MATH 240 Fall 2024

notes by

Marcel Goh

A note on these notes. After each class, this document will be updated with the new material that was just covered. The datestamps in the left margin indicate when the notes from each day start. Subsections labelled with a $*$ are optional. This document is heavily based on notes by Jeremy Macdonald, but any errors are likely my own. Please email me if you find any.

1. Set theory

29.VIII A *set* is a collection of distinct objects, called its *elements* or its *members*. If x is a member of set A, then we write $x \in A$, and if x is not an element of A, then we write $x \notin A$. Sets can be written by listing out its elements. For example,

$$\left\{1, 4, 7, 10, \sqrt{782}\right\}$$

and

$$\left\{\{1, 2\}, \pi, \{4\}\right\}$$

are both sets (the second example shows that sets can themselves contain other sets). The order of the elements is not important, and duplicate elements are ignored (so $\{1, 2\} = \{2, 1\} = \{1, 1, 2\}$). The notation using { and } is useful for defining small, concrete examples, but expressing large sets can become very cumbersome. The first way one can describe larger sets is to use the ... symbol and the power of suggestion. For instance, anyone faced with the notation

$$A = \{1, 3, 5, 7, 9, \ldots\}$$

can quickly guess that this set is supposed to contain all the positive odd integers. We can also use ... to define finite sets. Most Canadians will be able to tell you that the set

$$\{\text{Alberta}, \text{British Columbia}, \ldots, \text{Yukon}\}$$

of provinces and territories contains 13 elements. But this notation inherently produces some ambiguity. For example, since the sequence of positive palindromic binary numbers starts $1, 3, 5, 7, 9, 15, 17, 21, 27, \ldots$, we are left with some doubt as to whether the set A above should be the set of odd numbers or the set of palindromic binary numbers.

But there is another, less ambiguous way to define large sets. It is called set-builder notation and it refers to any construction of the form

$$\{x \in U : P(x)\},$$

where x is a variable, U is a set, and P is a statement about x. The resulting set contains *all x such that $P(x)$ holds*. For example, letting $\mathbf{N}$ denote the set $\{0, 1, 2, 3, \ldots\}$ of counting numbers (more on this later), to define the set of all odd numbers, we can write

$$A = \{x \in \mathbf{N} : \text{there exists } k \in \mathbf{N} \text{ such that } x = 2k + 1\}.$$

Note that the statement $P(x)$ must contain x, but it may also contain other previously defined symbols, as well as new symbols defined within the statement (such as k in the example above).

Special sets of numbers. There are certain infinite sets of numbers that are used so often as to be given special bold notation. Back in elementary school, you learned to use the counting numbers $0, 1, 2, 3, \ldots$. We already saw this set in the previous paragraph; in the business, this set is known as the *natural numbers*, because if you go on a nature hike you can use them to count the number of bluebells, donkeys, etc. that you see. (Many mathematicians use the symbol $\mathbf{N}$ to denote this set without zero. When in doubt, clarify with the person you're talking to; in this class, $0 \in \mathbf{N}$.)

Sometime towards the start of junior high you were introduced to the concept of natural numbers. The set

$$\mathbf{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$$

is called the set of *integers* or *whole numbers*. (The word *integer* just means "whole" in Latin; cf. French *entier*. We use the letter $\mathbf{Z}$ because of the German word *Zahl* meaning "number.")

Even before you learned about negative numbers, you probably learned about fractions. They can be defined in set-builder notation as a collection of ratios of integers, where the denominator is not zero:

$$\mathbf{Q} = \{p/q : p, q \in \mathbf{Z}, q \neq 0\}.$$

(The nitpicky reader will notice that p/q is not stipulated to be a member of any set here. This is because a rigorous definition of $\mathbf{Q}$ involves quotienting out by an equivalence relation, which we don't know how to do yet.) This is the set of *rational numbers*. Remember that sets only contain distinct elements, so $2/3$, $4/6$, $6/9$, etc. are all considered the same rational number.

Lastly, we have the set $\mathbf{R}$ of real numbers. Constructing this set using only notions from set theory and logic is quite the byzantine task and well outside the scope of this course, but you can think of $\mathbf{R}$ as the set of decimal numbers with

a finite number of digits to the left of the decimal point, and a possibly infinite number of digits to the right of the decimal point.

A set of numbers near and dear to many mathematicians' hearts is the set of *prime* numbers P, defined by

$$P = \big\{p \in \mathbf{N} : p \geq 2, \text{ and if } p = ab \text{ then } \{a, b\} = \{1, p\}\big\}.$$

(You may have met a different definition of prime numbers in the past. Pause a moment and convince yourself that the statement above defines the set of prime numbers as you know it.)

Set inclusion. When we use set builder notation $B = \big\{x \in A : P(x)\big\}$, every element of B is necessarily a member of the set A as well, since B is defined to be the set of all x in A satisfying $P(x)$. This is one way in which we can obtain a *subset* of another set. More generally, we write $B \subseteq A$ if every element of B is also an element of A, and $B \supseteq A$ if every element of A is an element of B. Sometimes, if $B \supseteq A$, we say that B *contains* A, or B *includes* A. For example, we have the chain

$$\mathbf{N} \subseteq \mathbf{Z} \subseteq \mathbf{Q} \subseteq \mathbf{R}$$

for the special sets of numbers defined earlier.

Symbols like $\subseteq$, $\supseteq$, and $=$ (that are used to produce statements) can be negated with a slash; for example, if there is some element of B that is not an element of A, then we write $B \not\subseteq A$.

The concept of set inclusion is important, because the most common way to prove that two sets A and B are equal is to show that A is a subset of B, then show that B is a subset of A. We illustrate this with the following example.

Proposition 1. *The sets*

$$A = \{x \in \mathbf{Z} : \textit{there exists } k \in \mathbf{Z} \textit{ such that } x = 2k + 1\}$$

and

$$B = \{x \in \mathbf{Z} : \textit{there exists } l \in \mathbf{Z} \textit{ such that } x = 2l + 5\}$$

are equal. (Both are different ways of expressing the set of all odd integers.)

