater than some integer.

3. No integer is greater than every other integer.
4. There is a smallest positive integer.
5. No one loves everybody.
6. Everybody loves someone.
7. Bob loves Alice, but she does not love him.
8. There is a person who is loved by only one person.
9. For natural numbers a and b, a **divides** b if and only if there is a natural number k for which $ak = b$.
10. A natural number is **prime** if its only divisors are 1 and itself.

8.2 Statements

Every statement of Propositional Logic is still a statement of Predicate Logic. In addition we define the following statements.

Definition 29. If x is any variable and W is a lambda expression that simplifies to a statement when applied to any expression having the same type as x, then $(\forall x, W(x))$ and $(\exists x, W(x))$ are both statements.

We say that the *scope* of the quantifier in $(\forall x, W(x))$ and $(\exists x, W(x))$ is everything inside the outer parentheses. Sometimes these parentheses are omitted when the scope is clear from context. All occurrences if x throughout the scope are said to be bound by the quantifier.

8.3 Variable declaration

Before using a free identifier for the first time in any expression in our proofs we should tell the reader what that identifier represents. There are four ways to introduce a new free identifier.

1. It can be declared to be a variable (a variable declaration).
2. It can be declared to be a constant (a constant declaration).
3. It can be defined as temporary new notation, usually as an abbreviation for a larger expression (a notational definition).
4. It can occur free in an expression preceding the proof itself, such as in the statement of the theorem, in a premise that is given, or declared globally prior to the start of the proof (globally declared).

Bound variables do not have to be declared. They can be any identifier you like, as long as that identifier is not in the scope of more than one quantifier that binds it.

8.4 Rules of Inference

The rules of inference for these two quantifiers are as follows. These are expressed in recipe notation.

Rules of Inference for Logical Quantifiers*

$\forall+$	$\forall-$
Let s be arbitrary (variable declaration) **Show:** $W(s)$ $\leftarrow$ **Conclude:** $\forall x, W(x)$	**Show:** $\forall x, W(x)$ **Conclude:** $W(t)$
$\exists+$	$\exists-$
Show: $W(t)$ **Conclude:** $\exists x, W(x)$	**Show:** $\exists x, W(x)$ *For some* c (constant declaration) **Conclude:** $W(c)$

*Restrictions and Remarks:

- In $\forall+$, s must be a new variable in the proof, cannot appear as a free variable in any assumption or premise, and $W(s)$ cannot contain any constants which were produced by the $\exists-$ rule. The indentation and $\leftarrow$ symbol indicate the scope of the declaration of s. Variables s and x must have the same type.
- In $\forall-$ and $\exists+$, no free variable in t may become bound when t is substituted for x in $W(x)$. Variable x and expression t must have the same type.
- In $\exists+$, t can be an expression, and $W(x)$ can be the expression obtained by replacing one or more of the occurrences of t with x. The identifier x cannot occur free in $W(t)$. Variable x and expression t must have the same type.
- In $\exists-$, c must be a new identifier in the proof. Also $W(c)$ must immediately follow the constant declaration for c in the proof. The scope of the declaration continues indefinitely or until the end of the scope of any subproof block or variable declaration scope that contains the constant declaration. Variable x and constant c must have the same type.

  One consequence of this is that it enforces the restriction on $\forall+$ that prohibits any constant declared with $\exists-$ to appear in $W(s)$ because after the application of $\forall+$ any free occurrence of c is no longer in the scope of the original declaration (and therefore undeclared).

Exercise 30. (*Alpha Substitution.)* Show that the choice of bound variable doesn't matter (for new variables in the proof) by giving a formal proof for

$$(\forall x, P(x)) \Rightarrow (\forall y, P(y))$$

and

$$(\exists x, P(x)) \Rightarrow (\exists y, P(y))$$

Example 31. Let's prove yet another DeMorgan's Law using a formal proof in our Predicate Logic.

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

Proof.

