Sets

REASONING

elaine rich
alan kaylor cline

The Logicians on our cover are:

Euclid (? - ?)

Augustus De Morgan (1806 – 1871)

Charles Babbage (1791 – 1871)

George Boole (1815 – 1864)

Aristotle (384 BCE – 322 BCE)

George Cantor (1845 – 1918)

Gottlob Frege (1848 – 1925)

John Venn (1834 – 1923)

Bertand Russell (1872 – 1970)

REASONING

AN INTRODUCTION TO LOGIC, SETS, AND FUNCTIONS

CHAPTER 9
SETS

Elaine Rich
Alan Kaylor Cline

The University of Texas at Austin

Image credits:

Party invitations: © Lynda Trader, 2014.

Sandwiches: © Lynda Trader, 2014.

Short order cooks: © Lynda Trader, 2014.

Planets: http://nineplanets.org/

Fortune cookies: http://www.theboredninja.com/featured/the-powerball-fortune-cookie/

Hindu gods: http://www.dollsofindia.com/product/hindu-posters/brahma-vishnu-shiva-reprint-on-paper-AE99.html

Beatles:

http://www.cs.utexas.edu/learnlogic
Library of Congress Cataloging-in-Publication Data

Rich, Elaine, 1950 -
Reasoning—An Introduction to Logic Sets and Functions / Elaine Rich.— 1st ed. p. cm.
ISBN x-xxx-xxxxx-x 1

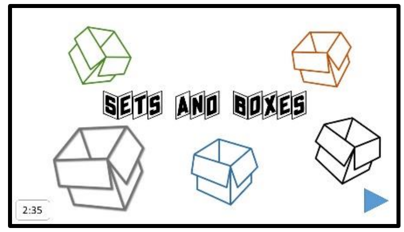
Table of Contents

Chapter: Sets

The Key Ideas

What Is a Set?

A ***set*** is an *unordered* collection of *unique* objects (which we'll call ***elements***). The easiest way to specify a set is simply to list its elements. By convention, they are enclosed in braces.



https://www.youtube.com/watch?v=nmNuHwhq6LA

All of the following are sets:

$S_1 =$ {1, 3, 789}

$S_2 =$

When it is obvious that we are using a particular, previously defined sequence, we will allow ourselves to use the … notation.

Let S_4 = {a, b, c, … z}. This means that S_4 contains the 26 lower case letters of the Latin alphabet.

We’ll use this notation when we need it, but we must be careful that we do it only when no confusion can result.

Next, we define two symbols that we can use to talk about the elements of a set:

x is an element of set S:	$x \in S$
x is not an element of set S:	$x \notin S$

Given the definition of S_3 above:

 $\in S_3$  $\notin S_3$  $\notin S_3$

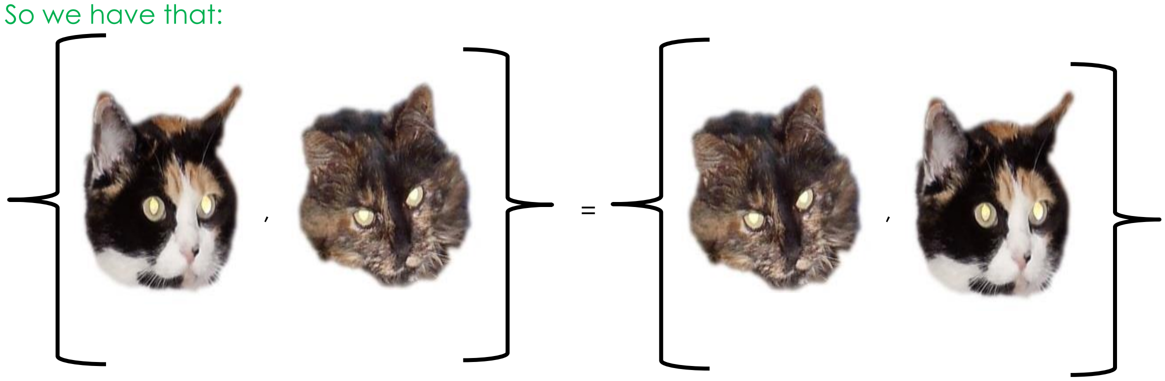
Notice that a claim, such as $x \in S$ or $x \notin S$ is either true or false. So it’s a logical statement and we can use it to build larger logical statements using the logical operators.

For example, let $S_1 = \{1, 3, 789\}$. Then these logical statements are all true:

[1] $(3 \in S_1) \wedge \neg(5 \in S_1)$
[2] $(25 \notin S_1) \wedge (3 \in S_1)$
[3] $(25 \notin S_1) \vee (3 \in S_1)$

We’ll say that two sets S and T are ***equal*** if and only if they contain the same elements. In this case, we’ll write $S = T$. Note that, given this definition, the order in which we list the elements plays no role in determining what the set actually is.

So we have that:



And:

{cat, dog, hedgehog, bird } = {hedgehog, cat, dog, bird} = {cat, dog, bird, hedgehog}

Also note that duplicating an element does not change the value of the set.

So we have that:

{cat, dog, cat, bird, cat} = {cat, dog, cat, bird, cat, bird} = {cat, dog, bird}

And:

{2, 2, 2, 2, 2} = {2}

Problems

1. Let *S* = {Dasher, Prancer, Blitzen}.

   For each of the following, indicate whether or not it is equal to *S*.
   a) {Dasher, Blitzen, Prancer} .
   b) {Dasher, Blitzen, Donder}

2. Let *Greeks* = {Aristotle, Socrates, Plato}. For each of the following indicate T/F:

   a) Aristotle $\in$ *Greeks*
   b) Plato $\notin$ *Greeks*
   c) Diosthenes $\notin$ *Greeks*
   d) (Aristotle $\in$ *Greeks*) $\wedge$ $\neg$(Sophocles $\in$ *Greeks*)

3. Let *Babies* = {cub, cygnet, fawn, gosling, kitten, piglet, puppy}. For each of the following sets *S*, mark True if *S* = *Babies* and False otherwise.

   a) *S* = {kitten, puppy, piglet, fawn, gosling, cub, cygnet}
   b) *S* = {fawn, piglet, kitten, fawn, gosling, cub, piglet, cygnet, puppy}
   c) *S* = {kitten, puppy, piglet, tadpole, gosling, cub, cygnet}

Some Important Sets

The smallest set is the set that contains no elements. We'll call that set the ***empty set***. There are two common ways to write the empty set:

- $\{\}$
- $\varnothing$

Formally, we can define $\varnothing$ by saying:

$$\forall x \, (x \notin \varnothing)$$

Let's now define some important sets that actually contain elements. (Note that the names we are giving to these particular sets are very standard.)

Z = the ***integers***.
Z⁺ = the ***positive integers*.**
N = the ***natural numbers***. We'll define **N** to contain the integers that are greater than or equal to 0, but be careful here. Some books do not include 0 as a natural number.
Q = the ***rational numbers*** (i.e., those numbers that can be represented as the ratio of two integers).
R = the ***real numbers***.

Note that we could have defined **R** to be the **R**ationals. It's just convention to define it to be the reals instead. Then it was necessary to pick something else as a name for the rationals.

Problems

1. Which of these statements is/are true:

$7 \in \varnothing$
$7 \in \mathbf{N}$
$7 \in \mathbf{Z}$
$7 \in \mathbf{Q}$
$7 \in \mathbf{R}$

2. Which of these statements is/are true:

$\frac{4}{2} \in \mathbf{N}$

$\frac{4}{2} \in \mathbf{Z}$

$\frac{4}{2} \in \mathbf{Q}$

$\frac{4}{2} \in \mathbf{R}$

3. Which of these statements is/are true:

$\frac{3}{2} \in$ **N**

$\frac{3}{2} \in$ **Z**

$\frac{3}{2} \in$ **Q**

$\frac{3}{2} \in$ **R**

4. Which of these statements is/are true:

-3 $\in$ **N**

-3 $\in$ **Z**

-3 $\in$ **Q**

-3 $\in$ **R**

5. Let *S* = {2, 3, 5}. For each of the following sets, mark True if it is equal to *S* and False otherwise. (Note here that we are considering sets of numbers, not the symbols that we use to denote them.)

{2, $\frac{4}{2}$, 3, $\frac{-8}{-4}$, 5}

{5, 5, $\sqrt[+]{25}$, 3, 2}

{5, 3, 3, 2}

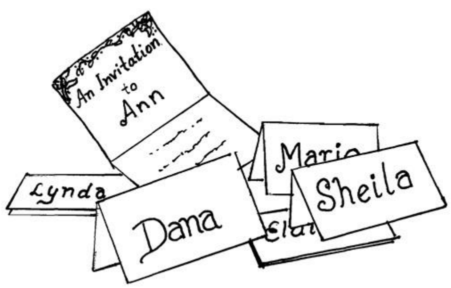
Sets, Multisets and Lists

Recall that a set is an *unordered* collection of *unique* objects. Why have we defined it in this way? Is this always what we want?

We've defined it this way because this notion plays an important role in mathematics. We need to understand it so that we can exploit it.

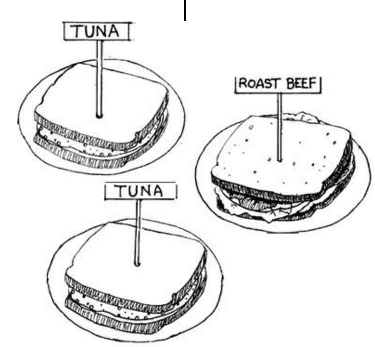
But is it always what we want, particularly if we want to write programs to solve real problems? Now the answer is no:

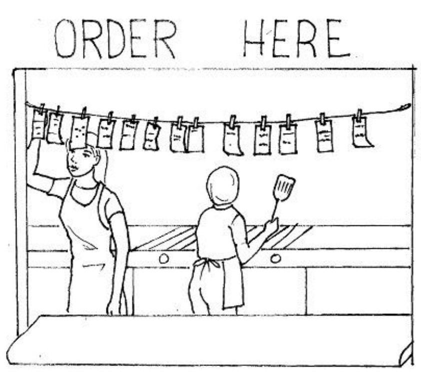
- Sometimes we want duplicates to matter.
- Sometimes we want order to matter.



Suppose that you and a group of friends are planning a party. You've agreed that each of you can add whomever you like to the invitation list. If you want to write a program to manage the invitations, the data structure that you want here is a set. If two people add Dana to the list, she still gets just a single invitation.

But now suppose that one of you has offered to run to the deli for sandwiches. Everyone is asked to submit an order. Now, if two people ask for tuna, it's important that tuna get entered twice.





Finally, suppose that you're the manager of the deli. You want to keep track of the orders that have come in. Now you not only want to consider duplicates. You'd better also keep track of which orders came in ahead of which other ones. If you don't, you'll have some very unhappy customers.

Because all three of these situations happen a lot, programming environments typically support these three important data structures:

Set Corresponds to the mathematical definition of set that we've presented. We'll be able to use the theory that we develop here if we want to prove things about programs that use sets.

Multiset (sometimes called *bag*) Like a set except that duplicates are allowed. We'll come back to this idea later and build on our theory of sets to construct a corresponding one for multisets.

List An ordered set or ordered multiset. It's possible also to build a theory that lets us reason about programs that exploit lists.

> ***Nifty Aside***
>
> We've described here a simple list, for example (roast beef, ham and swiss, roast beef, peanut butter and jelly}. In real programs, there are actually many other structures that capture the notion of order. Many of them are more efficient to implement, for some problems, than the simple list.

At this point, you may be thinking, "But you said that the integers are a set (not a list). Yet they clearly have order. There's a very useful sense in which 1 'comes before' 2." True. But what we're going to do is to start by treating the integers (and the natural numbers and the rationals and the reals) as sets. Then we'll see ways to impose order on them when we need to do that.

Problems

1. For each of the following problems, choose the best data structure:

(Part 1) Eligible voters in our voting precinct.

a) Set: We don't want duplicates and order doesn't matter.
b) Multiset: We do want duplicates but order doesn't matter.
c) List: Order matters.

(Part 2) People standing outside the concert venue, hoping to be able to buy tickets.

a) Set: We don't want duplicates and order doesn't matter.
b) Multiset: We do want duplicates but order doesn't matter.
c) List: Order matters.

(Part 3) Names on the raffle tickets that we've sold for our fundraiser.

a) Set: We don't want duplicates and order doesn't matter.
b) Multiset: We do want duplicates but order doesn't matter.
c) List: Order matters.

Defining a Set

We've already seen a simple technique for defining a set: We simply write down the elements.

> For example, S_1 is a set of symbols that correspond, in various alphabets (Latin, Greek, Russian, Arabic, Hindi, Thai, Tibetan), to the initial sound of the word "lemon".
>
> S_1 = {L, Λ, Л, ل, ल, ล, ལ}

Sometimes it's impossible or impractical to list all the elements of a set. An alternative is to write a logical description of the set. To do this, we write a predicate that is true of all and only the elements of the set that we wish to define. Then we can use any of these notations:

$S = \{x : P(x)\}$ or $S = \{x \mid P(x)\}$ or $S = \{x \ni P(x)\}$

Read all of them as, "*S* is the set of all objects such that *P* is true of *x*}. We'll use the first of these notations, but the second one is also common; the third is less so.

It is also sometimes useful to start by restricting the values we consider to ones that belong to some other set that has already been defined. In that case, we can write:

$S = \{x \in R : P(x)\}$

Read this as, "*S* is the set of all objects in *R* such that *P* is true of *x*}.

> For example, suppose that we want to define S_2 to be the set of all even integers greater than 10. Assume that we have already defined, on the integers:
>
> $Div(x, y) \equiv ((y \neq 0) \wedge \exists z\ (x = y \cdot z))$ (x is divisible by *y*)
>
> Then we can write:
>
> $S_2 = \{x \in \mathbf{Z} : (x > 10) \wedge (Div(x, 2))\}$
>
> Read this as, "S_2 is the set of all objects x in **Z** (the integers) such that x > 10 and *Div*(x, 2) is true.

> Suppose that we want to define S_3 to be the set of all people who are mothers of at least two children. Then we can write:
>
> $S_3 = \{x \in People : \exists y,z\ (MotherOf(x, y) \wedge MotherOf(x, z) \wedge (y \neq z))\}$
>
> Read this as, "S_3 is the set of all people x such that there exist a *y* and a z, where x is the mother of *y* and x is the mother of z and *y* and z are different."