Proof. Let $x \in A$. Then there exists $k \in \mathbf{Z}$ such that $x = 2k + 1$. Letting $l = k - 2$, we find that l is an integer (since k was). Furthermore,

$$x = 2k + 1 = 2k - 4 + 5 = 2(k - 2) + 5 = 2l + 5.$$

We have found l such that $x = 2l + 5$, so $x \in B$. This shows that $A \subseteq B$.

On the other hand, let $x \in B$, so that there exists $l \in \mathbf{Z}$ such that $x = 2l+5$. Now we let $k = l + 2$; $k \in \mathbf{Z}$ since $l \in \mathbf{Z}$. We have

$$x = 2l + 5 = 2l + 4 + 1 = 2(l + 2) + 1 = 2k + 1.$$

This shows that $x \in A$, and we have proved that $B \subseteq A$. This combined with the previous paragraph shows that $A = B$. ∎

The above result is not so important, but pay attention to the structure of this proof. It is called a "proof by double inclusion," since we have shown that A includes B and B includes A.

Set operations. Now we describe a number of operations that may be performed on sets to produce other sets. They can all be built up from the following two operations.

- The *union* $A \cup B$ of two sets A and B is the set of all elements that are either in A or in B (or both).
- The *intersection* $A \cap B$ of A and B is the set of all elements that are in both A and B.

As an example, if $A = \{1, 2, 4\}$ and $B = \{1, 3, 5\}$, then $A \cup B = \{1, 2, 3, 4, 5\}$ and $A \cap B = \{1\}$.

To avoid logical difficulties, we always assume that the sets we're working with are a subset of some larger ambient set U, often called the *universe*. Once we know what U is, we may define the *complement* of a set A to be the set $\overline{A}$ of all the elements in U except those that are in A. So if $U = \{1, 2, 3, 4, 5\}$ in the example above, then $\overline{A} = \{3, 5\}$ and $\overline{B} = \{2, 4\}$. What about $\overline{A \cup B}$? Well since $A \cup B$ is all of U, its complement must be empty, and we can denote it $\{\}$. This is one valid notation for the *empty set*. The other is $\emptyset$.

Now is a good time to define the cardinality $|A|$ of a set A. This is the number of elements in it, so $|A| = |B| = 3$ in our example, and $|A \cup B| = 5$, etc. We have $|\emptyset| = 0$, and it is possible for the cardinality of a set to be infinity; for example, $|\mathbf{N}| = \infty$. We also have $|\mathbf{R}| = \infty$, but this infinity is, in some sense, larger than $|\mathbf{N}|$. (More on that later.)

Next, we define the *difference* $B \setminus A$ (sometimes $B - A$) of two sets. This is the set of all elements in B that are *not* in A. So, using the complement notation we just learned about, we can express $B \setminus A = B \cap \overline{A}$. It is not necessary that A be a subset of B. In the small example above, we have $A \setminus B = \{2, 4\}$ and $B \setminus A = \{3, 5\}$.

Lastly, we define the *symmetric difference* $A \triangle B$ of two sets A and B to be the set of all elements that are either in A or in B *but not both*. Invoking the above example one last time, we have $A \triangle B = \{2, 3, 4, 5\}$. We can express it as using unions, intersections, and complements as

$$A \triangle B = (A \cup B) \cap \overline{A \cap B}. \tag{1}$$

To practise using all the different operations we just learned, convince yourself that the following are three more valid ways to express the symmetric difference:

$$A \triangle B = (A \cup B) \setminus (A \cap B) = (A \setminus B) \cup (B \setminus A) = \overline{\overline{A \cup B} \cup (A \cap B)}$$

More set identities abound. We state the following proposition without proof; you should try going though this list and convincing yourself that each identity holds, for all sets A, B, and C. (This is a great way of practising proofs by double inclusion.)

Proposition 2. *Let A, B, and C be subsets of a universe U. Then*

i) $A \cap U = A$ *and* $A \cup \emptyset = A$;
ii) $A \cup U = U$ *and* $A \cap \emptyset = \emptyset$;
iii) $A \cup A = A$ *and* $A \cap A = A$;
iv) $\overline{\overline{A}} = A$;
v) $A \cup B = B \cup A$ *and* $A \cap B = B \cap A$;
vi) $A \cup (B \cup C) = (A \cup B) \cup C$ *and* $A \cap (B \cap C) = (A \cap B) \cap C$);
vii) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ *and* $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$;
viii) $A \cup (A \cap B) = A$ *and* $A \cap (A \cup B) = A$; *and*
ix) $A \cup \overline{A} = U$ *and* $A \cap \overline{A} = \emptyset$. ∎

These laws have names, some of which we will use more often than others: (i) is called the identity law, (ii) the domination law, (iii) the idempotent law, (iv) the law of double negation, (v) the commutative law, (vi) the associative law, (vii) the distributive law, (viii) the absorption law, and (ix) the complement law.

***Analogy with addition and multiplication.** Some of these laws bear some resemblance to laws about numbers that you already know. As an exercise, replace $\cup$ with $+$ (addition), $\cap$ with $\cdot$ (multiplication), $\overline{A}$ with $-A$ (negation), U with 1, and $\emptyset$ with 0 in all the formulas above, and now assume that A, B, and C are arbitrary real numbers. Which identities still hold in the number setting, and which ones don't? As a more advanced exercise, try replacing $\cup$ with $\triangle$ in the identities above (some statements will have to be tweaked a bit so that they're actually true, some won't). Now do the same replacement as before, except replace $\triangle$ with $+$. You will find that many more identities carry over.

03.IX **De Morgan's laws.** There are two important laws relating complements with union and intersection. We shall state them as a proposition, this time giving a proof (of one of them).