1.	Assume $\neg(\forall x, P(x))$	-
2.	Assume $\neg(\exists x, \neg P(x))$	-
3.	Let t be arbitrary	-
4.	Assume $\neg P(t)$	-
5.	$\exists x, \neg P(x)$	$\exists+$; 4
6.	$\rightarrow\leftarrow$	$\rightarrow\leftarrow+$; 2, 5
7.	$\leftarrow$	-
8.	$P(t)$	$\neg-$; 4, 7, 8
9.	$\leftarrow$	-
10.	$\forall x, P(x)$	$\forall+$; 3, 8, 9
11.	$\rightarrow\leftarrow$	$\rightarrow\leftarrow+$; 1, 10
12.	$\leftarrow$	-
13.	$\exists x, \neg P(x)$	$\neg-$; 2, 11, 12
14.	$\leftarrow$	-
15.	$\neg(\forall x, P(x)) \Rightarrow (\exists x, \neg P(x))$	$\Rightarrow+$; 1, 13, 14
16.	Assume $\exists x, \neg P(x)$	-
17.	For some c	const dec
18.	$\neg P(c)$	$\exists-$; 16, 17

19.	Assume $\forall x, P(x)$	-
20.	$P(c)$	$\forall-$; 17, 19
21.	$\rightarrow\leftarrow$	$\rightarrow\leftarrow+$; 20, 18
22.	$\leftarrow$	-
23.	$\neg(\forall x, P(x))$	$\neg+$; 19, 22
24.	$\leftarrow$	-
25.	$(\exists x, \neg P(x)) \Rightarrow \neg(\forall x, P(x))$	$\Rightarrow+$; 16, 23, 24
26.	$\neg(\forall x, P(x)) \Leftrightarrow (\exists x, \neg P(x))$	$\Leftrightarrow+$; 15, 25

□

Example 32. The following proof shows that we can interchange $\forall$ quantifiers with each other:

$$(\forall x, \forall y, P(x, y)) \Rightarrow (\forall y, \forall x, P(x, y))$$

Can you fill in the line number inputs for the rules of inference below?

Proof.

1.	Assume $\forall x, \forall y, P(x, y))$	-
2.	Let s be arbitrary	-
3.	Let t be arbitrary	-
4.	$\forall y, P(t, y)$	$\forall-$
5.	$P(t, s)$	$\forall-$
6.	$\leftarrow$	-
7.	$\forall x, P(x, s)$	$\forall+$
8.	$\leftarrow$	-
9.	$\forall y, \forall x, P(x, y)$	$\forall+$
10.	$\leftarrow$	-
11.	$(\forall x, \forall y, P(x, y)) \Rightarrow (\forall y, \forall x, P(x, y))$	$\Rightarrow+$

□

Example 33. The following proof shows that we can interchange $\exists$ quantifiers with each other:

$$(\exists x, \exists y, P(x, y)) \Rightarrow (\exists y, \exists x, P(x, y))$$

Can you fill in the line number inputs for the rules of inference below?

Proof.

1.	Assume $\exists x, \exists y, P(x, y))$	-
2.	For some t	constant declaration
3.	$\exists y, P(t, y)$	$\exists-$
4.	For some s	constant declaration
5.	$P(t, s)$	$\exists-$
6.	$\exists x, P(x, s)$	$\exists+$
7.	$\exists y, \exists x, P(x, y)$	$\exists+$
8.	$\leftarrow$	-
9.	$(\exists x, \exists y, P(x, y)) \Rightarrow (\exists y, \exists x, P(x, y))$	$\Rightarrow+$

□

Definition 34. If x, y are any variables of the same type and W is a lambda expression not containing x or

y that simplifies to a statement when applied to any expression having the same type as x and y, we define

$$(\exists! x, W(x)) \Leftrightarrow \exists x, (W(x) \text{ and } \forall y, W(y) \Rightarrow y = x)$$

The statement $\exists! x, W(x)$ is read "There exists a unique x such that $W(x)$."

Rules of Inference for Unique Existence*

$\exists!+$	$\exists!-$
Show: $W(s)$ *Let y be arbitrary.* *Assume* $W(y)$ **Show:** $y = s$ ← ← **Conclude:** $\exists! x, W(x)$	**Show:** $\exists! x, W(x)$ **Conclude:** $\exists x, W(x)$ and $\forall y, W(y) \Rightarrow y = x$

Equality

Definition 35. The equality symbol, $=$, is defined by the following two rules of inference.