Sometimes we want to define a set by a program that we can run. In that case, we've got two choices:

- We can write a ***generator*** (also called an ***enumerator***). Its job is to output (in some unspecified order) all and only the elements of the set.

- We can write a ***recognizer***. Its job is to implement a predicate definition. In other words, it must examine a candidate and return *True* if the candidate is in the set and *False* otherwise.

If we do either of these things, we can say that our program defines the corresponding set.

Suppose that we have a database that contains the personnel records of all the employees of our company. The online-games division needs to work with the set of people who are eligible for a 5-year pin at the upcoming division retreat. We'll call this set *Gets_pin*. Assume that we've already defined three procedures that access our database: employees returns the set of badge numbers of all company employees. division is given a badge number and returns the name of the division in which the employee with that badge number works. service_years is given a badge number and returns the number of years of service of the employee with that badge number.

Here's a generator for *Gets_pin*. It takes no input. It prints a list of all the elements of the set.

```
def Gets_pin():
   # Walk through set of employees, checking division and service_years.
   for badge in employees:
       if division(badge)== "online-games" and service_years(badge)== 5:
            print(badge)
```

Here's a recognizer for *Gets_pin*. It takes a single input, an employee badge number. It returns True if that employee is an element of *Gets_pin*. It returns False otherwise.

```
def Gets_pin(badge):
    # Check a particular badge.
    if division(badge)== "online-games" and service_years(badge)== 5:
       return(True)
    else:
       return(False)
```

Recognizers are particularly useful in real applications.

Let *Java* be the set of syntactically well-formed Java programs.

How shall we give a formal definition of this set?

Answer: The Java compiler is a recognizer for this set. If you give the compiler a string that is a legal Java program, it will compile. If you give it a string that isn't a legal Java program, it will complain and produce error messages.

Nifty Aside

There exist sets for which it's not possible to write a recognizer or a generator. For example:

Let *JavaWithoutInfiniteLoops* be the set of syntactically well-formed Java programs that are guaranteed to halt on all inputs.

It's easy to understand what we mean by this definition. And equally easy to see why we'd want to be able to make sure that a program that we've written is in the set. But it's possible to prove that no generator or recognizer (in any programming language, running on any sort of machine) will ever exist for this language. This observation is called the ***undecidability of the halting problem***.

Problems

1. Assume that we are working just with elements of **N** (the natural numbers). Mark each of the following statements True or False.

a) $7 \in \{n : \exists y > 5\ (n = y + 2)\}$
b) $2 \in \{n : \exists x, y\ (n = x/y)\}$
c) $34 \in \{n : \exists x\ (prime(x) \wedge n = 2x)\}$
d) $2 \in \{n : \forall y\ ((1 < y < n) \rightarrow prime(y))\}$
e) $5 \in \{n : \forall y\ ((1 < y < n) \rightarrow prime(y))\}$
f) $81 \in \{n : \exists x\ (x \text{ is a perfect square} \wedge n = x^2)\}$
g) $\{2, 3\} \in \{n : prime(n)\}$

2. We want to define the set $S = \{1, 4, 9, 16, 25, 36, \ldots\}$ using the logical predicate form. In other words, we want to write:

$S = \{x \in$ ___ $: P(x)\}$, for some value filled in the blank and some predicate P.

For each of the following expressions, choose True if it correctly describes S. Choose False otherwise:

a) $S = \{x \in \mathbf{Z} : \exists y\ (y \in \mathbf{N} \wedge y^2 = x)\}$
b) $S = \{x \in \mathbf{N} : \exists y\ (y \in \mathbf{N} \wedge x^2 = y)\}$
c) $S = \{x \in \mathbf{N} : \exists y\ (y \in \mathbf{Z} \wedge y \neq 0 \wedge y^2 = x)\}$
d) $S = \{x \in \mathbf{N} : \exists y\ (y \in \mathbf{Z} \wedge y \neq 0 \wedge x^2 = y)\}$

3. Let *Elements* be the set of elements in the Periodic Table. Consider the set:

$S = \{x \in Elements : molecular\text{-}weight(x) < 50 \wedge valence(x) = 1\}$

True or false?

a) Fe $\in S$
b) Mg $\in S$
c) Na $\in S$

4. Let $S = \{x \in \mathbf{N} : (x > 10) \rightarrow Div(x, 2)\}$. For each of the following, indicate True if it correctly describes all and only the elements of S; write False otherwise.

a) {0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, …}
b) {1, 3, 5, 7, 9, 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, …}
c) {12, 14, 16, 18, 20, 22, 24, 26, …}
d) {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 22, …}
e) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 22, …}

Using Sets in Logical Expressions

Now that we've got an explicit definition of what we mean by a set, we can use it to describe the universe with respect to which we want to make a logical claim. In particular, we can write expressions of the form:

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

Read this as, "For all *x* in *S*, *P* is true of *x*." Or, more concisely, "For all *x* in *S*, *P* of *x*."

> Suppose that we want to define the set of perfect squares. We can write:
>
> $PerfectSquares = \{x \in \mathbf{N} : \exists y \in \mathbf{N}\ (y^2 = x)\}$
>
> Read this as: *PerfectSquares* is the set of all values x in **N** such that there exists a *y* in **N** where $y^2 = x$. Or we can say that *PerfectSquares* is the set of all natural numbers x such that exists a natural number *y* such that $y^2 = x$.

A huge win of formal notations like this is that they are substantially less ambiguous than English is.

> Returning to the perfect squares example: Someone reading the English definition might wonder whether 0 is included. The formal definition we've given, however, is clear. 0 is in.

The use of set notation as a way to write an explicit description of a universe is particularly useful if we want to write multiple quantifiers whose domains are different.

> Suppose that we want to assert that any *real number* that has an *integer* square root must also be an integer. We can write:
>
> $\forall x \in \mathbf{R}\ ((\exists y \in \mathbf{Z}\ (y^2 = x)) \rightarrow x \in \mathbf{Z})$

> Suppose that we want to say that every student got a recommendation from a teacher. We could define the universe to be the set of all people. Then we could write:
>
> $\forall x\ (Student(x) \rightarrow \exists y\ (Teacher(y) \land WroteRecommendationFor(y, x)))$
>
> We could also specify more restricted domains for each quantified expression. Then we could write a simpler expression:
>
> $\forall x \in Students\ (\exists y \in Teachers\ (WroteRecommendationFor(y, x)))$

In this last example, notice the natural correspondence between being a member of a set and being an object of which some predicate is true.

Given the appropriate definitions, we would have that:

$x \in Students$ if and only if $Student(x)$

This should not be surprising given that one natural way to define a set (in this case *Students*) is as a collection of objects about which some predicate (in this case *Student*) is true.

We'll have more to say about this later after we've introduced operations on sets. We'll see that there's a natural correspondence between them and the operations $\wedge$ and $\vee$ that can be performed on logical expressions.

Problems

1. Consider the following partially written formula:

$\forall x \in$______ $((x + 1) \in \mathbf{Z})$

Assume we want to make this statement as strong as we can (i.e., want to make a claim about all the elements of which it must be true). What value should we fill in the blank?

a) **Z** (the integers)
b) $\mathbf{Z}^{+}$ (the positive integers)
c) **N** (the natural numbers)
d) **Q** (the rationals)
e) **R** (the reals)

2. Consider the following partially written formula:

$\forall x,y \in$______ $((y \neq 0) \rightarrow \exists z \in \mathbf{Q}\ (z = \frac{x}{y}))$

Assume we want to make this statement as strong as we can (i.e., want to make a claim about all the elements of which it must be true).

What value should we fill in the blank?

a) **Z** (the integers)
b) $\mathbf{Z}^{+}$ (the positive integers)
c) **N** (the natural numbers)
d) **Q** (the rationals)
e) **R** (the reals)

How Large is a Set?

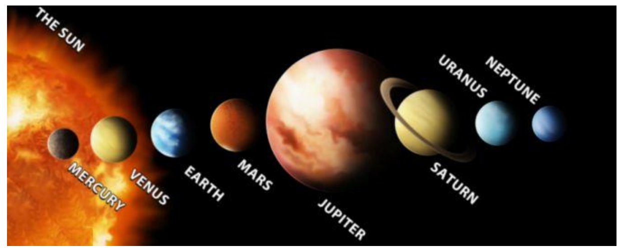
A set S is ***finite*** if it has exactly n distinct elements where n is some natural number.

In this case, we'll say that the ***cardinality*** of S is n. We'll write it as:

$|S| = n$ or $Card(S) = n$

Let *Planets* = {Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune}. (Notice that we've accepted that Pluto got demoted and is no longer in the set.)

| *Planets* | = 8



If a set is not finite, we'll say that it is ***infinite***.

The set S = {22, 28, 32, 33, 39, 40} is finite.

| S | = 6.

The set **Z** (the integers) is not finite.

The empty set contains no elements. So we have:

$|\varnothing| = |\{\,\}| = 0$

Remember that duplicates don't change the value of a set.

| {2, 4, 8, 16} | = | {2, 2, 2, 4, 4, 4, 8, 16} | = 4

Problems

1. Let:

- P_1 be the set of prime numbers between 0 and 10.
- P_2 be the set of prime numbers between 10 and 20.

Prove that $|P_1| = |P_2|$.

When you've finished your proof, look at these claims. Indicate which of these claims are true and which are false:

a) There are 4 elements in P_1.
b) $|P_2| = 5$.
c) P_1 finite.
d) P_2 is not finite.

2. Alan Turing is a hero to computer scientists. He played a major role in breaking the German U-boat code during World War II. He proved a fundamental result that shows that there are limits to what we'll ever be able to compute. He wrote a famous paper that attempted to describe a technique by which we'd know whether a computer was "thinking".

Let C_T be the set of Alan Turing's children. What is $|C_T|$? (Hint: Use Google.)

3. Cardinalities are numbers. So we can, of course, do arithmetic with them. For each of the following statements, indicate whether it is True or False:

a) $|\{1, 2, 3, 3\}| - |\{1, 2, 3\}| = 1$
b) $|\{3, 9, 27\}| * |\emptyset| = 1$
c) $|\{4, 9, 25\}| = |\{2, 3, 5\}|$

Subsets, Supersets, Powersets and Partitions

Subsets and Supersets

We'll say that A is a ***subset*** of B if every element of A is also in B. In this case, we'll write:

$$A \subseteq B$$

More formally:

$A \subseteq B$ if and only if $\forall x\,((x \in A) \rightarrow (x \in B))$

The subset relationship holds between some pairs of very important sets.

> For example, every natural number is also an integer. So we have:
>
> $\mathbf{N} \subseteq \mathbf{Z}$

The subset relationship often also holds between sets that are significant for particular problems.

> Every freshman is a student. So we have:
>
> *Freshmen* $\subseteq$ *Students*

A very simple claim about $\subseteq$ follows directly from its definition:

Every set is a subset of itself. In other words:

$$\forall S\,(S \subseteq S)$$

This must be so since every element of S is also in S.

> So we have, for example:
>
> $\mathbf{Z} \subseteq \mathbf{Z}$

Sometimes we want a slightly different notion. We'd like to say that A is a subset of B and it isn't exactly B. In other words, every element of A is in B and there's at least one element of B that is left out – it isn't also in A.

In this case, we'll say that A is a ***proper subset*** of B and we'll write:

$$A \subset B$$

More formally:

$A \subset B$ if and only if $\forall x\ (((x \in A) \rightarrow (x \in B)) \wedge (A \neq B))$

Notice the analogy between $\subseteq$ / $\subset$ and $\leq$ / $<$.

Every natural number is also an integer. And there are integers (for example, -3) that are not natural numbers. So we have:

$\mathbf{N} \subset \mathbf{Z}$

But this claim is false:

$\mathbf{Z} \subset \mathbf{Z}$

Just as we have both $\leq$ / $<$ and $\geq$ / $>$, we can write the subset relation in the other direction. We'll say that A is a ***superset*** of B if every element of B is also in A. In this case, we'll write:

$A \supseteq B$

So we have:

$\mathbf{Z} \supseteq \mathbf{N}$ (The integers are a superset of the natural numbers.)

Students $\supseteq$ *Freshmen*

Similarly, if A is a superset of B, but not equal to B, we'll say that A is a ***proper superset*** of B and we'll write:

$A \supset B$

Just as we have that, if $a \leq b$, then $b \geq a$, we have the following theorem about subsets and supersets:

For any sets S and T: $(S \subseteq T) \equiv (T \supseteq S)$

$(S \subset T) \equiv (T \supset S)$

Problems

1. For each of the following claims, indicate whether it is true or false:

a) $\{5, 9, 4, 3\} \supseteq \{3, 4, 5, 9\}$
b) {U.S. state capitals} $\subset$ {U.S. cities with population < 1 million}
c) {prime numbers} $\subset$ **Z**.
d) {dogs} $\supseteq$ {poodles}

2. For each of the following claims, indicate whether it is true or false:

a) $\varnothing \subseteq \varnothing$
b) $\varnothing \subset \varnothing$
c) $\varnothing \in \varnothing$

3. For each of the following claims, indicate whether it is true or false:

a) $\varnothing \subseteq \{\varnothing\}$
b) $\varnothing \in \{\varnothing\}$
c) $\varnothing = \{\varnothing\}$

4. Consider the claim:

$$\varnothing \subseteq A$$

This claim is true for:

a) All sets *A*.
b) Some but not all sets *A*.
c) No set *A*.

5. Consider the claim:

$$\varnothing \in A$$

This claim is true for:

a) All sets *A*.
b) Some but not all sets *A*.
c) No set *A*.

Sets of Sets

So far, almost all of the set elements that we've considered have been individual values, like 3, Austin, and Mars. The one exception has been {∅}, the set whose single element is a set that has no elements.

But we can also define sets whose elements are other more interesting sets.

Every freshman is a student. So we have:

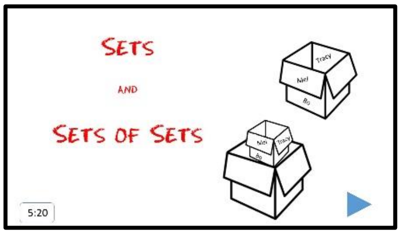
Freshmen ⊆ *Students*

Suppose that we want to describe the choices that will confront the diners at our new restaurant. Define:

Menu = {*Appetizers*, *Entrees*, *Sides*, *Desserts*}

= {{ceviche, nachos}, {fajitas, carnitas, enchiladas}, {beans, rice}, {sopapillas, flan}}

It's very important to be careful about the distinction between a set and an element of a set.



https://www.youtube.com/watch?v=oXRRu45xYgo

Let S_1 = {{a}}. Then a ∉ S_1. S_1 has a single element, {a}, which is a set that, in turn, contains the single element a.