Proposition 3 (*De Morgan's laws*). *Let A and B be sets. Then*

i) $\overline{A \cup B} = \overline{A} \cap \overline{B}$; *and*
ii) $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

Proof. Let $x \in \overline{A \cup B}$. This means that x does not belong to the union of A and B, x cannot be in A, nor can it be in B. Since $x \notin A$, $x \in \overline{A}$, and since $x \notin B$, $x \in \overline{B}$. Therefore, $x \in \overline{A} \cap \overline{B}$. This shows that $\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$.

Now assume that $x \in \overline{A} \cap \overline{B}$. So $x \in \overline{A}$ and $x \in \overline{B}$, meaning that $x \notin A$ and $x \notin B$. Since x is in neither A nor B, it is also not a member of the union $A \cup B$.

We conclude that $x \in \overline{A \cup B}$. We have shown that $\overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$, which fact, combined with the previous paragraph, completes the proof of (i).

The proof of (ii) is similar and left to the reader as an exercise. ∎

Armed with all of these laws, we are able to perform lots of mechanical set manipulations to simplify expressions. For example, consider the expression

$$\big((A \setminus B) \cup A\big) \cap \overline{A \cap \overline{B}}.$$

Since $A \setminus B = A \cap \overline{B}$ and invoking the second De Morgan law on the right of the intersection yields

$$\big((A \cap \overline{B}) \cup A\big) \cap (\overline{A} \cup B).$$

Now, we can use absorption on the left-hand side to obtain

$$A \cap (\overline{A} \cup B),$$

and then distributing gives us

$$(A \cap \overline{A}) \cup (A \cap B) = \emptyset \cup (A \cap B) = A \cap B.$$

We thus see that the nasty expression $\big((A \setminus B) \cup A\big) \cap \overline{A \cap \overline{B}}$ is simply another way of writing $A \cap B$.

The Cartesian product and power set. From the real line $\mathbf{R}$, we can geometrically construct the Cartesian plane $\mathbf{R}^2$ by lining up parallel copies of $\mathbf{R}$, one for each element of the original line and all parallel to the original line. Notationally, $\mathbf{R}^2$ is the set of all ordered pairs (a, b), where $a, b \in \mathbf{R}$. Generalising this, for any sets A and B we can define the *Cartesian product* $A \times B$ to be the set

$$A \times B = \big\{(a, b) : a \in A, b \in B\big\}.$$

We sometimes write A^2 for $A \times A$, and more generally A^n for the n-fold Cartesian product of A with itself. (This explains the notation $\mathbf{R}^2$ for the Cartesian plane, and $\mathbf{R}^n$ for the n-dimensional vector space over $\mathbf{R}$.) Note that $A \times B$ is not equal to $B \times A$ in general.

Proposition 4. *If A and B are finite sets, then $|A \times B| = |A| \cdot |B|$.*

Proof. The set $A \times B$ consists of all ordered pairs (a, b) where $a \in A$ and $b \in B$. There are $|A|$ choices for a, and for a, there are $|B|$ ways to pair it with a b from B. So there are $|A| \cdot |B|$ pairs in total. ∎

Now we define the *power set*. For a set A, this is the set of all subsets of A, and is commonly denoted by $\mathcal{P}(A)$ or 2^A. (We will use the latter notation in these notes.) In set-builder notation, we have

$$2^A = \{X \subseteq U : X \subseteq A\}.$$

As an example, if $A = \{1, 2, 3\}$, then

$$2^A = \big\{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\big\}.$$

Note that even though, say, $1 \in A$, we do not have $1 \in 2^A$. We do, however, have $\{1\} \in 2^A$.

Another example is $2^{\mathbf{Z}}$, the set of all subsets of integers. If P is the set of primes, then $P \in 2^{\mathbf{Z}}$. Also in $2^{\mathbf{Z}}$ is the set $S = \{n^2 : n \in \mathbf{Z}\}$ of square numbers.

There is a way of encoding subsets with strings of 0s and 1s. Suppose we have a set

$$\{-3, 1, 7, 19, 23\}.$$

Fixing this order of the elements, for any arbitrary subset of this set, we can associate to it a binary string. Consider the subset $\{1, 19, 23\}$. This corresponds to the binary string 01011: since the first element, 3, is not in the subset, we write a 0. Then 1 is in the subset, so we write a 1, and so on. This is a reversible process. Given a binary string of length 5, say, 00101, we can reconstruct the subset that corresponds to it. The first two 0s mean that -3 and 1 do not belong to the set, but 7 does, 19 doesn't, and 23 does. So the subset is $\{7, 23\}$. In this way we see that there is a one-to-one correspondence between the elements of 2^A and binary strings of length $|A|$. We'll use this fact in the proof of the following proposition.

Proposition 5. *If A is finite, then $|2^A| = 2^{|A|}$.*

Proof. We just saw that there is a one-to-one correspondence between elements of 2^A and binary strings of length $|A|$. So it suffices to count the number of binary strings of length $|A|$. Well, each digit can be either 0 or 1, and there are $|A|$ digits, so the number of strings is

$$\underbrace{2 \cdot 2 \cdot \cdots \cdot 2}_{|A| \text{ times}} = 2^{|A|}. \quad \blacksquare$$

Counterexamples in proofs. We finish this subsection with a little example problem. *Is it true that $2^A \cup 2^B = 2^{A \cup B}$ for all sets A and B?*

Let's start by trying to prove the statement is true. As usual, we will attempt a double-inclusion proof. Let $2^A \cup 2^B$. This means X is either a subset of A or it is a subset of B. Either way, X is a subset of $A \cup B$, so $X \in 2^{A \cup B}$. So far so good; we have proved that $2^A \cup 2^B \subseteq 2^{A \cup B}$.

Now we try the other direction. Let $X \in 2^{A \cup B}$. So X is a subset of $A \cup B$. From here we want to say that X must be a subset of A or it must be a subset of B, but is that necessarily true? It is possible that X is contained slightly in A and slightly in B. So we have failed to prove that $2^{A \cup B} \subseteq 2^A \cup 2^B$ in general. But just because we have failed to prove that something is true doesn't mean we have proved it is false!