Rules of Inference for Equality

Reflexivity of $=$	Substitution*
Conclude: $x = x$	**Show:** $x = y$ **Show:** W **Conclude:** W with the *nth* free occurrence of x replaced by y.

**Restrictions and Remarks*

- Note that in the Reflexive rule there are no inputs, so you can insert a statement of the form $x = x$ into your proof at any time.
- No free variable in y can become bound when y is substituted for x.

Exercise 36. Prove the following with a formal proof.

$$(\exists! x, P(x)) \text{ and } (P(a) \text{ and } P(b)) \Rightarrow a = b$$

Precedence

While this is not universally agreed on in mathematics, in our formal system, quantifiers have a lower precedence than $\Leftrightarrow$. Thus they quantify the largest statement to their right possible unless specifically limited by parentheses. For instance,

$$\forall x, P(x) \Rightarrow Q$$

means the same thing as $\forall x, (P(x) \Rightarrow Q)$ but is different from $(\forall x, P(x)) \Rightarrow Q$.

Peano Axioms for the Naturals

It is possible to define the Natural Numbers and addition, multiplication, and $<$ for those numbers from scratch. One famous way of doing that is with the following axioms which were developed by Giuseppe Peano at the end of the 19^{th} century.

Axiom Name	Definition
N0	0 is a natural number
N1	$\forall n, \sigma(n)$ is a natural number
N2	$\forall n, \forall m, \sigma(n) = \sigma(m) \Rightarrow n = m$
N3	$\forall n, 0 \neq \sigma(n)$
N4	$P(0)$ and $(\forall k, P(k) \Rightarrow P(\sigma(k))) \Rightarrow \forall n, P(n)$
A0	$\forall n, n + 0 = n$
A1	$\forall n, \forall m, m + \sigma(n) = \sigma(m + n)$
M0	$\forall n, n \cdot 0 = 0$
M1	$\forall n, \forall m, m \cdot \sigma(n) = m + m \cdot n$
I	$\forall n, \forall m, m \leq n \Leftrightarrow \exists k, m + k = n$

In N0 the symbol 0 is a constant (i.e., N0 is a constant declaration). In all of the axioms the quantified variables have natural number type, so that in particular we can only apply the $\forall-$ rule for expressions which also are type natural number. In N4 above and in the following, $P(n)$ is a statement about a natural number variable n (i.e., P is a lambda expression that returns a statement when applied to a natural number variable n). Axiom N4 is called *mathematical induction*, or simply *induction*.

While not strictly necessary, the following definitions are useful.

Definition (**base ten representation**)**.** We define the usual base ten representations of natural numbers such that $1 = \sigma(0)$, $2 = \sigma(1)$, $3 = \sigma(2)$, $4 = \sigma(3)$,... and so on.

Definition (**less than**)**.** $\forall m, \forall n, m < n \Leftrightarrow m \leq n$ and $m \neq n$.

We can now prove basic theorems about arithmetic in the naturals. We will now use shortcuts freely, writing our proofs in a more informal manner, using the above axioms as rules of inference. For instance, axiom $N2$ can be interpreted as a rule of inference as follows:

Show: $\sigma(n) = \sigma(m)$

Conclude: $n = m$

We will also be using the facts that equality is reflexive, symmetric, and transitive freely, and write **transitive chains** of $=$ as we did for $\Leftrightarrow$ above. In particular, we will assume $=$ satisfies:

- **Reflexive property:** $\forall a, a = a$
- **Symmetric property:** $\forall a, \forall b, a = b \Rightarrow b = a$
- **Transitive property:** $\forall a, \forall b, \forall c, a = b$ and $b = c \Rightarrow a = c$.

We can now prove our first theorem about natural numbers.

Theorem 37. $1 + 1 = 2$.