Let S_2 = {{a}, a}. Then a ∈ S_2. S_2 has two elements, one of which is the set {a}, and the other of which is a.

Let S_3 = {∅}. $|S_3|$ = 1 because it contains exactly one element. That element, in turn, is the empty set. The empty set has cardinality 0, but that's not what we're talking about.

Nifty Aside

Let R be the set that contains all sets that are not elements of themselves. Stated more formally, $R = \{S : S \notin S\}$. So, for example $\{1, 2, 3\} \in R$. Clearly $\{1, 2, 3\}$ is not an element of itself since it contains no sets at all.

But now for the hard one: Is R an element of R? The answer to this question must be either yes or no:

- Suppose that R is an element of R. Show that this leads to a contradiction (just using the definition of R).
- Okay, then suppose that R is not an element of R. Show that this also leads to a contradiction.

We said that one of these had to be the answer. Yet neither of them can be. This problem is called ***Russell's Paradox*** (named for the logician Bertrand Russell). We can back ourselves out of this corner by disallowing set definitions of this sort. But that's beyond the scope of this course.

Problems

1. Let S = {{azul, cyan, navy}, {pink, fuscia, rose, magenta}, {chartreuse, lime}}. What is $|S|$?

2. Let $S = \{\{\}, \varnothing, \{\varnothing\}\}$. What is $|S|$?

3. Let $S = \{a, \{a\}, S, b\}$. For each of the following elements, mark True if it is in S and False otherwise:

$\{a\}$
$\{\{a\}\}$
$\{b\}$
$\{S\}$
S

The Powerset of S

There's one set of sets that is particularly useful.

If S is a set, define the ***powerset*** of S to be the set of all subsets of S. We'll denote the powerset of S as:

$$\wp(S)$$

For any sets A and S:

$A \in \wp(S)$ if and only if $A \subseteq S$

Suppose that S = {cake, brownies, cookies, candy, fondue, fudge, tarts, cupcakes}. We're planning a chocolate party. $\wp(S)$ is the set of all of our menu options. Some of the elements of $\wp(S)$ are:

{cake, candy, tarts, brownies}
{cupcakes, fondue}
{candy}
{}
{cake, brownies, cookies, candy, fondue, fudge, tarts, cupcakes}

Problems

1. Let *HinduGods* = {Brahma, Vishnu, Shiva}.

   List the elements of $\wp$(*HinduGods*). How many are there?

2. Again let *HinduGods* = {Brahma, Vishnu, Shiva}.

   Mark each of the following as True if it is an element of $\wp$(*HinduGods*). Mark it False otherwise.

   a) {Vishnu, Brahma, Shiva}
   b) {Brahma}
   c) $\varnothing$
   d) {Ganesha}

3. Let *B* = {Paul, Ringo, John, George}.

   List the elements of $\wp$(*B*). How many are there?

4. Again let *B* = {Paul, Ringo, John, George}.

   Mark each of the following as True if it is an element of $\wp$(*B*). Mark it False otherwise.

   a) {Paul, Ringo}
   b) $\varnothing$
   c) {$\varnothing$}
   d) {George, John, Paul}
   e) Ringo

5. Let's see if we can formulate a general claim about the size of a power set.

(Part 1) Let *S* = $\varnothing$. How many elements are there in its powerset?

(Part 2) Let *S* = {a}. How many elements are there in its powerset?

(Part 3) Let *S* = {a, b}. How many elements are there in its powerset?

(Part 4) Let *S* = {a, b, c}. How many elements are there in its powerset?

(Part 5) Let *S* = {a, b, c, d}. How many elements are there in its powerset?

6. Assume any set S of n elements. Which of the following formulas for the size of the powerset of S is consistent with our observations above:

a) $2n$
b) n^2
c) 2^{n+1}
d) 2^n
e) 2^n+1

7. Now that we think we know the size of the powerset of any set of n elements, we should prove our claim. Prove by induction that, if $|S| = n$, for some integer n, then S has 2^n subsets. (Alternatively, $|\wp(S)| = 2^n$.)

To simplify writing this proof, let's define the following notation:

Let $S{:}n$ mean "any set S of n elements".

We know that the first step of this proof is to write an explicit statement of the claim we are trying to prove. We do that by stating the claim in terms of a predicate, $P(n)$. Notice that $P(n)$ needs to make a claim not just about the value n. It needs to make a claim about all sets of size n.

Write out $P(n)$.

Next, we must prove the base case. What value of n do we need to use as the base case?

8. Now we must complete our proof that, if $|S| = n$, for some integer n, then S has 2^n subsets. (Alternatively, $|\wp(S)| = 2^n$.)

Prove the base case.

Now write a proof of the induction step.

8. Prove or disprove the following claim:

[1] $\exists S\ (\wp(S) = \{\varnothing\})$

Once you have your proof and you know whether the claim is true or false, indicate which of these statements is true:

a) [1] is true and there's exactly one set S that has the given property.
b) [1] is true and there are multiple sets that have the given property.
c) [1] is false. There are no sets with the given property.

Partitions

Sometimes it's useful to divide a set *S* into a collection of subsets with the property that every element of *S* occurs in exactly one of the subsets.

Let *S* be the set of students at our school. Then every student is an element of exactly one of these subsets:

- {*Freshmen, Sophomores, Juniors, Seniors, Graduate students*}

We'll call a collection of subsets that has this property a partition. More formally:

A ***partition*** *P* of a set *S* is a set of subsets of *S* such that every element of *S* occurs in exactly one element of *P* and no element of *P* is empty.

- {*Negative integers*, {0}, *Positive integers*} is a partition of **Z** (the integers).
- {*Freshmen, Sophomores, Juniors, Seniors, Graduate students*} is a partition of *Students*.
- {*MathMajors*, *CSMajors*, *EnglishMajors*} is not a partition of *Students*. Some students (ones who are double majoring) are in two of its elements, and some students (e.g., Business majors) are in none of its elements.
- {*Freshmen, Sophomores, Juniors, Seniors, Graduate students, Martians*} is not a partition of *Students*. One of its elements is the empty set.

Problems

1. In each of the following problems, you'll see one set (call it the universe, *U*), then a set of sets. Mark True if the set of sets is a partition of *U*. Mark False otherwise.

a)	**N** (the natural numbers)	{*Primes*, *Composites*}
b)	**Z** (the integers)	{*Evens*, *Odds*}
c)	**Z** (the integers)	{*DivisibleBy3*, *DivisibleBy5*}
d)	*Cars*	{*Gasoline-Cars*, *Electric-Cars*}
e)	*Pets*	{*Cats*, *Dogs*, *Birds*, *Rabbits*}
f)	*Employees*	{*Fulltimes*, *Parttimes*}

Partitions and Proof by Case Enumeration

The technique of proof by Case Enumeration depends on one of the two key properties of a partition: every element of the base set occurs in *some* element of the partition.

Recall our proof of this claim: $\forall n\ (Even(n^2 + n))$

Case 1: n is even. Then $n = 2i$ for some integer i. Then

$$\begin{aligned} n^2 + n &= (2i)^2 + 2i \\ &= 4i^2 + 2i \\ &= 2(2i^2 + i) \quad \text{which is even.} \end{aligned}$$

Case 2: n is odd. Then $n = 2i+1$ for some integer i. Then

$$\begin{aligned} n^2 + n &= (2i+1)^2 + 2i + 1 \\ &= 4i^2 + 4i + 1 + 2i + 1 \\ &= 4i^2 + 6i + 2 \\ &= 2(2i^2 + 3i + 1) \quad \text{which is even} \end{aligned}$$

This proof is correct because {*Evens*, *Odds*} is a partition of **Z** (the integers). Every integer belongs to exactly one of those sets. Thus every integer is covered by one of the cases we've shown.

Partitions and Code for Solving Problems

When we design programs to solve many kinds of interesting problems, we partition the input into cases. Most programming languages provide explicit support for this. Sometimes the construct is called `case`; sometimes it has another name.

Suppose that we're running the alumni affairs office of our university. We are planning to send out a mailing to encourage our alums to come back to campus for homecoming. We might define the following partition on the set *Alumni*:

{*Recentgrads*, *Five-to-TwentyFiveYearsOut*, *Oldergrads*}

Then we might write this program that uses three cases to generate a tailored letter for an individual alum *w*. (Read elif as "else if".)

```
if w in Recentgrads:
      include appealing sections (e.g., concerts)
elif w in Five_to_TwentyFiveYearsOut:
      include appealing sections (e.g., child care)
else:
      include appropriate sections (e.g., back in the day)
```

Problems

1. Suppose that we are designing a direct mail campaign that needs to be tailored for geographic areas. We decide to do this by partitioning the set of U.S. customers based on zipcode. Let *C* be the set of U. S. customers. Let *P* be a partition of *C* such that two elements *x* and *y* of *C* are in the same element of *P* if and only if they have the same five-digit zipcode.

(Part 1) Is the following claim true or false? (Note, we are not asking about who the customers are. Just consider the types of the objects involved.)

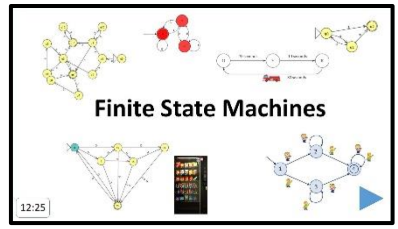
It is possible that Sal Q. Customer is an element of *P*.

(Part 2) Make the strongest statement you can about |*P*|?

a) It cannot be more than about 50.
b) It cannot be more than about 23,000.
c) It cannot be more than about 45,000.
d) It cannot be more than about 125,000.
e) It could be as much as 2,000,000.

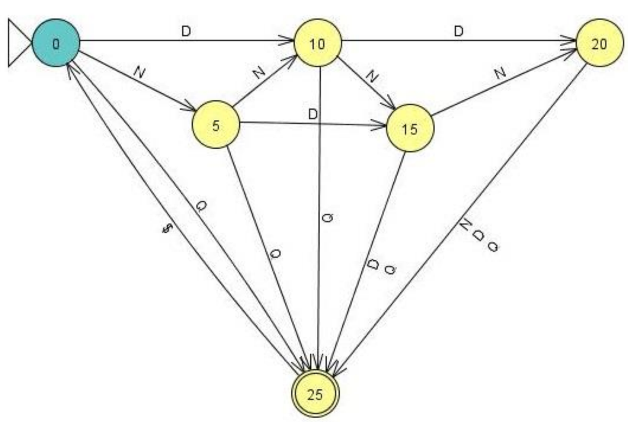
Finite State Machines Partition Inputs

The finite state machine (or FSM) is a formal model of computation. The set of states in such a machine represents a *partition* of the set of machine histories. What happens next depends only on the current state and the next input. This means that, if two different histories have driven the machine to the same state, their differences will have no effect on future behavior.



https://www.youtube.com/watch?v=lh2onWfBrxk

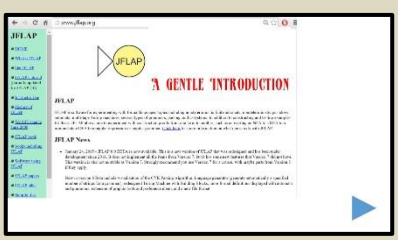
A vending machine is a great example of an FSM. An unbounded number of different machine histories can be partitioned just based on the amount of money (regardless of the specific coins or their order) that has been inserted since the last sale. The machine accepts nickels (N), dimes (D), and quarters (Q). Suppose that it costs 25¢ to buy something.



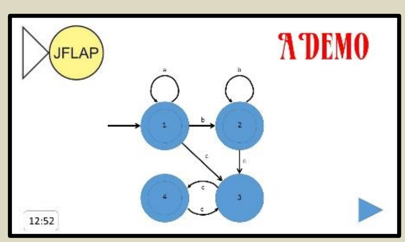
Bu… leposit more than that if you happen not to… uy) button whenever there's at least 25¢ in… ıe resets to 0¢. All of this machine's past "e… 5¢, 20¢, 25¢}.

Nifty Aside

JFLAP is a tool that makes it easy to design and test finite state machines.



https://www.youtube.com/watch?v=irewHV3S0_M



https://www.youtube.com/watch?v=cEvWgEHd0pE

Operations on Sets

Venn Diagrams

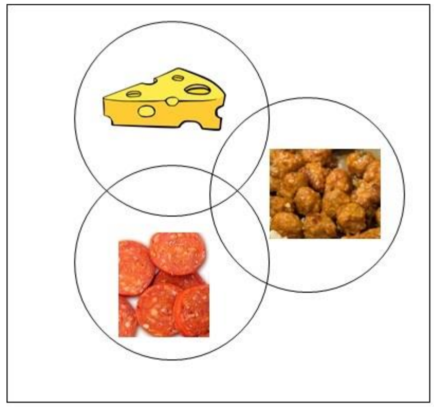
We're about to define some useful operators that can be applied to sets. But, before we do that, let's introduce a graphical way to describe relationships among sets.

A ***Venn diagram*** illustrates the logical relationships among a collection of sets. An outer boundary outlines the universe of discourse. Enclosed regions (often circles or ellipses) correspond to the sets we're considering. Overlaps between and among those regions contain elements that are in more than one of the sets.

Consider this Venn diagram that illustrates kinds of pizza. The box corresponds to the universe, in this case, the set of all pizzas. The circle that is labelled with the wedge of cheese will correspond to pizzas with cheese. Similarly for the circles labelled with pepperoni and sausage.

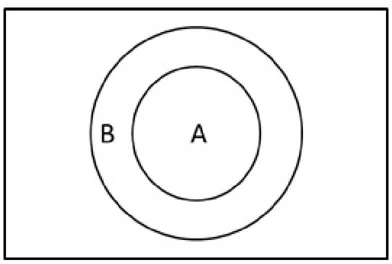
The smallest region, in the middle, corresponds to pizzas with cheese *and* pepperoni *and* sausage.

The region that is outside all the circles corresponds to pizzas that have no cheese or pepperoni or sausage. It should make all our vegan friends happy.