To actually prove that $2^{A \cup B} \subseteq 2^A \cup 2^B$ doesn't hold in general, we need to find a *counterexample*. That is, we need to construct sets A and B such that

the statement is false. In this case, we can let $A = \{1, 2\}$, $B = \{3, 4\}$, so that $A \cup B = \{1, 2, 3, 4\}$. Then the set $\{1, 3\}$ is a subset of $A \cup B$ but is not a subset of A and it is not a subset of B. In other words, $\{1, 3\} \in 2^{A \cup B}$ but $\{1, 3\} \notin 2^A \cup 2^B$, proving that $2^{A \cup B} \not\subseteq 2^A \cup 2^B$.

05.IX **Russell's paradox.** Earlier, we said that the sets we are working with need to be a subset of a universe U, which has already been proven to be a set. We gave lots of ways to make sets out of new sets, such as the union and intersection operations, etc. Starting with the assumption that the empty set $\emptyset$ is a set, it is possible to define the set of natural numbers as follows. We can define $0 = \emptyset$, $1 = \{0\}$, and $2 = \{0, 1\}$, and so on. Now we take the set of all of these, and call this $\mathbf{N}$. (We can also define addition and multiplication on these set-theoretic "numbers" so that they behave like addition and multiplication do on $\mathbf{N}$.) From here we can do more funky stuff to define $\mathbf{Z}$, $\mathbf{Q}$, and $\mathbf{R}$, and prove that these are all sets (you can see this in a higher-level course on set theory, if you're interested). So there isn't much of a problem with all the sets we have played with so far; they are all subsets of things that are already known to be sets.

Ungodly things can happen if we don't stick by these rules. An example, due to Bertrand Russell, is the "set"

$$R = \{\text{sets } X : X \notin X\}.$$

In plain English, R is defined to be the set of all sets that contain themselves. This is not a subset of any known thing, so by our criterion above we would not consider it a set. But supposing it is, let us ask ourselves the following question. Does R contain itself? If it does not, then $R \notin R$, so R would be a set that satisfied the condition of R, so $R \in R$. But on the other hand, if $R \in R$, then R violates the condition defining R, so $R \notin R$. Round and round we go in a circle of contradiction.

Such is the price of meddling with "sets" that aren't subsets of known sets. This also shows that there is no such thing as "the set of all sets."

2. Propositional logic

A *proposition* is a statement that is true or false. For example "8 is even" is a statement we know to be true, and "8 is prime" is a statement we know to be false. The statement "n is prime" is not a proposition because its truth or falsity depends on what n is. The statement "$2^{2^{240}} - 1$ is prime" is a proposition, because it is either true or false (even though you or I might not know which one it is).

A *propositional variable* or a *boolean variable* is a variable which can take either the value 0 or 1, where 0 means "false" and 1 means "true." Usually we use letters p, q, and r to denote propositional variables. The simplest logical operator is negation, defined by the table

p	$\neg p$
0	1
1	0

.

This is also called the NOT operator, since if p is true, then $\neg p$ is false, and vice versa. Next is *conjunction*, which has the table

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

.

This is also called the AND operator, because $p \wedge q$ is true if and only if p and q are both true. The OR operator, also called *disjunction*, has the table

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

.

We see that $p \vee q$ is true if p or q is true (or both). The symbol $\vee$ is meant to recall the Latin word *vel*, meaning "or." (One of the most important early treatises on mathematical logic and set theory was *Arithmetices principia, nova methodo exposita*, published in Latin in 1889 by Giuseppe Peano. It established the now-standard axiomatisation of the natural numbers.)

On the other hand, in English, we often use the word "or" to mean an *exclusive* or; that is, either p and q are true but not both. In mathematics, on the other hand, "or" is usually *inclusive*, so both p and q are allowed to hold at the same time. It is possible to express the exclusive disjunction (often called XOR) by a table, however. We will use the symbol $\oplus$ for this operator, and its table looks like this:

p	q	$p \oplus q$
0	0	0
0	1	1
1	0	1
1	1	0

So $p \oplus q$ is true if p is true or q is true, but not both.

A *formula* is an expression containing propositional variables, 0, 1, logical operators, and parentheses. The formula must syntactically make sense; for instance, $0(\vee \wedge q\neg$ is not a formula. Just as in ordinary mathematical notation, parentheses are used to clarify which operators should be evaluated first. We will assume that negation applies first, but an expression such as $p \vee q \wedge r$ is ambiguous. (In many programming languages, conjunction has higher priority than disjunction, but in this class, just add parentheses to clarify.)

Above we have illustrated the basic logical operators by writing out their *truth tables*. These are tables that give the value of a formula for all possible

values of its variables. We can write truth tables for more complex formulas as well:

p	q	$p \wedge q$	$\neg(p \wedge q)$	$\neg(p \wedge q) \oplus q$
0	0	0	1	1
0	1	0	1	0
1	0	0	1	1
1	1	1	0	1

Strictly speaking, the third and fourth columns are not necessary, but these intermediary columns help us verify the accuracy of the following ones.

Two formulas f_1 and f_2 are said to be *logically equivalent* if they have the same truth table; that is, they produce the same output if given the same input. In this case we write $f_1 \equiv f_2$. For example, let $f_1 = \neg(p \wedge q) \oplus q$, the formula whose truth table is illustrated above. Now let $f_2 = p \vee \neg q$. Its truth table is

p	q	$\neg q$	$p \vee \neg q$
0	0	1	1
0	1	0	0
1	0	1	1
1	1	0	1

,

so we conclude that $f_1 \equiv f_2$. As a larger example, suppose we want to find all values of p, q, and r such that

$$f = (p \vee q) \wedge (\neg q \vee \neg r)$$

evaluates to 1. The truth table

p	q	r	$p \vee q$	$\neg q \vee \neg r$	f
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	1	1	1
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	0	0

shows that f evaluates to 1 precisely when

$$(p, q, r) \in \big\{(0, 1, 0), (1, 0, 0), (1, 0, 1), (1, 1, 0)\big\}.$$

To write out the truth table for a formula with n variables, we need 2^n rows, so this method is unsuitable for formulas with more than three or four variables.