Proof. Recall that 1 is defined to be $\sigma(0)$ and 2 is defined to be $\sigma(1)$. So, we have

$$\begin{aligned} 1 + 1 &= 1 + \sigma(0) && (1)\\ &= \sigma(1 + 0) && (2)\\ &= \sigma(1) && (3)\\ &= 2 && (4) \end{aligned}$$

where line (2) follows from Axiom A1 and line (3) follows from axiom A0. □

The most important axiom defining the natural numbers is **induction**, axiom N4. If we break this down as a rule of inference, it would say:

Show: $P(0)$
Show: $\forall k, P(k) \Rightarrow P(\sigma(k))$

Conclude: $\forall n, P(n)$

We can further break down the second step into several easier steps:

Show: $P(0)$
Let k be an arbitrary natural number.
Assume $P(k)$
Show: $P(\sigma(k))$
$\leftarrow$
$\leftarrow$

Conclude: $\forall n, P(n)$

Showing $P(0)$ is often called the **base case**, and the rest of the proof is called the **induction step**. The assumption $P(k)$ is often referred to as the **inductive hypothesis** (the textbook calls it the **hypothesis of induction**, which is the same thing).

We can now show that the "successor" function σ is the same as adding 1, using induction.

Theorem 38. *Prove that $\sigma(m) = 1 + m$ for all natural numbers m.*

Proof. We'll prove this by induction.

For the base case, note that $\sigma(0) = 1$ by the definition of 1, and $1 = 1+0$ by axiom A0. So, by transitivity of equality, $\sigma(0) = 1 + 0$. Thus the statement is true for $m = 0$.

For the induction step, let k be an arbitrary natural number. Assume $\sigma(k) = 1 + k$. Then

$$\begin{aligned} \sigma(\sigma(k)) &= \sigma(1 + k) && \text{(by substitution)} \\ &= 1 + \sigma(k) && \text{(by Axiom A1)} \end{aligned}$$

and so the statement holds for $\sigma(k)$. This completes the proof. □

We can also prove the following elementary properties of the naturals by induction:

8.5 Properties of the successor function

Theorem 39 (Nonzero implies successor). $\forall n, n \neq 0 \Rightarrow \exists m, n = \sigma(m)$

Theorem 40 (No number is its own successor). $\forall n, n \neq \sigma(n)$

8.6 Properties of addition

Theorem 41 (Alternate definition of σ). $\forall m, \sigma(m) = m + 1$

Theorem 42 (Associativity of addition). $\forall m, \forall n, \forall p, m + (n + p) = (m + n) + p$

Theorem 43 (Additive Identity). $\forall m, 0 + m = m = m + 0$

Lemma 44. $\forall m, 1 + m = m + 1$

Theorem 45 (Commutativity of addition). $\forall m, \forall n, m + n = n + m$

Theorem 46 (Zero sums). $\forall m, \forall n, m + n = 0 \Rightarrow m = n = 0$

Theorem 47 (Cancellation Law of addition). $\forall p, \forall m, \forall n, m + p = n + p \Rightarrow m = n$

8.7 Properties of multiplication

Lemma 48. $\forall m, 0 \cdot m = 0$

Theorem 49 (**Multiplicative identity**). $\forall m, 1 \cdot m = m = m \cdot 1$

Theorem 50 (**Left Distributive Law**). $\forall n, \forall m, \forall p, p \cdot (m + n) = p \cdot m + p \cdot n$

Theorem 51 (**Right Distributive Law**). $\forall p, \forall n, \forall m, (m + n) \cdot p = m \cdot p + n \cdot p$

Theorem 52 (**Commutativity of multiplication**). $\forall m, \forall n, m \cdot n = n \cdot m$

Theorem 53 (**Associativity of multiplication**). $\forall p, \forall m, \forall n, (m \cdot n) \cdot p = m \cdot (n \cdot p)$

Theorem 54 (**Zero divisors**). $\forall m, \forall n, m \neq 0$ and $m \cdot n = 0 \Rightarrow n = 0$

Theorem 55 (**Cancellation Law of multiplication**). $\forall p, p \neq 0 \Rightarrow \forall m, \forall n, p \cdot m = p \cdot n \Rightarrow m = n$

8.8 Properties of order

Theorem 56 (**Nonzero is positive**). $\forall n, 0 \leq n$ and $(0 < n \Leftrightarrow n \neq 0)$

Theorem 57 (**Successors are bigger**). $\forall n, n < \sigma(n)$.