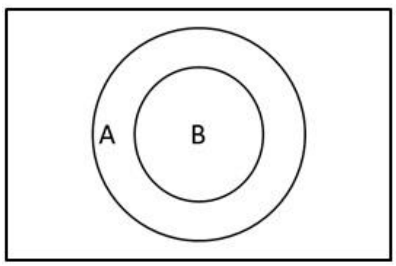


Problems

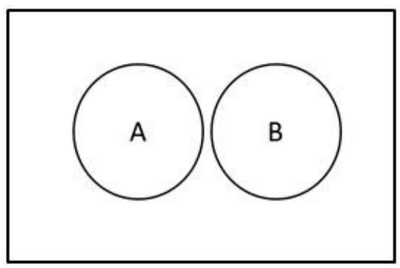
1. Which of the following Venn diagrams illustrates the fact that *A* is a subset of *B*:



(a)



(b)



(c)

Union

Sets are useful because we can do things with them. That means that we need to define operations on them.

We write the ***union*** of two sets A and B as:

$A \cup B$

We define the union of two sets A and B as:

$x \in A \cup B$ if and only if $(x \in A) \vee (x \in B)$

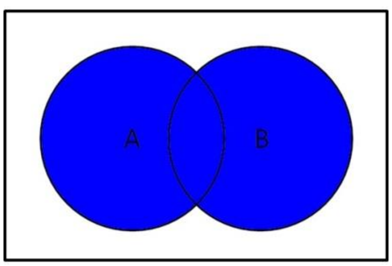
Read this as:

x is in A union B if and only if: x is an element of A *or* x is an element of B.

Recall these definitions:
$\mathbf{Z}$ = the integers
$\mathbf{N}$ = the natural numbers
$\mathbf{Z}^-$ = the negative integers

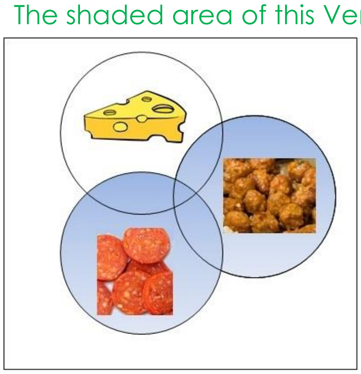
Then: $\mathbf{Z} = \mathbf{N} \cup \mathbf{Z}^-$

We can use a Venn diagram to illustrate union:



The shaded area corresponds to the union of A and B.

The shaded area of this Venn diagram corresponds to pizza for carnivores:



Sausage $\cup$ *Pepperoni*

We can describe the insertion of a new element into a set using the union operator:

Assume that $x \notin S$. Then $S \cup \{x\}$ is the set that results if we insert x into S.

Whenever we add *new* elements to a set, its cardinality changes. In particular:

If $x \notin S$, then $|S \cup \{x\}| = |S| + 1$.

Problems

1. Let: *Band* = {snare drum, tympani, marimba, castanets}.
Orchestra = {piano, tympani}.

What is |*Band* ∪ *Orch*|?

2. Let: *TexasSeasons* = {spring, summer, fall}.

What is |*TexasSeasons* ∪ ∅|?

3. It is possible that there exist sets *A* and *B* and an element *x* such that:

$(x \in A \cup B) \quad \wedge \quad (x \notin A)$

4. It is possible that there exist sets *A* and *B* and an element *x* such that:

$(x \in A) \;\wedge\; (x \notin (A \cup B))$

Intersection

We write the ***intersection*** of two sets A and B as:

$$A \cap B$$

We define the intersection of two sets A and B as:

$x \in A \cap B$ if and only if $(x \in A) \wedge (x \in B)$

Read this as:

x is in A intersect B if and only if: x is an element of A *and* x is an element of B.

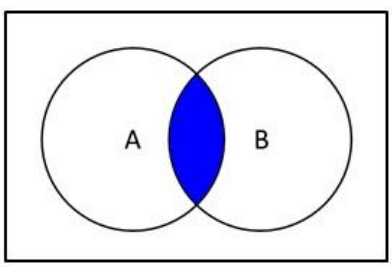
Let *HugeCities* be the set of cities with population over 1 million.

TexasCities $\cap$ *HugeCities* = {Houston, Dallas, San Antonio}

Define: $\mathbf{N}$ = the natural numbers
$\mathbf{Z}^-$ = the negative integers

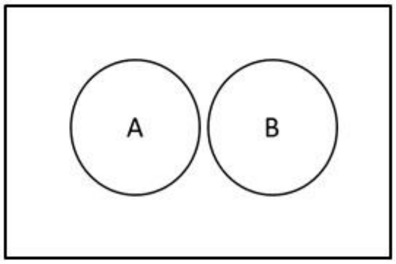
Then: $\mathbf{N} \cap \mathbf{Z}^- = \varnothing$ The intersection of the natural numbers and the negative Integers is empty.

We can use a Venn diagram to illustrate intersection:



The shaded area corresponds to the intersection of A and B. But we should note that, while we draw it this way in the general case, we are not saying that there are necessarily any elements in the set $A \cap B$. It is possible that the intersection is empty.

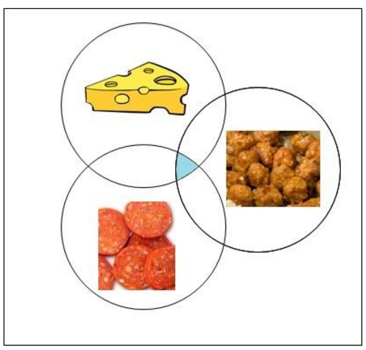
In fact, we'll say that two sets are ***disjoint*** just in case their intersection is empty. If *A* and *B* are known to be disjoint, we can draw this Venn diagram of them:



Let *Evens* = {*n*: *n* is an integer that is divisible by 2}. Let *Odds* = {*n*: *n* is an integer that is not divisible by 2}. Then *Evens* and *Odds* are disjoint since:

$Evens \cap Odds = \varnothing$

The shaded area of this Venn diagram corresponds to "monster protein" pizza:



$Cheese \cap Sausage \cap Pepperoni$

Problems

1. Let: *Band* = {snare drum, tympani, marimba, castanets}.
Orchestra = {piano, tympani}.

   What is $|Band \cap Orch|$?

2. Let: *TexasSeasons* = {spring, summer, fall}.

   What is $|TexasSeasons \cap \varnothing|$?

3. Define two sets:

   - *Primes* = {*n*: *n* is a positive integer greater than 1 that is not evenly divisible by any integer except itself and 1}.
   - *Evens* = {*n*: *n* is an integer that is divisible by 2}.

   What is $Primes \cap Evens$?

   {}
   {1, 2}
   {0, 2}
   {-2, 0, 2}
   {2}

4. Recall that *n* factorial (written *n*!) is:

   $n \cdot (n\text{-}1) \cdot (n\text{-}2) \cdot \ldots \cdot 1$, for any positive integer *n*.

   For example, $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$.

   Let $F = \{k : \exists n\ (k = n!)\}$. In other words, *F* is the set that contains exactly the values of *n*! (for some positive integer *n*).

   Let $P = \{k : \exists n \in \mathbf{N}\ (k = 2^n)\}$. In other words, *P* is the set that contains exactly the values of 2^n, where *n* is some natural number.

   What is $|F \cap P|$? (Hint: See if you can enumerate the elements of $F \cap P$.)

5. It is possible that there exist sets *A* and *B* and an element *x* such that:

   $(x \in A) \land (x \notin (A \cap B))$

Subtraction (Set Difference)

The ***difference*** between two sets *A* and *B* is the set of elements that are in *A* but not in *B*.

There are two standard ways to write the difference between two sets *A* and *B*:

$A - B$ $\qquad\qquad$ $A \setminus B$

We'll use the first one. With that notation, we have that:

$x \in A - B$ $\qquad$ if and only if $\qquad$ $(x \in A) \wedge (x \notin B)$

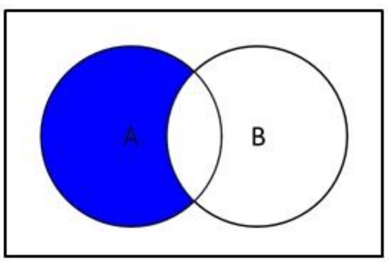
Read this as:

x is in *A* minus *B* if and only if: *x* is an element of *A* *and* *x* is *not* an element of *B*.

An alternative definition that is sometimes more useful is:

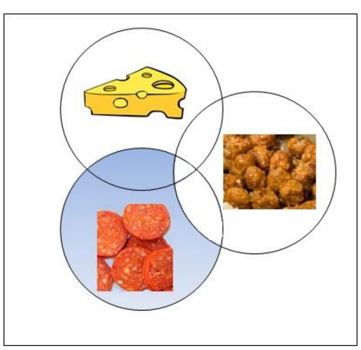
$A - B = A \cap {\sim}B$

We can use a Venn diagram to illustrate set difference:



The shaded area corresponds to the difference between *A* and *B*.

The shaded area of this Venn diagram corresponds to pizza for the pepperoni purist (no sausage allowed):



Pepperoni - Sausage

Notice that the set that we subtract out may contain elements that are not present in the set that we are subtracting from.

Let:	*Class* =	{Angel, Bobby, Dell, Frankie} be the set of students in our class.
	OnTime =	{Angel, Dell, Cody, Jean, Skyler} be the set of students in the whole school who showed up on time today.
Then:	*Class* - *OnTime* = {Bobby, Frankie} is the set of students in our class who were late.	

Problems

1. Define: *Students* = {Angel, Bobby, Chris, Dell, Frankie, Jean, Skyler}
DuesUnpaid = {Chris, Jean}
FailedAClass = {Chris, Skyler}

Now suppose we have: *FieldTripOK* = (*Students* - *DuesUnpaid*) - *FailedAClass*

List the elements of *FieldTripOK*.

2. Consider these two sets:

- P (the powers of 2): $P = \{n : n = 2^k \text{ for some natural number } k\}$.

  So $P = \{1, 2, 4, 8, 16, 32, 64, \ldots\}$.

- F (the elements of the Fibonnaci series). Recall that the series is defined:

  $f_1 = 1$, (The first element is 1.)
  $f_2 = 1$, (So is the next one.)
  for all $k \geq 2$, $f_{k+1} = f_k + f_{k-1}$ (Computed from the two previous ones.)

  So $F = \{1, 2, 3, 5, 8, 13, 21, 34, \ldots\}$.

For each of the following elements v, mark True if $v \in F - P$. Mark False otherwise.

6 1 0 5 4

3. Let: $A = \{x \in \mathbf{Z} : \exists y \in \mathbf{Z}\ (x = 2y)\}$
$B = \{x \in \mathbf{Z} : x < 0\}$

(Part 1) Which of the following sets is an element of $\wp(A) \cap \wp(B)$?

{-1, 0, 1} {2, 4, 8} {1, 2, 3} {-2, -4, -6} {2, 4, 6} {-1, -2, -3}

(Part 2) Which of the following sets is an element of $\wp(A) - \wp(B)$?

{-2, -4, -6} {2, -2} {-1, -2, -3} {1, 2, 3} {-2, -4, -8}

Complement

Suppose that we are willing to fix a set U that we'll call the ***universe***. Then we can talk about the ***complement*** of a set A, which we'll define to be the set of elements that are elements of U but *not* of A.

There are four fairly common notations:

$\neg A \qquad \sim A \qquad \overline{A} \qquad A'$

We will use the second of these. With that notation, we have that:

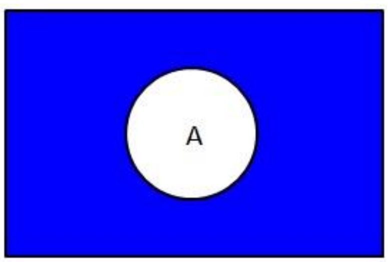
$x \in \sim A$ if and only if $(x \in U) \wedge (x \notin A)$

Read this as:

x is in the complement of A if and only if:

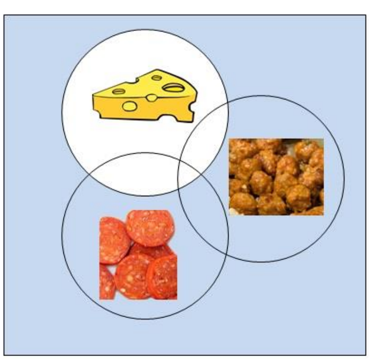
x is an element of the universe U *and* x is not an element of A.

We can use a Venn diagram to illustrate complement. As we've been doing all along, the outer rectangle corresponds to the universe:



The shaded area corresponds to $\sim A$.

Let the universe be the set of all pizzas. The shaded area of this Venn diagram corresponds to pizza for the lactose intolerant.



~Cheese

In order to use complement in a useful way, we must start by carefully choosing U.

Let *TexasSeasons* = {spring, summer, fall}. We probably want to define the universe so that we get:

winter $\in$ ~*TexasSeasons*

But it's hard to imagine a real problem in which we'd want to have:

artichoke $\in$ ~*TexasSeasons*

So, let *Seasons* = {winter, spring, summer, fall}. Let our universe be *Seasons*. Then we have:

~*TexasSeasons* = {winter}

So winter is the only element of ~*TexasSeasons*. artichoke $\notin$ ~*TexasSeasons*. True, it's not in *TexasSeasons*. But to be in the complement of *TexasSeasons*, it would have to be in the universe *Seasons*. But it isn't.

Notice that there's an obvious relationship between complement (with respect to some previously declared universe U) and set difference (where we specify both sets explicitly). For any set A:

$$\sim A = U - A$$

When we're working with a single universe (for example, the integers), it is often clearer simply to specify U and then use complement when we need it. If we are working with multiple sets from which we wish to subtract, it is clearer to use set difference than to attempt to restate U and keep it correct.

Recall the factorial set $F = \{k : \exists n\ (k = n!)\}$. In other words, F is the set that contains exactly the values of $n!$ (for some positive integer n). So $F = \{1, 2, 6, 24, 120, \ldots\}$.

Let $U = \mathbf{N}$ (the natural numbers). Then:

$\sim F = \{0, 3, 4, 5, 7, 8, 9, 10, 11, \ldots\}$

Now let $U = \mathbf{Z}$ (the integers). Then (recall that $\mathbf{Z}^-$ is the set of negative integers):

$\sim F = \mathbf{Z}^- \cup \{0, 3, 4, 5, 7, 8, 9, \ldots\}$

The complement of F now contains all the negative integers *and* all the natural numbers that aren't values of the factorial function.

Using complement is risky unless we have been crystal clear in defining the universe with respect to which we are taking the complement. Failure to do that can lead to serious ambiguity.

English Aside

The meaning of the word "only" includes a negation (complement), even though the

Take a look at the sign shown above the door in this picture. It says, "SERVICE ANIMALS ONLY".