Simplifying logical formulas. Just as we have rules for simplifying set expressions, there are ways to turn complicated logical formulas into simpler ones that are logically equivalent. What might surprise you is that the rules turn out to be exactly the same! To see the basis for this correspondence, consider the definition of a union of sets A and B. In set-builder notation, this is

$$A \cup B = \{x \in U : x \in A \text{ or } x \in B\}.$$

The "or" in the definition suggests that $\cup$ is intimately related to the $\vee$ operation in propositional logic. Repeating this process, we have the "dictionary"

Set theory	Propositional logic
sets A, B	variables p, q
unions $A \cup B$	disjunctions $p \vee q$
intersections $A \cap B$	conjunctions $p \wedge q$
complements $\overline{A}$	negations $\neg p$
symmetric differences $A \triangle B$	exclusive disjunctions $p \oplus q$
the empty set $\emptyset$	0
the universe U	1

Exploiting this connection, we have the following analogue of Proposition 2.

Proposition 2. *Let p, q, and r be propositional variables. Then*

i) $p \wedge 1 \equiv p$ *and* $p \vee 0 \equiv p$*;*
ii) $p \vee 1 \equiv 1$ *and* $p \wedge 0 \equiv 0$*;*
iii) $p \vee p \equiv p$ *and* $p \wedge p \equiv p$*;*
iv) $\neg\neg p \equiv p$*;*
v) $p \vee q \equiv q \vee p$ *and* $p \wedge q \equiv q \wedge p$*;*
vi) $p \vee (q \vee r) \equiv (p \vee q) \vee r$ *and* $p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$*);*
vii) $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ *and* $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$*);*
viii) $p \vee (p \wedge q) \equiv p$ *and* $p \wedge (p \vee q) \equiv p$*; and*
ix) $p \vee \overline{p} \equiv 1$ *and* $p \wedge \overline{p} \equiv 0$. ∎

Since they are essentially the same as their set equivalents, the names of these laws are the same as in the realm of sets. We also have the propositional equivalent of De Morgan's law, which states that

x) $\neg(p \vee q) \equiv \neg p \wedge \neg q$ *and* $\neg(p \wedge q) \equiv \neg p \vee \neg q$.

Using these rules, we can now show that $\neg(p \wedge q) \oplus q \equiv p \vee \neg q$, which we already saw earlier from their truth tables. First we observe that

$$p \oplus q \equiv (p \vee q) \wedge \neg(p \wedge q), \tag{2}$$

which is analogous to the identity (1) for symmetric differences. So

$$\begin{aligned}\neg(p \wedge q) \oplus q &\equiv \big(\neg(p \wedge q) \vee q\big) \wedge \neg\big(\neg(p \wedge q) \wedge q\big) \\ &\equiv (\neg p \vee \neg q \vee q) \wedge \big((p \wedge q) \vee \neg q\big) \\ &\equiv (\neg p \vee 1) \wedge \big((p \vee \neg q) \wedge (q \vee \neg q)\big) \\ &\equiv 1 \wedge \big((p \vee \neg q) \wedge 1\big) \\ &\equiv p \vee \neg q,\end{aligned}$$

where in the first line we use (2), in the second line we use De Morgan's law twice, in the third line we use the complement and distributive laws, the fourth line we use the domination and complement laws, and in the last line we use the identity law (twice). (It is not super important to remember the names of all these laws, as long as you remember their statements, but you may actually find it is easier to remember the names along with the statements, just as it might be easier to remember faces of people you've met if you also know their names.)

10.IX **Conditional and biconditional.** We now examine the *conditional* logical relation IF p THEN q. It is denoted by $p \Rightarrow q$ and its truth table is given by

p	q	$p \Rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

.

Within the conditional statement, p is called the *antecedent*; this is the assumption. The statement that is asserted, conditional that the antecendent holds, is called the *consequent* q. You can quickly check that the relation $p \Rightarrow q$ is equivalent to $\neg p \vee q$. This fact is useful when performing mechanical simplifications. In English, the statement "if p then q" asserts a causal relation between p and q. Take a moment and reconcile this idea with the truth table above. It is normal to get a little tripped up if it's your first time seeing this table. In the first row, q doesn't even happen, so it might feel weird to say that p has "caused" q to happen in this case, and in the second row, p is not true and q is true, so it might seem odd that we have set $p \Rightarrow q$ to true, because there seems to be no relation between p and q. But it should make sense if you think of p as a precondition to a promise q, and then considering whether the promise is broken. As an example, suppose your friend says: "If it snows tomorrow I'll work in Trottier with you." If it doesn't snow and she doesn't pull up, she hasn't technically broken her promise. If it doesn't snow and she shows up, then she still hasn't broken her promise. The only way she can break her promise is if it snows and she doesn't come to Trottier; this situation corresponds to the only 0-row in the truth table.

The conditional is a very important logical operator to understand, because most theorem statements assume some hypothesis and claim some conclusion.

You will be asked to prove statements of this form, so it is important to understand the logical nature of the statements to begin with.

The last relation is called the *biconditional*, and it asserts that variables p and q are logically equivalent. That is, p happens if and only if q happens. It's truth table

p	q	$p \Leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

is pretty self-explanatory; in it, $p \Leftrightarrow q$ is true whenever p and q have the same truth value. The name "biconditional" is suggested by (part of) the following proposition.