Theorem 58 (**Alternate definition of** $<$). $\forall m, \forall n, m < n \Leftrightarrow \exists k, k \neq 0$ and $m + k = n$

You can try to prove all of these from the axioms in your spare time, but from now on we will be using them as facts/proof techniques themselves. Note that the Cancellation Laws above give us a notion of subtraction and division, when they are defined. We will simply say "by arithmetic" or "by Peano arithmetic" for our reasons whenever we wish to justify something that follows from these basic facts.

9 Functions and equality

The standard definition of function is usually stated something like this:

Definition 59. (Naive.) A **function** from a set A to a set B is a relation f from A to B having the property that

$$\forall x \in A, \exists! y \in B, (x, y) \in f.$$

We can then define things like the domain, codomain, range, and function notation:

Definition 60. The **domain** of a function from A to B is A, and the **codomain** is B. The **range** is $\{y \in B : \exists x \in A, (x, y) \in f\}$. Instead of writing $(x, y) \in f$ we write $f(x) = y$, since there is a unique such y for each x. We also write

$$f : A \to B.$$

Note that a function is not simply its set of ordered pairs - **it is a set of ordered pairs along with a specified choice of domain and codomain.** So, a more rigorous definition of function would go like this:

Definition 61. A **function** is an ordered triple of sets (A, B, f) where f is a relation from A to B satisfying the property that $\forall x \in A, \exists! y \in B, (x, y) \in f$.

We often simply write 'f' to denote the function when we really mean the entire triple (A, B, f) consisting of the domain, the codomain, and the mapping between them. In some sense the notation $f : A \to B$ is simply another way of writing this triple.

Because of the $\exists!$ statement, we can also think of the function f as a rule for assigning each element of A to a unique element of B. So an equivalent way of defining a function is:

Definition 62. A **function** is an ordered triple (A, B, f) where A and B are sets and f is a rule for assigning, to each element of A, a unique element of B.

Hence definitions of functions of the form

$$f : \mathbb{R} \to \mathbb{R} \text{ by } f(x) = x^2 + 1.$$

Such a definition specifies the domain, the codomain, and the rule.

Either of Definitions 61 or 62 are valid, fully rigorous definitions of functions.

Function equality

The textbook describes function equality as follows:

Definition 63. (WARNING: DO NOT USE) Two functions f and g are equal if they are equal as sets of ordered pairs.

This definition has many problems with it. For one, two functions f and g may be equal with respect to this definition but have different codomains, such as $f : \mathbb{R} \to \mathbb{R}$ by $f(x) = x^2$ and $g : \mathbb{R} \to \mathbb{R}_{\geq 0}$ by $g(x) = x^2$. The reason this is a problem is that g is onto (surjective) and f is not! So the two functions don't even have the same properties.

The textbook gets around this issue by saying that functions can't be surjective by themselves, they can only be surjective onto a specified set. But most mathematicians just specify that set up front to be the codomain, and use that to distinguish the two functions.

A better, and more widely used, definition of function equality is the following, which we will be using throughout the class.

Definition 64. Two functions f and g are equal if they have the same domain and codomain and are equal as sets of ordered pairs.

Alternatively, two functions are equal if they have the same domain and codomain and have the same rule. So, to prove two functions f and g are equal, one could use the following steps:

1. Show that f and g have the same domain A.
2. Show that f and g have the same codomain B.
3. Let $x \in A$ be arbitrary.
4. Prove that $f(x) = g(x)$.
5. Conclude that $f = g$.

10 Cardinality and Combinatorics

Combinatorics can be defined as the study of counting. Roughly speaking, to *count* a collection of items means to determine the number of things in that collection. But what does that mean, precisely? What are we doing mathematically when we count to 5?

One way to put counting on a rigorous foundation is via the notion of bijections between sets.

Definition 65. Two sets A and B have the same *cardinality* if and only if there exists a bijection $f : A \to B$.

Exercise 66. Prove that the relation "*same cardinality as*" is an equivalence relation.