Who (or what) is actually allowed to use the door to get into the barbeque restaurant? Are customers allowed? Surely very few customers are service animals. Even the customers with service animals are not themselves service animals. Surely they're not banned.

Expressions that exploit complements (negations) are often ambiguous if they fail to make clear exactly what the universe is.

We can read this sign as defining a set of banned values:

Banned = *~ServiceAnimals*

But what set should we use as the universe with respect to which we compute *~ServiceAnimals*?

Suppose we have:

U = absolutely everything

Then we also have:

$\in$ *Banned* and $\in$ *Banned*

Suppose, on the other hand, that we have:

U = *NonhumanAnimals*

Then we get:  $\notin$ *Banned*  $\notin$ *Banned*  $\in$ *Banned*

This is probably what the sign writers intended.

Problems

1. Define: *Evens* = $\{n : \exists k \in \mathbf{Z}\ (n = 2k)\}$.
Odds = $\{n : \exists k \in \mathbf{Z}\ (n = 2k + 1)\}$.

Suppose that we want to be able to say that *Evens* = ~*Odds*. What universe *U* must we take complement with respect to in order for this to be true:

a) $U = \mathbf{N}$.
b) $U = \mathbf{Z}$.
c) $U = \mathbf{R}$.
d) *U* = *Odds*.
e) *U* = *Evens*.

2. Let $P = \{1, 2, 4, 8, 16, 32, 64, \ldots\}$. In other words, $P = \{n : n = 2^k$ for some natural number $k\}$. Now consider some larger set that is the universe with respect to which we can define the complement of *P* (~*P*). Call this universe *U*.

Suppose that we are told:

$6 \in {\sim}P$ $\quad$ $3 \notin {\sim}P$ $\quad$ $10 \in {\sim}P$ $\quad$ $-2 \notin {\sim}P$

Which of the following definitions of *U* is consistent with what we've been told:

a) **Z** (integers)
b) **N** (natural numbers)
c) *Evens*
d) *Evens* ∩ **N**
e) *Evens* ∪ **N**

Another Way to Define Set Difference

So far, we've defined set difference in terms of a typical element:

$x \in A - B$ if and only if $(x \in A) \wedge (x \notin B)$

Now that we've defined set complement, we can offer an alternative (and easier to use in some situations) definition of set difference. If $x \in U$ (Universe), then:

$x \in A - B$ if and only if $(x \in A) \wedge (x \in \sim B)$

So, using the definition of intersection:

$$A - B = A \cap \sim B$$

Problems

1. Which of the following sets must, for all A, B and C, be equal to $C - (B - A)$:

a) $C \cup (B - A)$
b) $C \cap (A - B)$
c) $C \cap \sim(B - A)$ Correct
d) $C \cap (\sim B \cup \sim A)$
e) $C \cap (\sim B \cap \sim A)$

Insertion and Deletion

When we work with sets, particularly in programs, we often insert and remove (delete) elements one at a time.

> Suppose that *Friends* is the set of Riley's Facebook friends. Riley accepts a friend request from Sage. Facebook needs to add Sage to *Friends*.

Because insertion and deletion are so useful, we'll define them explicitly. Notice, in these definitions, that *Insert* and *Delete* actually change the value of the set that they are given.

Insert(a, S) assigns the new value $S \cup \{a\}$ to S.

Recall that, if $a \notin S$, then $|S \cup \{a\}| = |S| + 1$.

Remove(a, S) assigns the new value $S - \{a\}$ to S. While this set expression is well defined regardless of whether a was initially in S, some implementations of this operation in real programs raise an error if a was not present in S.

> Suppose that *Professors* = {Albus Dumbledore, Severus Snape, Pomona Sprout}.
>
> Then Dolores Umbridge arrives, so the Hogwarts personnel office does this:
>
> *Insert*(Dolores Umbridge, *Professors*)
>
> At this point, *Professors* = {Albus Dumbledore, Severus Snape, Pomona Sprout, Dolores Umbridge}.
>
> Then Pomona Sprout retires. So they do:
>
> *Remove*(Pomona Sprout, *Professors*)
>
> At this point, *Professors* = {Albus Dumbledore, Severus Snape, Dolores Umbridge }.

Problems

1. Suppose that we start with an empty set *S*. We insert these elements (in this order):

5, 8, 1, 9, 5, 6

Mark each of the following as true if it describes the value of *S* after the six insertions:

I. {8, 1, 9, 5, 6}
II. {5, 8, 1, 9, 5, 6}
III. {5, 6, 8, 1, 9}

Summary of Set Operations

We've now defined ~ as an operation on a single set, and $\cap$, $\cup$, and - as operations on pairs of sets. But we often want to work with more than two sets. No problem. We just combine them.

> Suppose that our Human Relations department needs to send out somewhat nasty reminder messages to everyone who is late submitting their time sheets. They might describe the set of people who will get the message as:
>
> (*FullTimeEmployees* ∪ *PartTimeEmployees* ∪ *Interns*) - *PromptTimeSheetSubmitters*

> I have a meeting in the morning with the sales rep for Peacock Corp. I need to pull together my order for them. Here it is:
>
> ((*ThingsWe'reOutOf* ∪ *ThingsWe'reAlmostOutOf*) ∩ *ThingsWeBuyFromPeacock*) ∪ *NewStuff*

Notice how similar these examples feel to some of the ones we looked at when we were discussing logical expressions.

> Returning to the time sheet example: We could have described the letter receivers as:
>
> ∀x ((*FullTimeEmployee*(x) ∨ *PartTimeEmployee*(x) ∨ *Intern*(x)) ∧ ~*PromptTimeSheet*(x))
>
> ≡ *NastyLetter*(x))

What we notice is that:

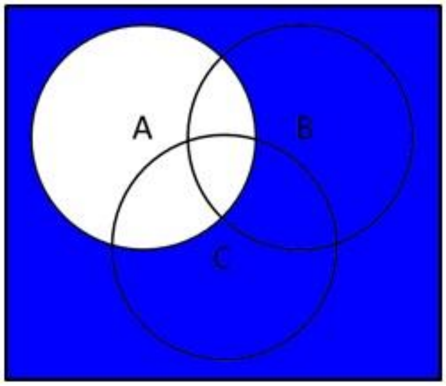
- Set complement corresponds to logical *not*.
- Set union corresponds to logical *or*.
- Set intersection corresponds to logical *and*.
- Set difference corresponds to logical *and not*.

We'll come back to this point soon. As we'll see, this correspondence is deep and it is the basis for many of our most useful ways of reasoning about sets.

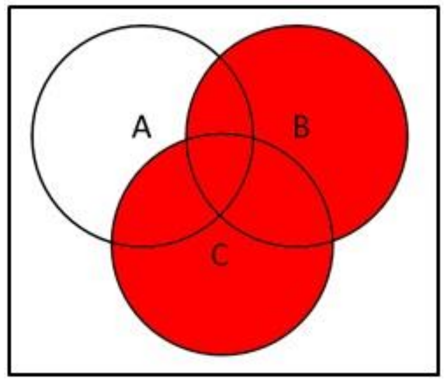
Venn Diagrams for Larger Expressions

Venn diagrams can be useful tools for working with expressions of three sets. Unfortunately, it becomes quite tricky to use them for more than three sets.

Suppose that we want to visualize ~A ∩ (B ∪ C). We can first build these two Venn diagrams (using two different colors):

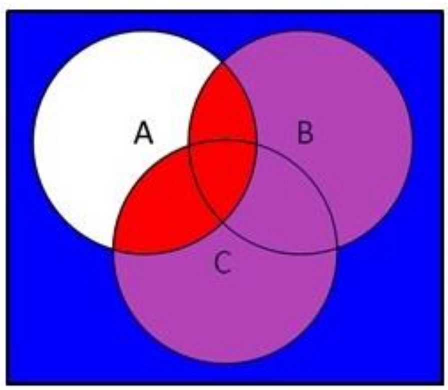


¬A



(B ∪ C)

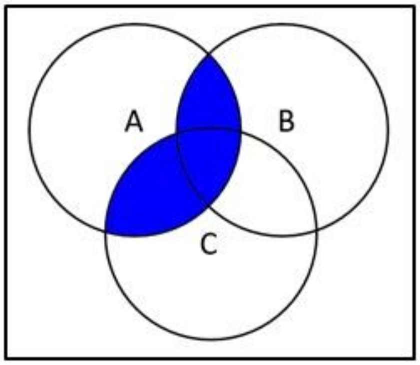
As a final step, we combine them. The best way to do that is to imagine laying one on top of the other. When we do that, we get:



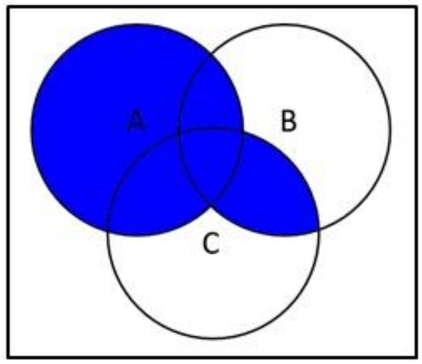
The region that has been colored both red and blue (purple) corresponds to ~A ∩ (B ∪ C).

Problems

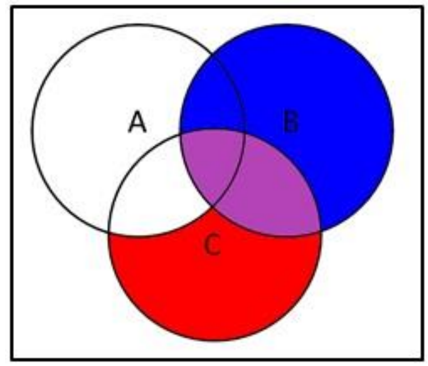
1. Consider the set $S = A \cup (B \cap C)$. Which of the following Venn diagrams corresponds to S: (If there are two colors, view the answer as the part that contains both of them. In particular, view purple as the combination of red and blue.)



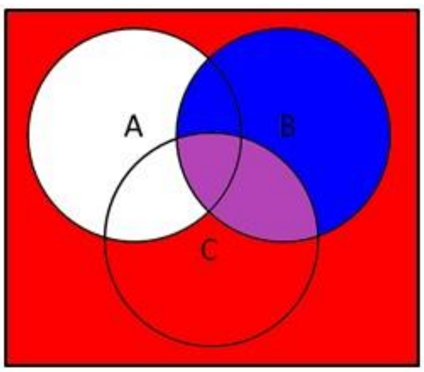
(a)



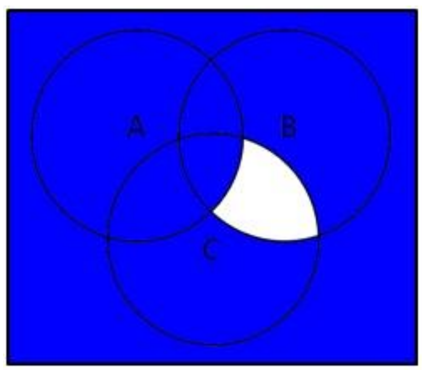
(b)



(c)

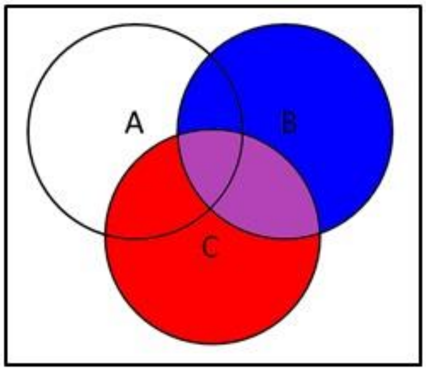


(d)

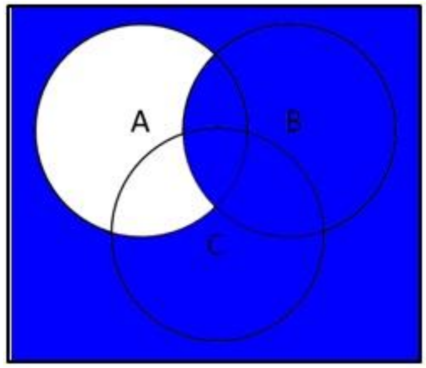


(e)

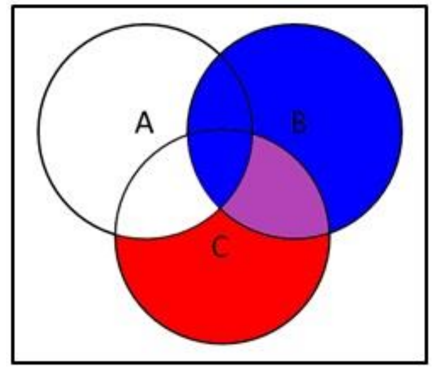
2. Consider the set $S = B \cap (C - A)$. Which of the following Venn diagrams corresponds to S: (If there are two colors, view the answer as the part that contains both of them. In particular, view purple as the combination of red and blue.)



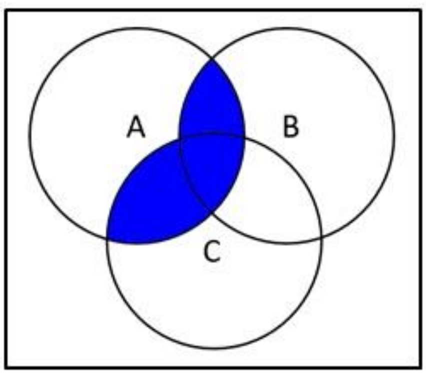
(a)



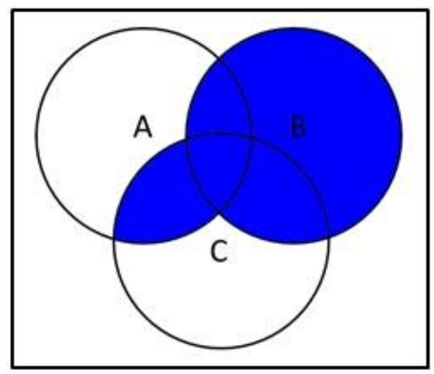
(b)



(c)

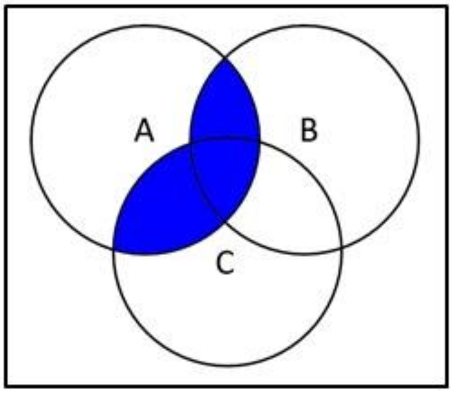


(d)



(e)

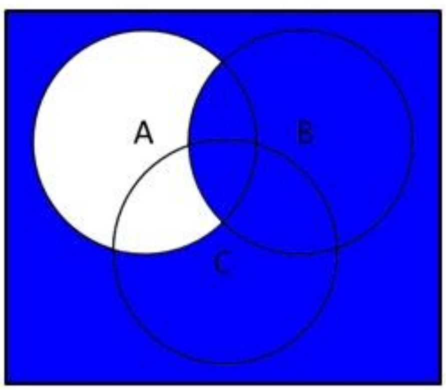
3. Consider this Venn diagram:



Which of the following set expressions corresponds to the blue region:

a) $C \cap (A \cup B)$
b) $A \cap (B \cup C)$
c) $B \cap (A \cup C)$
d) $A \cap (B \cap C)$
e) $C \cup (B \cap A)$

4. Consider this Venn diagram:



Which of the following set expressions corresponds to the blue region:

a) $(\sim A \cup B) \cup C$
b) $(C - A) \cup B$
c) $\sim A \cup B$
d) $(C \cup B) - A$
e) $\sim A \cap B$

Operator Precedence

Recall that ***operator precedence*** defines the order in which operations within an expression will be performed when there are not parentheses. Higher precedence operators are done before lower precedence ones.