Proposition 6. *Let p and q be propositional variables. Then*

$$\begin{aligned} p \Leftrightarrow q &\equiv (p \Rightarrow q) \wedge (q \Rightarrow p) \\ &\equiv \big((\neg p) \vee q\big) \wedge \big((\neg q) \vee p\big) \\ &\equiv (p \wedge q) \vee \big((\neg p) \wedge (\neg q)\big). \end{aligned}$$

Proof. We leave the first equivalence to the reader (writing out the truth table is one way of proving it). The second equivalence follows from our earlier observation that $p \Rightarrow q \equiv \neg p \vee q$, and the third equivalence follows from the distributive, complement, and identity laws. ∎

A formula f is called a

i) *tautology* if $f \equiv 1$, i.e., f always evaluates to true;
ii) *contradiction* if $f \equiv 0$, i.e., f always evaluates to false;
iii) *contingency* if f can evaluate to both 1 and 0, depending on the values of its variables;
iv) *satisfiable* if f evaluates to 1 for at least one input; and
v) *falsifiable* if f evaluates to 0 for at least one input.

An example of a tautology is $p \vee \neg p$ and an example of a contradiction is $p \wedge \neg p$. This follows from the complement laws. To say that something is satisfiable is precisely to say that it is not a contradiction, and to say that something is falsifiable is equivalent to saying that it is not a tautology. Contingencies are those formulas that are both satisfiable and falsifiable (in other words, formulas that are neither tautologies nor contradictions).

Suppose we are asked which of the above definitions the formula

$$f \equiv \big(p \wedge (p \Rightarrow q)\big) \Rightarrow q$$

satisfies. This formula only has two variables, so it is easy enough to use a truth table for this purpose, but we will take the opportunity to practise simplifying

the expression symbolically. First of all, let's change all conditionals of the form $r \Rightarrow s$, to disjunctions of the form $\neg r \vee s$. This gives us

$$f \equiv \neg\big(p \wedge (\neg p \vee q)\big) \vee q.$$

Now we use the distributive law to distribute the conjunction over the innermost disjunction, obtaining

$$f \equiv \neg\big((p \wedge \neg p) \vee (p \wedge q)\big) \vee q;$$

by the complement and identity laws in that order, this simplifies to

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

Now De Morgan's law and associativity give

$$f \equiv (\neg p \vee \neg q) \vee q \equiv \neg p \vee (\neg q \vee q),$$

and thus

$$f \equiv \neg p \vee 1 \equiv 1,$$

by the complement and domination laws in that order. We conclude that f is a tautology, which also means that it is satisfiable.

The fact that $\big(p \wedge (p \Rightarrow q)\big) \Rightarrow q$ is a tautology symbolically justifies the argument that of p is true and $p \Rightarrow q$ is true, then we should be able to conclude q. This form of argument is called *modus ponens*, and it dates back to ancient times. You probably use *modus ponens* all the time in everyday life without knowing it, and we will certainly use it in this class a lot.

Encoding problems in propositional logic. Many algorithmic and logical problems can be encoded in propositional logic (and then later solved by a computer program). For example, suppose we want to play 4×4 Sudoku. In this game, we have a 4×4 grid and we want to fill it with the numbers 1 through 4 such that

i) every row contains 1 through 4;
ii) every column contains 1 through 4; and
iii) the four subsquares each contain 1 through 4.

In a given instance of the game, some cells are already filled in. The puzzle is: *Is there a solution and if so, what is it?*

		2	
1	3		

Fig. 1. An example 4×4 Sudoku game.

To represent a Sudoku game in propositional logic, we can define boolean variables $p_{i,j,k}$, where i, j, and k range over $\{1,2,3,4\}$. (So there are 4^3 variables in total.) We shall set

$$p_{i,j,k} = \begin{cases} 1, & \text{if the number } k \text{ is in row } i \text{ and column } j; \\ 0, & \text{otherwise.} \end{cases}$$

We'll number the rows increasing from the top and the columns increasing from left to right. For example, in Fig. 1 there is a 2 in row 1 and column 3, so $p_{1,3,2} = 1$.

Now we set to work encoding the conditions of a Sudoku grid in propositional logic:

i) To stipulate that every row contain 1 through 4, we first define auxiliary variables

$$r_{i,k} = p_{i,1,k} \vee p_{i,2,k} \vee p_{i,3,k} \vee p_{i,3,k},$$

for $i, k \in \{1,2,3,4\}$. With these helper variables, we now see that

$$r_{1,1} \wedge r_{1,2} \wedge r_{1,3} \wedge r_{1,4}$$

encodes the requirement that row 1 contains one of each number. We do the same for rows 2, 3, and 4 as well, and then combine with AND.

ii) We do a similar thing as in part (i) for each of the four columns.

iii) Ditto for subsquares.

iv) We need to set the initial values of the grid. For the grid in Fig. 1, we have the formula

$$p_{1,3,2} \wedge p_{3,1,1} \wedge p_{3,2,3}.$$

v) Lastly, we need to make sure that there is not more than one number per cell. To do this, for each cell $(i,j) \in \{1,2,3,4\}^2$ we write

$$p_{i,j,1} \Rightarrow (\neg p_{i,j,2} \wedge \neg p_{i,j,3} \wedge \neg p_{i,j,4}),$$

and so on (four conditionals in total). Of course we'll need to AND all these together.

Now we combine the formulas from each of these five steps into one long formula f such that f is satisfiable if and only if the grid has a solution, and the values for $p_{i,j,k}$ give a solution.

Defining all of these variables was a rather arduous and cumbersome process, and not entirely worth it for a 4×4 game of Sudoku (which can just be solved by eyeballing the grid). But one could imagine writing a general computer program to encode larger and larger grids. In fact, there are lots of problems that can be *reduced* to the problem of determining if a boolean formula is satisfiable. This means that if we have a program capable of taking a formula f as input and

spitting out whether or not it is satisfiable (in a reasonable amount of time), then there are lots of real-world problems that this program could be applied to.

This problem is called the *boolean satisfiability problem*, often abbreviated SAT. One way of solving it for any given f is to just compute its truth table. We already know the downside of this approach: if f has n variables, then its truth table will have 2^n rows. Given a few minutes, you are certainly capable of writing down a formula f that has 300 variables, call them $p_1, \ldots, p_{300}$. The truth table of f will have 2^{300} rows, which is more than the number of atoms in the observable universe. You can learn a lot more about SAT in a higher-level class on computational complexity (e.g., COMP 360/362).