The following theorem is extremely useful for the above exercise and for combinatorics in general:

Theorem 67. *A function is a bijection if and only if it has an inverse.*

Definition 68. The *cardinality* of a set A, written $|A|$ or $\#A$, is its equivalence class under the relation "*same cardinality as*". For any positive integer n, if A has the same cardinality as $\{1, 2, \ldots, n\}$, we say that A has cardinality n, or "has n elements".

Example 69. If $A = \{2, ☺, 5, -\pi\}$ then $|A| = 4$, since there is a bijection from $\{1, 2, 3, 4\}$ to A.

If a set has cardinality n for some positive integer n, we say it is a *finite* set. To *count* the elements in a finite set A is to determine the cardinality of A. There are plenty of infinite cardinalities as well (in fact, $\mathbb{N}$ and $\mathbb{R}$ have different cardinalities!) but we usually do not use the word "counting" in the infinite case.

Often, when counting we more loosely use the term "collection" rather than "set", and "object" or "thing" rather than "element", so that we can define combinatorial operations on collections that may be cumbersome to define in terms of set theory. For instance, the above notion of cardinality can be informally stated as follows.

Combinatorial interpretation of n: The number n is the size of a collection of n things.

In general a combinatorial interpretation of an expression is a set whose cardinality is given by that expression.

11 Counting basics

The usual notion of counting is to count by 1's: add 1 to a running total for each new element encountered. This can be expressed as the fact that adding one thing to a collection increases the number of elements by 1. The more general principle is this:

> **Addition Principle:** The sum $a + b$ counts the total number of things in a collection formed by adding a collection of b things to a collection of a things.

The set-theoretic operation of adding a set B to a set A is called the *disjoint union*, written $A \sqcup B$, defined as a union in which any elements in both A and B are treated as distinct copies. The addition principle says that $|A| + |B| = |A \sqcup B|$. More formally, we have:

Theorem 70. *Suppose $A \cap B = \emptyset$, $C \cap D = \emptyset$, and $|A| = |C|$ and $|B| = |D|$. Then $|A \cup B| = |C \cup D|$.*

We therefore can define addition on cardinalities. There is also a natural combinatorial definition of multiplication.

> **Multiplication Principle:** The product $a \cdot b$ counts the number of ways to choose one thing from a collection of a things and then choose another from a collection of b things.

We can similarly make this rigorous using cross products:

Theorem 71. *Suppose $|A| = |C|$ and $|B| = |D|$. Then $|A \times B| = |C \times D|$.*

So the product of two cardinalities is the cardinality of the cross product of any two of their representative sets.

Example 72. For finite numbers a and b, we always have that the sum of the cardinalities a and b is the cardinality of the number $a + b$, and likewise for the product. But for the countably infinite cardinal $\aleph_0$, we have $\aleph_0 + \aleph_0 = \aleph_0$ and $\aleph_0 \cdot \aleph_0 = \aleph_0$. Can you see why?

Note that we can now replace Peano addition and multiplication with cardinal addition and multiplication.

Example 73. In order to prove that $2 + 3 = 5$ using the combinatorial definitions of numbers and addition, we note that $\{1, 2\}$ has cardinality 2 and $\{3, 4, 5\}$ has cardinality 3. When added together, we get the set $\{1, 2, 3, 4, 5\}$, which has 5 elements.

Below are some other essential combinatorial definitions. These are the building blocks for many combinatorial expressions that represent cardinalities.

Expression	Combinatorial definition
n	The size of a collection of n things.
$a+b$	The total size of a collection formed by adding b things to a things.
$a-b$	The number of elements left after removing b things from a things.
$a \cdot b$	The number of ways to choose one thing from a things and then one thing from b things.
a/b	If a things can be sorted into collections with b things in each collection, then a/b is the number of collections.
$(n)_k$	The number of ordered lists of k distinct things chosen from n things.
$(n)^k$	The number of ordered lists of k not-necessarily-distinct things chosen from n things.
$\binom{n}{k}$	The number of ways to choose k distinct things from n things where order doesn't matter.
$\left(\binom{n}{k}\right)$	The number of ways to choose k not-necessarily-distinct things from n things where order doesn't matter.
$n!$	The number of ways to permute (rearrange) n distinct things in a row.