There's no general consensus on the use of different precedence levels for intersection, union and difference, so we'll assign equal precedence to the three of them.

We've seen the first three columns in this table before. What we've done now is to add a fourth column that specifies the precedence of the set operators that we've just defined.

	Arithmetic	Logic	Sets
Highest:	*unary minus*	*not*	~
	exponentiation	*and*	$\cap$, $\cup$, -
	* and /	*or*	
	+ and -	*implies*	$\supset$, $\supseteq$, $\subset$, $\subseteq$, $\in$, $\notin$
Lowest:		*is equivalent to*	

If there are multiple operators with the same precedence level, then we associate left to right.

So we interpret:

$A \cap B \cup C$ as $(A \cap B) \cup C$

Big Idea

Use parentheses to be sure.

Problems

1. Consider the expression:

$\sim A \cap B \subseteq C \cap D \cup S$

Which of the following parenthesizations corresponds to the interpretation defined above:

a) $((\sim A) \cap (B \subseteq C)) \cap (D \cup S)$
b) $((\sim A) \cap B) \subseteq ((C \cap D) \cup S)$
c) $\sim(A \cap B) \subseteq ((C \cap D) \cup S)$
d) $((\sim A) \cap B) \subseteq (C \cap (D \cup S))$
e) $\sim(A \cap B) \subseteq (C \cap (D \cup S))$

The Natural Analogy between Sets and Logic

The Big Idea

Whenever we write a logical expression, like $P(x)$, we've actually defined a set: those objects that satisfy the predicate P.

Coming from the other direction, whenever we define a set $S = \{ \ldots. \}$, we have effectively also defined a predicate S_P that is true of all and only the elements in S.

Thus it should come as no surprise that there is a natural analogy between the ways that we reason in logic and the ways that we reason with sets.

Recall our problem of sending nasty letters to time sheet scofflaws. We wrote this set expression to describe the people who will get letters:

$$(\mathit{FullTimeEmployees} \cup \mathit{PartTimeEmployees} \cup \mathit{Interns}) - \mathit{PromptTimeSheetSubmitters}$$

We also described the same situation as a logical expression:

$$\forall x\ ((\mathit{FullTimeEmployee}(x) \vee \mathit{PartTimeEmployee}(x) \vee \mathit{Intern}(x)) \wedge {\sim}\mathit{PromptTimeSheet}(x))$$

$$\equiv \mathit{NastyLetter}(x))$$

Specifically, we'll see (by looking at the definitions of the logical operators) that:

- Set complement corresponds to logical *not*. $\sim$ / $\neg$
- Set union corresponds to logical *or*. $\cup$ / $\vee$
- Set intersection corresponds to logical *and*. $\cap$ / $\wedge$
- Set difference corresponds to logical *and not*. $-$ / $\wedge\neg$
- Set subset corresponds to logical *implies*. $\subseteq$ / $\rightarrow$

Big Idea

The natural correspondence between logical and set operations gives us a powerful tool for reasoning about sets.

Problems

1. Let *Fluffies* = {kitten, puppy, bunny, chick}. Define *Fluffy*(x) to be true just in case $x \in$ *Fluffies*. True or false:

a) *Fluffy*(bunny) is true.
b) *Fluffy*(duckling) is true.
c) ~*Fluffy*(kitten) is true.
d) *Fluffy*(chick) $\wedge$ *Fluffy*(piglet) is true.
e) *Fluffy*(bunny) $\vee$ *Fluffy*(duckling) is true.

2. Let *DemocraticPresidents* = {x : *USPresident*(x) $\wedge$ *Democrat*(x)}. True or false:

a) Andrew Jackson $\in$ *DemocraticPresidents*
b) Ulysses Grant $\in$ *DemocraticPresidents*
c) Alexander Hamilton $\in$ *DemocraticPresidents*
d) Dwight Eisenhower $\in$ *DemocraticPresidents*
e) Woodrow Wilson $\in$ *DemocraticPresidents*

Set Operations / Logical Operations

Assume some fixed universe U. Let:

- S_P be the *set* of values of which some predicate P is true.
- S_Q be the *set* of values of which some predicate Q is true.

Union / Or

The definition of union: $x \in \{S_P \cup S_Q\}$ if and only if $x \in \{S_P\} \vee x \in \{S_Q\}$

So: $S_P \cup S_Q = \{x : P(x) \vee Q(x)\}$

We could define the set of healthy foods as:

Healthy = *Fruits* ∪ *Vegetables* = {x : *Fruit*(x) ∨ *Vegetable*(x)}

Intersection / And

The definition of intersection: $x \in \{S_P \cap S_Q\}$ if and only if $x \in \{S_P\} \wedge x \in \{S_Q\}$

So: $S_P \cap S_Q = \{x : P(x) \wedge Q(x)\}$

Our friend, a top chef, thinks that food must be fantastic to eat and stunning to look at. So she describes the foods she'll serve at her restaurant as:

MyFoods = *Yummies* ∩ *Beautifuls* = {x : *Yummy*(x) ∧ *Beautiful*(x)}

We need to be careful when we're moving back and forth between the set operations *union*/*intersection* and the logical operations *or*/*and*, particularly if we're starting with English expressions.

In English, we might paraphrase this bus sign as, "Seniors *and* disabled people are welcome to sit here."

Using set notation, we can describe the set of welcome sitters as:

Seniors ∪ *Disabled*

It appears that the set operation *union* corresponds to English "and".

But we write the same claim in logic using *or*:

$\forall x\ ((Senior(x) \vee Disabled(x)) \rightarrow CanSitHere(x))$

This corresponds to the other way we could have described the situation in English: "If you are a senior *or* you are disabled, you are welcome to sit here."

In English, when we're thinking about sets, we use "and" to mean *union*. Of course, when we're thinking about logical predicates, we use "or" to mean *or*. But we've just seen that set *union* and logical *or* correspond. So, sadly, in English, it may be possible to use both "and" and "or" to mean the same thing. Now, at least, you know why.

Complement / Complement

Notice that this one must consider the universe *U*.

The definition of complement: $x \in {\sim}S_P$ if and only if $(x \in U) \wedge (x \notin S_P)$

So: ${\sim}S_P = \{x \in U : \neg P(x)\}$

Timmy, in the throes of the terrible twos, know exactly what food is. Let *Moms* be the set of things his Mom wants him to eat and let *MomApproves*(x) be true just of elements of *Moms*. Let *Small* be the set of things small enough to fit into his mouth. Then Timmy's notion of food can be described as follows with respect to the universe *Small*:

$Food = {\sim}Moms = \{x \in Small: \neg Mom(x)\}$

Subtraction (Set Difference) / And Complement

Now the explicitly stated set S_P plays the role that the universe *U* did when we defined complement.

The definition of set difference: $x \in S_P - S_Q$ if and only if $(x \in S_P) \wedge (x \notin S_Q)$

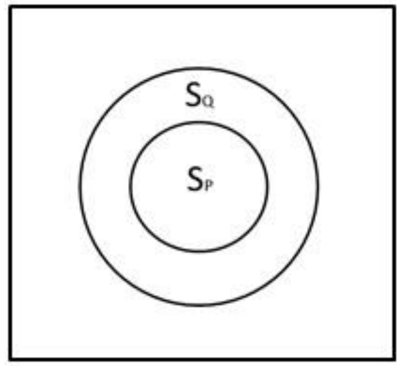
So: $S_P - S_Q = \{x : P(x) \wedge \neg Q(x)\}$

Consider the food options available to a poverty-stricken student who is trying, nevertheless, to eat healthy food. Let *Wholes* be the set of foods available at the local organic foods store and let *Whole*(x) be true just in case that store sells x. Let *Crazies* be the set of outrageously priced foods and let *CrazyPrice*(x) be true just in case the price of x is outrageous. Then our hapless student's options correspond to this (sadly, very small) set:

$StudentOptions = Wholes - Crazies = \{x : Whole(x) \wedge \neg CrazyPrice(x)\}$

Subset / Implies

Recall the subset relationship, which we can illustrate with a Venn diagram:



The definition of subset: $S_P \subseteq S_Q$ if and only if $\forall x\,((x \in S_P) \rightarrow (x \in S_Q))$

So: $S_P \subseteq S_Q \equiv \forall x\,(P(x) \rightarrow (Q(x))$

Notice that, in this case, both expressions are logical ones (i.e., they have truth value), so we'll say that they are equivalent.

> Recall that **Z** is the set of integers and **R** is the set of Reals. So:
>
> Since $\mathbf{Z} \subseteq \mathbf{R}$, we have that $\forall x\,(Integer(x) \rightarrow (Real(x))$

Suppose that $S_P \subset S_Q$ (i.e., S_P is a *proper* subset of S_Q). Then we make a *stronger* claim when we say that $x \in S_P$ than we make when we say that $x \in S_Q$. You can think of it as smaller circles make stronger claims (because fewer elements qualify).

Problems

1. Let $S = (Fruits \cup Vegetables) - RedThings$. Then we can say:

$x \in S$ if and only if ____.

Which of the following logical expressions goes into the blank:

a) $Fruit(x) \vee Vegetable(x) \vee \neg Red(x)$
b) $(Fruit(x) \vee Vegetable(x)) \wedge \neg Red(x)$
c) $(Fruit(x) \wedge Vegetable(x)) \vee \neg Red(x)$
d) $(Fruit(x) \vee Vegetable(x)) \vee Red(x)$
e) $Fruit(x) \wedge Vegetable(x) \wedge \neg Red(x)$

2. Consider the classic (chivalrous) idea:

Circle all of the following expressions (some logical, some sets) that are consistent with the intent of our sign:

I. $Firsts = Women \cup Children$
II. $Firsts = Women \cap Children$
III. $\forall x\, ((Child(x) \wedge Woman(x)) \rightarrow First(x))$
IV. $\forall x\, ((Child(x) \vee Woman(x)) \rightarrow First(x))$
V. $\forall x\, ((\neg Child(x) \wedge \neg Woman(x)) \rightarrow \neg First(x))$

3. Assume a universe of people in our club. Suppose that you're told:

[1] $MyFriends \subseteq (EmeryFriends \cap LandryFriends) - SawyerFriends$

(Part 1) Consider the logical expression:

[2] $\forall x\, (\neg EmeryFriend(x) \rightarrow \neg MyFriend(x))$

Which of the following is true:

a) [2] is equivalent to [1].
b) [2] must be true if [1] is.
c) [2] is consistent with [1] although not guaranteed by it.
d) [2] is inconsistent with [1]

(Part 2) Consider the logical expression:

[3] $\forall x\ (\neg SawyerFriend(x) \rightarrow MyFriend(x))$

Which of the following is true:

a) [3] is equivalent to [1].
b) [3] must be true if [1] is.
c) [3] is consistent with [1] although not guaranteed by it.
d) [3] is inconsistent with [1]

(Part 3) Consider the logical expression:

[4] $\exists x\ (MyFriend(x) \wedge EmeryFriend(x) \wedge SawyerFriend(x))$

Which of the following is true:

a) [4] is equivalent to [1].
b) [4] must be true if [1] is.
c) [4] is consistent with [1] although not guaranteed by it.
d) [4] is inconsistent with [1]

(Part 4) Consider the logical expression:

[5] $\forall x\ (EmeryFriend(x) \rightarrow MyFriend(x))$

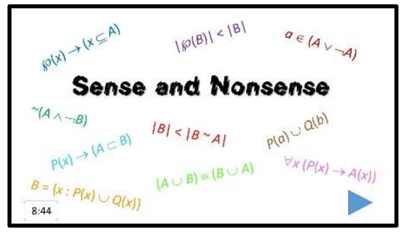
Which of the following is true:

a) [5] is equivalent to [1].
b) [5] must be true if [1] is.
c) [5] is consistent with [1] although not guaranteed by it.
d) [5] is inconsistent with [1]

Making Sure that We Don't Write Nonsense

At this point, we will often find ourselves working with several different ***types*** of things, including:

- Primitive objects like numbers, people and courses.
- Sets of objects.
- Logical statements.



https://www.youtube.com/watch?v=hrzMkiBuNgU

It's important, if we don't want to spout nonsense, that we use all of these things only in ways that are defined for their types.

> ***Nifty Aside***
>
> Modern programming languages also make extensive use of the notion of type. Operations are defined for one or more specific types and don't make any sense when applied to objects of other types. For example, this is nonsense Python code:
>
> ```
> "abc" - "ab"
> ```
>
> The problem is that subtraction isn't defined for objects of type string.

Consider the expression: $x \in A \wedge B$

It's nonsense. If *A* and *B* are logical expressions, then so is $A \wedge B$. So far so good. But logical expressions don't have elements. If, on the other hand, *A* and *B* are sets, then *and* ($\wedge$) can't be applied to them. It is defined only for logical expressions.