3. Predicate logic

12.IX Propositional logic allows us to work with simple declarations, but this isn't powerful enough to express some deeper mathematical concepts. For this purpose, we now introduce the notion of a *predicate*. This is a statement involving some number of variables, each of which may take values coming from a universe U, such that the statement evaluates to either true or false *once all variables are assigned values*. The statement $P(n)$ given by "n is prime" is an example of this, where n can take any value in the universe $\mathbf{Z}$. An example with two variables is the predicate $L(x, y)$ defined by "x is less than y." (A more commonly-used notation for this predicate is "$x < y$.")

Predicates contain variables, but at the moment we don't have any way of introducing new variables into a statement. This is done using two different *quantifiers*. The first is the *universal quantifier*, denoted $\forall$ and meaning "for all." The statement $\forall n : P(n)$ is true if and only if $P(n)$ is true for every possible value that n can take. The second quantifier is the *existential quantifier*, written "$\exists$" and with the meaning "there exists." The statement $\exists n : P(n)$ is true if and only if there is (at least) one value that n can take such that $P(n)$ is true. The colon doesn't really have any mathematical meaning in these formulas; they just visually set the quantifiers apart from the predicates that follow.

For instance, taking $P(n)$ to be the statement "n is prime," where the universe U is $\mathbf{N}$, the statement $\exists n P(n)$ is true and the statement $\forall n : P(n)$ is false. What about the statement $\forall x \, \exists y : y < x$? Well, if the universe U is taken to be $\mathbf{N}$, then the statement is false, because setting x equal to 0, there is no element y of $\mathbf{N}$ such that $y < 0$. But if $U = \mathbf{Z}$, then the statement is true, since for every integer x, we can put $y = x - 1$, so that $y < x$.

Now we practise translating converting mathematical statements written in English into formulas in predicate logic. Suppose we want to write: "Every integer is even or odd." The universe here is is the set $\mathbf{Z}$ of integers. The word "every" has the same meaning as "for all," so right off the bat, we can reexpress the statement as: "For all $n \in \mathbf{Z}$, n is even or n is odd." In symbols, this is

$$\forall n \, (n \text{ even} \vee n \text{ odd}).$$

Lastly, we need to figure out how to express the property of being even or being odd. An integer n is even if an only if it is a multiple of two; that is, if there is

some integer k such that $2k = n$. Likewise, an integer is odd if and only if it is one more than a multiple of two. The corresponding formula is $\exists k : 2k + 1 = n$. So our statement can be expressed

$$\forall n \big((\exists k : n = 2k) \vee (\exists k : n = 2k + 1) \big).$$

The variable k appears twice in this formula, but its first instance is independent of its second instance, because of the parentheses. (Readers who write computer programs will be familiar with the concept of a variable "going out of scope.") So there is nothing wrong with this formula, but to be extra clear that the first k is different from the second k, why don't we replace it with a different letter? Thus we arrive at

$$\forall n \big((\exists k : n = 2k) \vee (\exists l : n = 2l + 1) \big),$$

a formula in predicate logic that means "every integer is even or odd."

Restrictions using quantifiers. It is not true that every real number has a multiplicative inverse, since one cannot divide by zero. However, the statement "every nonzero real number has a multiplicative inverse" is true. How should we write this as a formula in predicate logic? One way is to use the conditional: over the universe $U = \mathbf{R}$, we could write

$$\forall x : (x \neq 0 \Rightarrow \exists y : xy = 1).$$

Another is to use subscripts: in the same universe, we write

$$\forall x_{x \neq 0} \, \exists y : xy = 1.$$

Using subscripts is slightly informal, since we didn't formally define above what a subscript is supposed to mean, but it is something that you might encounter. The last way is to simply restrict the universe itself: in the universe $U = \mathbf{R} \setminus \{0\}$, the statement

$$\forall x \, \exists y : xy = 1$$

is true.

Multiple quantifiers. Withing a formula, quantifiers cannot be interchanged willy-nilly. The order of $\forall$ and $\exists$ matters! They are read from left to right. Consider the following examples, over the universe $U = \mathbf{R}$. The statement

$$\forall x \, \exists y : x + y = 0$$

is true, since for each given x we can take y to be $-x$. On the other hand,

$$\exists y \, \forall x : x + y = 0$$

is false, since it would mean that there is some integer that adds up to zero with *any* integer. Of course, sometimes switching the order of quantifiers, doesn't change the truth value of a statement. Both

$$\exists y \, \forall x : xy = 0$$

and

$$\forall x \, \exists y : xy = 0$$

are true, since in the first case, we can take $y = 0$, and in the second case, we can set y to 0 no matter what x is given.

So we know that the order of $\forall$ and $\exists$ matters in general, but repeated instances of the *same* quantifier *can* be interchanged. For instance,

$$\forall x \, \forall y : x^2 + y^4 \geq 0$$

is the same as

$$\forall y \, \forall x : x^2 + y^4 \geq 0,$$

and we can even write $\forall x, y : x^2 + y^4 \geq 0$, to introduce both variables simultaneously.