We can now count more complicated objects by substituting numbers or other expressions into the formulas above.

Exercise 74. Find a combinatorial expression that describes the number of ways of picking out one shirt and one pair of pants from a wardrobe that has 10 shirts and 4 pairs of pants.

Exercise 75. Find a formula for the number of ways to choose a set of t different k-element subsets of $\{1, 2, \ldots, n\}$.

We now need ways of simplifying these expressions. In particular, even though the answer to Exercise 74 is clearly $10 \cdot 4$, we don't yet know an easy combinatorial way to show that $10 \cdot 4 = 40$.

The formulas below can be proven combinatorially using the tools described in the next two sections.

- $a \cdot b = b + b + \cdots + b$, the sum of a copies of b
- $n! = n \cdot (n-1) \cdot (n-2) \cdot \cdots \cdot 3 \cdot 2 \cdot 1$
- $n^k = n \cdot n \cdot \cdots \cdot n$, the product of k copies of n
- $(n)_k = n \cdot (n-1) \cdot (n-2) \cdot \cdots \cdot (n-k+1) = \frac{n!}{(n-k)!}$
- $\binom{n}{k} = \frac{n!}{k!(n-k)!}$
- $\left(\binom{n}{k}\right) = \binom{n+k-1}{k}$

12 Counting in two ways

We can prove some of the formulas listed above using the principle called *counting in two ways*.

Counting in Two Ways: If A has n elements and it also has m elements then $n = m$.

This sounds like a trivial principle, but it is incredibly useful in proving algebraic identities combinatorially, such as the first formula listed above.

Example 76. Suppose we wish to show that for any a and b, we have

$$a \cdot b = \underbrace{b + b + \cdots + b}_{a \text{ summands}}$$

where the sum has a copies of b. By repeated application of the Addition Principle, we see that the right-hand side, $b+b+\cdots+b$, counts the number of students in a school that has a classes with b students each. By the Multiplication Principle, $a \cdot b$ counts the number of ways to first walk into one of the a classrooms and then single out one of the b students in that class. Thus $a \cdot b$ is also equal to the total number of students, and so $a \cdot b = b + b + \cdots + b$.

Exercise 77. Prove, by counting in two ways, that the Peano axioms of addition and multiplication are satisfied by the combinatorial definitions of addition and multiplication. Prove also that addition and multiplication are associative, commutative, and satisfy the distributive law.

Example 78. To prove that

$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k},$$

note that $\binom{n+1}{k}$ counts the number of ways of choosing k people from a classroom that has n students and 1 teacher. There are $\binom{n}{k}$ such choices that *do not* contain the teacher, and $\binom{n}{k-1}$ that *do* contain the teacher (since we are choosing only $k-1$ of the students). The result follows.

13 Bijections: Changing the interpretation

Sometimes finding one set to count in two ways is not enough - we might have to start with a combinatorial interpretation of the left-hand side and then use a bijection to transform it into something counted by the right-hand side of the equation.

> **Bijective Proofs:** If A has m elements, B has n elements, and there is a bijection $f : A \to B$, then $m = n$.

Again, this sounds like a very simple principle, but it can be tricky to execute in practice.

Example 79. Suppose we wish to show that

$$\left(\binom{n}{k}\right) = \binom{n+k-1}{k}.$$

We can interpret the left hand side as the number of ways to fill a cup with k M&M's, where there are n different possible colors of M&M's. The right hand side counts the number of sequences consisting of k zeroes and $n-1$ ones, since such a sequence has $n+k-1$ entries total and is determined by choosing which k positions contain the 0's. It suffices to construct a bijection between these two sets.

To do so, given a cup of colored M&M's, dump them out onto your plate and sort them by color, in a row, in a specified ordering of the colors. Now there are k M&M's divided into n (possibly empty) sections. If we put a toothpick between each section, then we need $n-1$ toothpicks total. Finally, put on color-filter goggles so that all the M&M's look grey, and we end up with a row of k grey M&M's and $n-1$ toothpicks. Represent each grey M&M by a 0 and each toothpick by a 1, and we obtain a sequence of k zeroes and $n-1$ ones.