If someone wrote this, they might perhaps have been trying to say that x is in both *A* and *B*. In that case, they could correctly have written either of these expressions:

$x \in A \wedge x \in B$
$x \in A \cap B$

Problems

1. Mark each of the following claims as Sense if it is well-defined and Nonsense if it is not well-defined. (Note that we're not actually asking whether or not the claim is True. Just whether it is syntactically well-formed.) Assume that *a* and *b* are primitive objects, *A* and *B* are sets and *P* and *Q* are logical predicates.

a) $(A \wedge B) \subseteq (A \vee B)$
b) $\forall x\ (P(x) \rightarrow (x \in A))$
c) $(a \in A) \rightarrow P(a \vee b)$
d) $(A \subseteq B) \rightarrow \forall x\ (P(x))$
e) $a \in (A \subset B)$
f) $(a \in A) \vee (a \subseteq B)$

Proving Claims about Sets I

Checking Proofs about Sets

StepWise (the proof checking tool that we used for Boolean and predicate logic) also works for proofs about sets.

You will notice that, as we introduce the identities and inference rules that we'll use for reasoning about sets, those things will show up as allowable justifications in StepWise.

Convert the Set Problem to a Logic Problem

Recall that we have defined all the set operations in terms of logical ones.

For example: The definition of intersection: $x \in A \cap B$ if and only if $(x \in A) \wedge (x \in B)$

The structure of these definitions suggests a powerful way to reason about sets:

1. Convert the set problem to a logical one by exploiting the definitions of the set operators.
2. Reason in logic.
3. Convert the result back to a set expression, again using the definitions of the set operators.

The details of how we do this depend on what we are trying to prove.

Suppose that we want to prove that two sets A and B are equal. One approach is to prove that the claim that some element x is in A is logically equivalent to the claim that it is in B.

We will proceed as follows:

1. We'll take as a conditional premise the claim that x is in A.

2. Using the convert-to-logic technique that we've just described, we'll then prove that, if x is in A, then it must also be in B. We will be careful to use only logical identities and computation as we do this. (In other words, we use only reversible rules.)

3. We observe that, since the proof we just did is reversible, we have also proved the other direction, namely that, if $x \in B$, then $x \in A$.

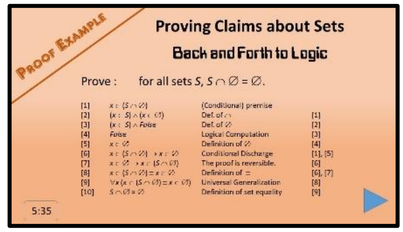
4. Having proved both directions, we have that $x \in A \equiv x \in B$.

5. Now we use Universal Generalization to generalize from a claim about an arbitrary x to a claim about any x. We then have: $\forall x\ (x \in A \equiv x \in B)$.

6. Finally, we appeal to the definition of set equality to assert that, since A and B contain the same elements, $A = B$.

Prove that, for all sets S, $S \cap \varnothing = \varnothing$.

For any set S and any arbitrary element x of S:



https://www.youtube.com/watch?v=QR6akpA-FYA

[1]	$x \in (S \cap \varnothing)$	(Conditional)premise	
[2]	$(x \in S) \wedge (x \in \varnothing)$	Def. of $\cap$	[1]
[3]	$(x \in S) \wedge$ *False*	Def. of $\varnothing$	[2]
[4]	*False*	Logical Computation	[3]
[5]	$x \in \varnothing$	Definition of $\varnothing$	[4]
[6]	$x \in (S \cap \varnothing) \rightarrow x \in \varnothing$	Conditional Discharge	[1], [5]
[7]	$x \in \varnothing \rightarrow x \in (S \cap \varnothing)$	The proof is reversible.	[6]
[8]	$x \in (S \cap \varnothing) \equiv x \in \varnothing$	Definition of $\equiv$	[6], [7]
[9]	$\forall x\ (x \in (S \cap \varnothing) \equiv x \in \varnothing)$	Universal Generalization	[8]
[10]	$S \cap \varnothing = \varnothing$	Definition of set equality	[9]

In this very simple proof, we only needed one purely logical step (the one that derived [4]). Then we were ready use set definitions to go back the other way. We were able to go from [4] to [5] by observing that the claim that $x \in \varnothing$ is a contradiction (since no elements are in $\varnothing$). Thus it is equivalent to *False*.

As we write our proofs, when there is no confusion, we will focus on the logic of each individual proof. So we may omit explicit mention of the steps that are numbered [6] – [10] above. In particular, when we do proofs using our proof checker, we will skip those steps since they are the same in every proof.

Problems

1. Prove that, for all sets S, $S - \varnothing = S$

Write a proof that holds for any set S and any arbitrary element x of S.

Hint: You should be able to simplify and use Boolean computation to get the desired result.

Set Identities

While we can always reason about sets by converting set claims to logical ones and then reasoning in logic, it can be useful to be able to reason more directly about sets themselves.

Fortunately the natural analogy between set expressions and logical ones suggests a collection of set identities analogous to the logical ones that we have already described and proved. Recall the analogies that we've presented:

- Set complement corresponds to logical *not*. $\sim$ / $\neg$
- Set union corresponds to logical *or*. $\cup$ / $\vee$
- Set intersection corresponds to logical *and*. $\cap$ / $\wedge$
- Set difference corresponds to logical *and not*. - / $\wedge\neg$
- Set subset corresponds to logical *implies*. $\subseteq$ / $\rightarrow$

We present the set identities here. Then we'll show how they can be proved. Most of the proofs will be left as exercises.

Let *A*, *B*, and *C* be arbitrary sets. Let *U* be the universe.

As we write expressions involving sets, we'll see that sometimes we write expressions whose values are sets.

$(A \cap B)$ and $\sim A \cap (B \cap C)$ are expressions whose value is a set.

But sometimes we write logical expressions about sets. In this case, the value of the expression is not a set; it is *True* or *False*.

$(A = B)$ and $(A \subseteq B)$ are logical expressions. They are either true or false.

As we have been doing, we'll use = to indicate that two *sets* are equal (i.e., that they contain the same elements). We'll use $\equiv$ to indicate that two *logical expressions* are equivalent (i.e., that they have the same truth values).

For each of these identities, we'll show the logical version on the left, then the set version, in red, on the right.

Double Negation

$$p \equiv \neg(\neg p) \qquad\qquad A = \sim(\sim A)$$

Equivalence

$$(p \equiv q) \equiv (p \rightarrow q) \wedge (q \rightarrow p) \qquad\qquad (A = B) \equiv (A \subseteq B) \wedge (B \subseteq A)$$

Idempotence

$(p \wedge p) \equiv p$ $\qquad (A \cap A) = A$

$(p \vee p) \equiv p$ $\qquad (A \cup A) = A$

De Morgan$_1$

$(\neg(p \wedge q)) \equiv (\neg p \vee \neg q)$ $\qquad \sim(A \cap B) = \sim A \cup \sim B$

De Morgan$_2$

$\neg(p \vee q) \equiv (\neg p \wedge \neg q)$ $\qquad \sim(A \cup B) = \sim A \cap \sim B$

Commutativity of *or* — **Commutativity of Union**

$(p \vee q) \equiv (q \vee p)$ $\qquad (A \cup B) = (B \cup A)$

Commutativity of *and*: — **Commutativity of Intersection**

$(p \wedge q) \equiv (q \wedge p)$ $\qquad (A \cap B) = (B \cap A)$

Associativity of *or*: — **Associativity of Union**

$(p \vee (q \vee r)) \equiv ((p \vee q) \vee r)$ $\qquad A \cup (B \cup C) = (A \cup B) \cup C$

Associativity of *and*: — **Associativity of Intersection**

$(p \wedge (q \wedge r)) \equiv ((p \wedge q) \wedge r)$ $\qquad A \cap (B \cap C) = (A \cap B) \cap C$

Distributivity of *and* over *or*: — **Distributivity of Intersection over Union**

$(p \wedge (q \vee r)) \equiv ((p \wedge q) \vee (p \wedge r))$ $\qquad A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Distributivity of *or* over *and*: — **Distributivity of Union over Intersection**

$(p \vee (q \wedge r)) \equiv ((p \vee q) \wedge (p \vee r))$ $\qquad A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Conditional Disjunction:

$(p \rightarrow q) \equiv (\neg p \vee q)$ $\qquad (A \subseteq B) \equiv ((\sim A \cup B) = U)$

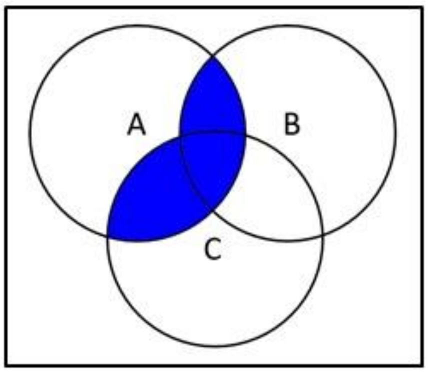
Contrapositive:

$(p \rightarrow q) \equiv (\neg q \rightarrow \neg p)$ $\qquad (A \subseteq B) \equiv (\sim B \subseteq \sim A)$

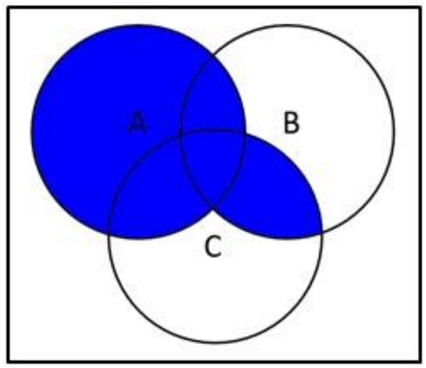
Of these, the one that is perhaps not obvious is Conditional Disjunction. Recall that if we say $p \rightarrow q$, we have equivalently said that at least one of $\neg p$ or q must, in all cases, be true. So, if being in A implies being in B, we have an equivalent claim that every element must be in at least one of $\sim A$ or B. Thus the two of them ($\sim A$ and B) together account for all of the universe U.

Problems

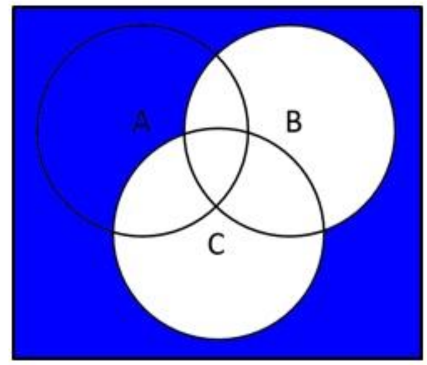
1. Consider the set $S = A \cup {\sim}(B \cap C)$. Which of the following Venn diagrams corresponds to S: Hint: Use the set identities to transform the description of S into something that more obviously matches one of these diagrams.



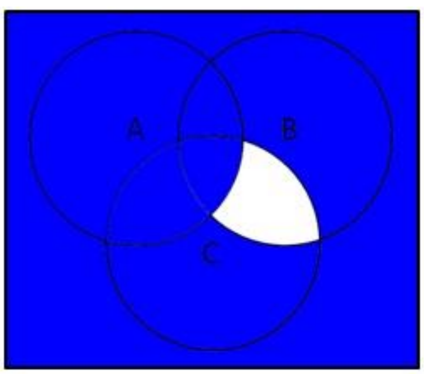
(a)



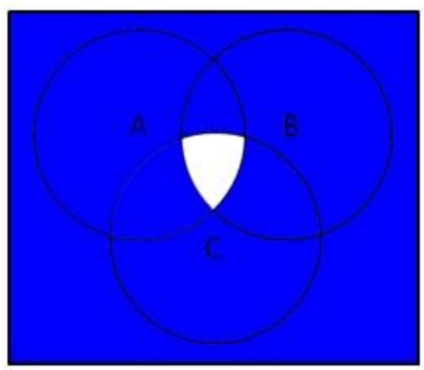
(b)



(c)



(d)



(e)

2. Let $S = A \cap (B \cup C)$. For each of the following sets R, mark True if it is must be equal to S for all sets A, B, and C, and false otherwise. To answer True, you should be able to transform S into R using the set identities. To answer False, you should be able to provide a counterexample (i.e., example sets A, B, and C such that there is some element that is in S but not in R, or vice versa.)

a) $R = A \cap (C \cup B)$
b) $R = A \cup (C \cap B)$
c) $R = (A \cap C) \cup (A \cap B)$

3. Let $S = A \cap \sim(B \cup C)$. For each of the following sets R, mark True if it is must be equal to S for all sets A, B, and C, and false otherwise. To answer True, you should be able to transform S into R using the set identities. To answer False, you should be able to provide a counterexample (i.e., example sets A, B, and C such that there is some element that is in S but not in R, or vice versa.)

a) $R = (A \cup \sim B) \cap \sim C$
b) $R = (A \cap \sim C) \cap \sim B$
c) $R = (A \cap \sim B) \cap (A \cap \sim C)$

Generalized Union and Intersection

Union and intersection are both associative and commutative.

So we have, for example:

$$(A \cup B) \cup C \quad = \quad A \cup (B \cup C) \quad = \quad (C \cup B) \cup A$$

And similarly for $\cap$.

So it makes sense to generalize both operations to an arbitrary number of sets. All we care about is what sets are involved. The order of the operations doesn't matter. It's useful to introduce a notation for doing this. We already have such a notation for sum and product (of numbers):

$$\sum_{i=1}^{n} a_i \qquad \prod_{i=1}^{n} a_i$$

Analogously for sets, we will write:

$$\bigcup_{i=1}^{n} S_i \qquad \bigcap_{i=1}^{n} S_i$$

Problems

1. Let

S_1 = {cherry, apple, pear, grape}
S_2 = {mango, pineapple, orange, apple, raspberry}
S_3 = {apple, peach, grape, orange}
S_4 = {apple, pear, pineapple}

What is the value of:

$$\bigcap_{i=1}^{4} S_i$$

2. For any positive integer i, let:

$S_i = \{i\}$

What is the value of:

$$\bigcup_{i=1}^{10} S_i$$

Law of the Excluded Middle

Recall that, when we began reasoning in logic, we took as an axiom the Law of the Excluded Middle.

It too has an analog as we reason about sets:

Law of the Excluded Middle:

$p \vee \neg p$ $\qquad\qquad$ $A \cup \sim A = U$

An alternative way to state this is in terms of an arbitrary element of A:

$$\neg(x \in A) \quad \equiv \quad (x \in \sim A)$$

We can use this axiom to prove useful properties about sets and operations on them.