Negating quantifiers. The universal quantifier is sort of like a big chain of conjunctions that goes over the whole of the universe. For example, in the universe $\mathbf{N}$, the statement $\forall n : P(n)$ is equivalent to

$$P(1) \wedge P(2) \wedge P(3) \wedge \cdots,$$

if this were a valid propositional formula (it isn't because we don't allow propositional formulas to be infinite). Likewise, the existential quantifier $\exists n : P(n)$ is equivalent to

$$P(1) \vee P(2) \vee P(3) \vee \cdots.$$

We know, by De Morgan's laws, that negating a big series of conjunctions requires us to flip all the ANDs to ORs. So, once again abusing notation somewhat, we expect

$$\neg\big(P(1) \wedge P(2) \wedge P(3) \wedge \cdots\big) \equiv \neg P(1) \vee \neg P(2) \vee \neg P(3) \vee \cdots.$$

Thus we conclude that

$$\neg\big(\forall n : P(n)\big) \equiv \exists n : \neg P(n).$$

You can play the same game with the other De Morgan's law to show that

$$\neg\big(\exists n : P(n)\big) \equiv \forall n : \neg P(n).$$

Going back to our example of $P(n)$ denoting "n is prime," the statement $\neg\big(\forall n : P(n)\big)$ is true, since not all integers n are prime, and we have just shown that

this is equivalent to the statement $\exists n : \neg P(n)$; that is, there exists n such that n is not prime.

We end this section with a longer example. Let's express the statement, "There is a nonzero real number such that every real number is not its inverse or is negative." In the universe $\mathbf{R}$, the formula corresponding to this statement is

$$\exists x : (x \neq 0 \wedge (\forall y : xy \neq 1 \vee y < 0)\big).$$

(Work it out yourself!) Is this statement true or false? It turns out that it is true. You might be able to stare at the formula long enough to convince yourself of this fact, but another way to see that it's true is to note that its negation is false. Let's do this now (it's a good excuse to practise negating a formula). We have

$$\begin{aligned}
\neg\Big(\exists x : (x \neq 0 \wedge (\forall y : xy \neq 1 \vee y < 0))\Big) & \\
&\equiv \forall x : \neg(x \neq 0 \wedge (\forall y : xy \neq 1 \vee y < 0)\big) \\
&\equiv \forall x : \big(x = 0 \vee \neg(\forall y : xy \neq 1 \vee y < 0)\big) \\
&\equiv \forall x : \big(x = 0 \vee \exists y : \neg(xy \neq 1 \vee y < 0)\big) \\
&\equiv \forall x : \big(x = 0 \vee \exists y : (xy = 1 \wedge y \geq 0)\big).
\end{aligned}$$

This negated statement is false, since if $x = -2$, then $x = 0$ doesn't hold, so the left-hand side of the OR isn't true, and there is no y such that $-2y = 1$ and $y \geq 0$ are both true, since the only y satisfying $-2y = 1$ is $-1/2$.

Negating a formula in predicate logic is entirely mechanical. The $\neg$ symbol moves from left to right like a bulldozer that flips quantifiers and negates predicates it finds along the way, until eventually its job is done and it disappears. More broadly, we write mathematical statements in formal logic to make things more precise and mechanical. This can be useful to humans, since English is often ambiguous whereas the notation we just established is not. It can be useful to machines as well, since, as we just saw, manipulating a formula is something that can very easily be automated.

4. Proofs

We're getting to the fun part of the course now. Further back in these notes, we already proved a few statements about sets. This was just a taste of what's to come, as the main focus of this course is to teach you how to prove mathematical statements. These will all be statements that can ultimately be stated in predicate logic, so using the tools from the previous section, you can boil them down to their logical skeleton. In this section, we will formally define what it means to prove a statement in predicate logic.

To prove a statement, we always process the quantifiers from left to right. Whenever we encounter something of the form $\exists x : P(x)$, we are allowed to choose the value for x (from the given universe), and we just need to show that $P(x)$ holds for that value of x. Here's an example.

Proposition 7. *There exists an integer $m > 0$ and an integer $n < 0$ such that $m^2 + n^2 = 25$.*

We've written the statement in words, and we will continue to do so for all the propositions in these notes because we are humans and not cyborgs, but notice that the underlying predicate logical formula here is

$$\exists m\, \exists n : m > 0 \wedge n < 0 \wedge m^2 + n^2 = 25,$$

with $U = \mathbf{Z}$. For the remainder of this section, we'll continue to write out the formulaic equivalents of propositions, to practise converting between the two worlds.

Proof. After a moment's contemplation, we notice that $9 + 16 = 25$. So we need a positive integer that squares to to 9 and a negative number that squares to 16. Hence we may pick $m = 3$ and $n = -4$, and $m^2 + n^2 = 25$. ∎

On the other hand, to prove a statement of the form $\forall x : P(x)$, we are not allowed to choose the value of x. Instead, we imagine it is given to us, and we still have to prove $P(x)$, no matter what the x might be. Here's what we mean.

Proposition 8. *For all $x \in \mathbf{Q}$, there exists $y \in \mathbf{Z}$ such that $xy \in \mathbf{Z}$.*

The formula this time is

$$\forall x\, \exists y : y \in \mathbf{Z} \wedge xy \in \mathbf{Z},$$

over $U = \mathbf{Q}$.

Proof. Let $x \in \mathbf{Q}$. Then x can be expressed as the ratio of two integers; write $x = m/n$ with $m, n \in \mathbf{Z}$. Then, setting $y = n \in \mathbf{Z}$, we have $xy = (m/n) \cdot n = m \in \mathbf{Z}$. ∎

Notice how we introduced the variable x in the above proof. Because we're trying to prove a "for all" statement, we use the word "let," to indicate that x is given to us by some higher power. In fact, we should sort of view this higher power as possibly being malicious. A very useful way to think about writing proof is to imagine a game in which you are trying to prove a statement in predicate logic, and a supernatural adversary is attempting to thwart you. Let's say this predicate has four variables, so the statement is

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

The variables in the statement are introduced from left to right. Each time you see a $\exists$ symbol, it's your turn. In the example above, you get to set the variable x to any element of the given universe (keeping in mind that your eventual goal is to prove $P(x, y, z, w)$). Each time there is a $\forall$ symbol, it's the adversary's turn, and you should be prepared for whatever he throws at you. So in our example, the adversary may set y and z to anything in the universe, and he knows you picked x. Lastly, you get to pick w. If $P(x, y, z, w)$ is true, you win, and if not, the adversary wins. Writing a proof is equivalent to describing a winning strategy for the player against the adversary.