Conversely, starting from a row of 0's and 1's, we can interpret this as a row of k grey M&M's separated by $n-1$ toothpicks, and then color the M&M's according to the specified ordering of colors and put them into a cup. Since these processes reverse each other, the two sets are in bijection, as desired.

Exercise 80. Write the above proof in a more rigorous fashion using sets and bijections.

14 Cycle notation for permutations

Recall that a **permutation** of a set A is a bijection from A to itself. If $A = \{1, 2, \ldots, n\}$, we can write any permutation in **cycle notation** due to the following theorem.

Theorem 81. *Let $A = \{1, 2, \ldots, n\}$ and let $f : A \to A$ be a bijection. Then there is a partition $\mathcal{P}$ of A into sets $A_1, \ldots, A_k$ for some k, such that each A_i has an ordering of its elements, say $a_{i,1}, \ldots, a_{i,m_i}$, for which $f(a_{i,j}) = a_{i,j+1}$ for all $j < m_i$ and $f(a_{i,m_i}) = a_{i,1}$. These A_i's are called the **cycles** of f, and the permutation f is written in cycle notation as*

$$(a_{1,1}a_{1,2}\cdots a_{1,m_1})(a_{2,1}a_{2,2}\cdots a_{2,m_2})\cdots(a_{k,1}a_{k,2}\cdots a_{k,m_k}).$$

Proof. We proceed by strong induction on n. For the base case, $n = 1$, there is only one bijection $f : A \to A$, namely the identity map in which $f(1) = 1$ is a single cycle, or in cycle notation, (1).

Now let $k \geq 1$ be arbitrary and assume the claim is true for all $j \leq k$. Let f be a bijection from $\{1, \ldots, k\}$ to itself. Consider the sequence

$$S = 1, f(1), f(f(1)), f(f(f(1))), \ldots.$$

Since this sequence is infinite and the set is finite, by the Pigeonhole Principle some two elements of the sequence must be equal to each other. So $f^k(1) = f^j(1)$ for some k and j. But since f is invertible, we can apply f^{-k} or f^{-j} to both sides to find $f^{|k-j|}(1) = 1$. In other words, eventually the sequence comes back to 1 and cycles.

Let A_1 be the set of numbers that appear in S. Then if we restrict f to $\{1, \ldots, k\} - A_1$, its restriction is still a bijection, because all the elements of A_1 map into A_1. So by the inductive hypothesis on this smaller set, we can partition the remaining elements into cycles as well, forming a partition into cycles as desired. □

For instance, the permutation (351)(26)(4) is the function $f : \{1, 2, \ldots, 6\}$ for which $f(1) = 3$, $f(2) = 6$, $f(3) = 5$, $f(4) = 4$, $f(5) = 1$, and $f(6) = 2$.

Remark. Often we drop singleton cycles from our notation. So (351)(26)(4) can be simply written as (351)(26). In other words, if a number doesn't appear in a cycle it is assumed to be fixed by f.

In addition, there is more than one way to write any given permutation in cycle notation. The above permutation could also be written as (513)(62) or as (26)(135).

14.1 Composing permutations

Permutation composition is simply a special case of function composition, but we can do it easily in cycle notation as follows. To compute (351)(26) ∘ (1324), we start with any number from 1 to 6, say 1, and open a cycle with it:

$$(1\cdots$$

Then we plug in 1 to the rightmost cycle, and at each step plug the output into the previous cycle. So $1 \to 3$ in the last cycle, and $3 \to 3$ in (26), and finally $3 \to 5$ in (351). So $1 \to 5$ overall, and we continue with a 5:

$$(15\cdots$$

We now do the same computation starting with 5 and we see $5 \to 1$, so we close the cycle:

$$(15)$$

and now we open another cycle with a number we haven't used yet, say 2 in this case, and start the process over again. Eventually we find:

$$(15)(2436)$$

and we're done since there are no other integers appearing in any cycle.

Notice that we didn't need to write the ∘, since we are treating each cycle as its own permutation and composing them all at once. We can therefore compose any number of cycles together in this manner.