Here's a useful new set identity: $\qquad A \cap B \equiv \sim(\sim A \cup \sim B)$

Proof:

[1]	$x \in (A \cap B)$		
[2]	$(x \in A) \wedge (x \in B)$	Definition of $\cap$	[1]
[3]	$\neg(\neg(x \in A)) \wedge (x \in B)$	Double negation	[2]
[4]	$\neg(\neg(x \in A)) \wedge \neg(\neg(x \in B))$	Double negation	[3]
[5]	$\neg(x \in \sim A) \wedge \neg(\neg(x \in B))$	Excluded Middle (x not in A $\equiv$ x in $\sim A$).	[4]
[6]	$\neg(x \in \sim A) \wedge \neg(x \in \sim B)$	Excluded Middle (same for B)	[5]
[7]	$\neg((x \in \sim A) \vee (x \in \sim B))$	De Morgan (backwards from the way we more commonly use it)	[6]
[8]	$\neg(x \in (\sim A \cup \sim B))$	Definition of $\cup$	[7]
[9]	$x \in \sim(\sim A \cup \sim B)$	Excluded Middle	[8]

So, again, we have proved that two sets must be equal by showing that any element of one must necessarily be an element of the other.

One other comment about this proof: We did both Double Negation and Excluded Middle in two steps each. A more natural proof would collapse each pair of steps into one. We've done it this way here to match the way it must be done in the automatic proof checker that we are using.

Computing Set Values

We also, as we were working with logical expressions, saw that it may be possible to simplify expressions by doing some simple computations with the fundamental values T and F. There's a natural analogy with sets here too.

To imagine how these rules work, think of U (the universe) as corresponding to T. It is true that every element is in U. Think of $\varnothing$ as corresponding to F. For any element, it is false that it is in $\varnothing$.

On the left here you'll see the logical rules. On the right, in red, the corresponding set rules. The commutative version of all of these rules (i.e., $p \vee \neg p$ vs. $\neg p \vee p$) also holds.

- $p \vee \neg p \quad \equiv \quad T$ $\qquad S \cup {\sim}S \quad \equiv \quad U$
- $p \wedge \neg p \quad \equiv \quad F$ $\qquad S \cap {\sim}S \quad \equiv \quad \varnothing$
- $p \vee T \quad \equiv \quad T$ $\qquad S \cup U \quad \equiv \quad U$
- $p \vee F \quad \equiv \quad p$ $\qquad S \cup \varnothing \quad \equiv \quad S$
- $p \wedge T \quad \equiv \quad p$ $\qquad S \cap U \quad \equiv \quad S$
- $p \wedge F \quad \equiv \quad F$ $\qquad S \cap \varnothing \quad \equiv \quad \varnothing$

The first of these rules follows from the Law of the Excluded Middle. The others all have straightforward proofs. We've already seen the proof of the last rule.

Problems

1. Prove that, for any set S, $\quad S \cap {\sim}S = \varnothing$.

Write a proof that holds for any set S and any arbitrary element x of S.

Inference Rules

Identities are transformations that can be applied in either direction because they link expressions that are equivalent.

Inference rules, on the other hand, are one way streets. They allow us to derive new logical statements from ones we already know to be true.

The Boolean inference rules that we've been using to reason with arbitrary logical expressions suggest another set of inference rules. These are tailored to reasoning with logical claims about sets.

We'll list these rules here. As before, let A, B, and C be arbitrary sets and let U be the universe.

Modus Ponens:

From p and $p \to q$, infer q From $(x \in A) \wedge (A \subseteq B)$, infer $x \in B$

Modus Tollens:

From $p \to q$ and $\neg q$, infer $\neg p$ From $(A \subseteq B) \wedge (x \notin B)$, infer $x \notin A$

Disjunctive Syllogism:

From $p \vee q$ and $\neg q$, infer p From $(x \in A \cup B) \wedge (x \notin B)$, infer $x \in A$

Simplification:

From $p \wedge q$, infer p From $x \in A \cap B$, infer $x \in A$

Addition:

From p, infer $p \vee q$ From $x \in A$, infer $x \in A \cup B$

Conjunction:

From p and q, infer $p \wedge q$ From $(x \in A) \wedge (x \in B)$, infer $x \in A \cap B$

Hypothetical Syllogism:

From $p \to q$ and $q \to r$, infer $p \to r$ From $(A \subseteq B) \wedge (B \subseteq C)$, infer $A \subseteq C$

Contradictory Premises:

From p and $\neg p$, infer q From $(x \in A) \wedge (x \notin A)$, infer anything

Resolution:

From $p \vee q$ and $\neg p \vee r$, infer $q \vee r$ From $(x \in A \cup B) \wedge (x \in {\sim}A \cup C)$, infer $x \in B \cup C$

Big Idea

In the Appendix, you will find a one-page cheat sheet that lists all our set identities and rules of inference. You can print it or refer to it online.

Problems

1. Let A, B and C be any sets. Suppose that we know that $x \in A$, $x \in B$, and $x \notin C$. Consider each of the following claims. Mark a claim true if it must be true. Mark it false if there are any sets A, B and C for which it is false.

(Part 1) $x \in (A \cup B) \cup C$

(Part 2) $x \in (A \cup B) \cap C$

(Part 3) $x \in {\sim}C \cap A$

(Part 4) $x \in {\sim}B \cap C$

(Part 5) $x \in B \cup {\sim}A$

2. Two of the inference rules that we just presented are:

Addition:	From $x \in A$,	infer $x \in A \cup B$
Conjunction:	From $x \in A \wedge x \in B$, infer $x \in A \cap B$	

Let's consider whether we get additional sound inference rules if we reverse the two we've got. (Recall that an inference rule is sound just in case, whenever it is applied to a set P of premises, any conclusion that it produces is entailed by P, i.e., it must be true whenever P is.) In other words, can we run these rules backwards? Consider:

$\text{Addition}_{\text{FLIPPED}}$:	From $x \in A \cup B$,	infer $x \in A$
$\text{Conjunction}_{\text{FLIPPED}}$:	From $x \in A \cap B$,	infer $(x \in A) \wedge (x \in B)$

Which of the following statements is true? You should be able to prove your answer. In other words, if you argue that a flipped rule is sound, you should be able to prove it so. If you argue that it isn't, you should be able to show that with an example of its use that isn't correct.

a) Both of the flipped forms are sound.
b) $\text{Addition}_{\text{FLIPPED}}$ is sound but $\text{Conjunction}_{\text{FLIPPED}}$ is not.
c) $\text{Conjunction}_{\text{FLIPPED}}$ is sound but $\text{Addition}_{\text{FLIPPED}}$ is not.
d) Neither of the flipped forms is sound.

Proving that Two Sets are Equal

As we work with sets, one of the things that we most often want to do is to prove that two sets are equal. For example, that's what we'll need to do if we want to prove the correctness of the set identities that we've just listed.

How can we prove that two sets are equal? There are two approaches we can take:

- Write a chain of equivalent expressions. We can do this with set expressions using the set identities or we can do it with logical expressions and use the logical identities. In either case, suppose we have a chain such as:

  $E_1 \equiv E_2 \equiv E_3 \equiv. . . E_k$

  Then we have that $E_1 \equiv E_k$. Note that, to do this, we can use only *identities*. Inference rules are one-way streets, so they have no place here.

  This is the approach that we have been taking so far.

- Use inference rules. But, if we do this, *we must do two proofs*: To prove that two sets A and B are equal, we must prove both that every element of A is in B and that every element of B is in A. In other words, we must prove *both* $A \subseteq B$ and $B \subseteq A$.

In the sections that follow, we'll see examples of both of these approaches.

Proving the Correctness of the Identities and the Inference Rules

We'd like to be able to use the identities and the inference rules that we've just presented. Before we can do that, we have to prove that they're correct. Of course, in doing that, we can't actually use them. We have to do these proofs using the definitions of the set operators, coupled with the *logical* rules that we've already proven correct.

As we do these proofs, the close connection between logical expressions and set expressions will become even more obvious to us.

To prove that an identity is correct, we must prove that two sets are equal. Usually the easiest way to do this is to use a chain of logical identities.

Prove De Morgan$_1$: $\sim(A \cap B) = \sim A \cup \sim B$

For any sets *A* and *B* and any element x:

[1]	$x \in \sim(A \cap B)$		
[2]	$\neg(x \in (A \cap B))$	Excluded Middle	[1]
[3]	$\neg((x \in A) \wedge (x \in B))$	Def. of $\cap$	[2]
[4]	$\neg(x \in A) \vee \neg(x \in B)$	(Logical) De Morgan	[3]
[5]	$(x \in \sim A) \vee (x \in \sim B)$	Excluded Middle (twice)	[4]
[6]	$x \in (\sim A \cup \sim B)$	Def. of $\cup$	[5]

We've now shown that for any arbitrary x, the claim that it is in $\sim(A \cap B)$ is equivalent to the claim that it is in $\sim A \cup \sim B$. Since two sets are equal exactly in case they have the same elements, we have that:

$$\sim(A \cap B) = \sim A \cup \sim B$$

Notice that we didn't cheat. We used *logical* De Morgan (whose correctness we have already proved), not *set* De Morgan (the identity that we were trying to prove here.)

Problems

1. Prove that for all sets *A*: $A \cup \varnothing = A$

2. Prove that set union is commutative. In other words, prove that, for all sets *A* and *B*: $A \cup B = B \cup A$.

Proving Claims about Sets II

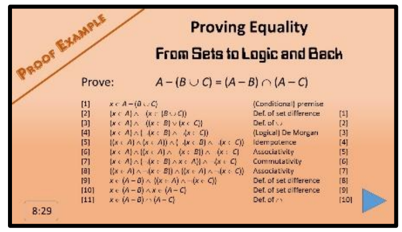
Proving Claims about Sets by Converting to Logical Claims

We now have a powerful enough set of tools to be able to prove many kinds of useful things about sets. Our most generally useful strategy will be to work with arbitrary set elements and convert our set problems to logic problems.

Prove that, for any sets A, B, and C,

$$A - (B \cup C) = (A - B) \cap (A - C).$$

We'll convert the set problem to a logic problem (and then back again). For any sets A, B, and C and any element x:



https://www.youtube.com/watch?v=Y3st4Kmm-VI

[1]	$x \in A - (B \cup C)$		
[2]	$(x \in A) \wedge \neg(x \in (B \cup C))$	Def. of set difference	[1]
[3]	$(x \in A) \wedge \neg((x \in B) \vee (x \in C))$	Def. of $\cup$	[2]
[4]	$(x \in A) \wedge (\neg(x \in B) \wedge \neg(x \in C))$	(Logical) De Morgan	[3]
[5]	$((x \in A) \wedge (x \in A)) \wedge (\neg(x \in B) \wedge \neg(x \in C))$	Idempotence	[4]
[6]	$(x \in A) \wedge ((x \in A) \wedge \neg(x \in B)) \wedge \neg(x \in C)$	Associativity	[5]
[7]	$(x \in A) \wedge (\neg(x \in B) \wedge x \in A)) \wedge \neg(x \in C)$	Commutativity	[6]
[8]	$((x \in A) \wedge \neg(x \in B)) \wedge ((x \in A) \wedge \neg(x \in C))$	Associativity	[7]
[9]	$x \in (A - B) \wedge ((x \in A) \wedge \neg(x \in C))$	Def. of set difference	[8]
[10]	$x \in (A - B) \wedge x \in (A - C)$	Def. of set difference	[9]
[11]	$x \in (A - B) \cap (A - C)$	Def. of $\cap$	[10]

We've written out every associativity and commutativity step. We've done that here just to be very careful and to show what an easily automatically checkable proof would look like. Usually when we write nontrivial proofs, we omit them.

Step [5] is perhaps the least obvious one. Why did we replace $(x \in A)$ with the redundant expression $(x \in A) \wedge (x \in A)$? Because we were looking at our goal ([11]) and we noticed that it mentions A twice.

Notice that we have been careful to use only definitions and identities. We've used no inference (one-way only) rules. So we do not need to do a second proof to show that:

$$x \in (A - B) \cap (A - C) \rightarrow x \in A - (B \cup C).$$

So we've shown that for any arbitrary x, it is in $A - (B \cup C)$ if and only if it is in $(A - B) \cap (A - C)$. Since two sets are equal exactly in case they have the same elements, we have that:

$$A - (B \cup C) \quad = \quad (A - B) \cap (A - C)$$

Problems

1. Prove or disprove the claim that for any sets A, B, and C, $A - (B \cap C) = (A - B) - C$.

2. Let the universe U be the set of companies. Consider the claim that, for any values of the sets *Customers*, *Suppliers*, and *Competitors*:

$$Customers - (Suppliers - Competitors) = (Customers - Suppliers) \cup (Customers \cap Competitors)$$

Let C = *Customers*, S = *Suppliers*, and X = *Competitors*. Prove this claim:

$$C - (S - X) = (C - S) \cup (C \cap X)$$

Working Directly with Set Expressions

Sometimes we want to show that two set expressions define the same set.

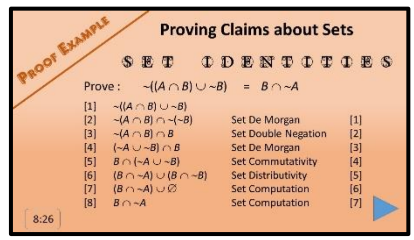
Sometimes we just want to simplify a long and messy expression that we've somehow derived.

In both cases, we know that we can (as we have just been doing) use the powerful general technique of reasoning about an arbitrary element of a set.

But sometimes it's easier to manipulate set expressions directly, using some combination of:

- the set identities that we've just described (plus any others that you want badly enough to prove the correctness of), and
- computation.

Prove that: $\sim((A \cap B) \cup \sim B) = B \cap \sim A$



http://youtu.be/Tlf2k_oPVps

[1]	$\sim((A \cap B) \cup \sim B)$		
[2]	$\sim(A \cap B) \cap \sim(\sim B)$	Set De Morgan	[1]
[3]	$\sim(A \cap B) \cap B$	Set Double Negation	[2]
[4]	$(\sim A \cup \sim B) \cap B$	Set De Morgan	[3]
[5]	$B \cap (\sim A \cup \sim B)$	Set Commutativity	[4]
[6]	$(B \cap \sim A) \cup (B \cap \sim B)$	Set Distributivity	[5]
[7]	$(B \cap \sim A) \cup \varnothing$	Set Computation	[6]
[8]	$B \cap \sim A$	Set Computation	[7]

Problems

1. Let $S = (A \cap B) \cup {\sim}B$. For each of the following sets R, mark True if R must equal S, regardless of the values of A and B. Mark False otherwise. You should be able to prove each of your answers. If you mark True, you should be able to show a proof that exploits the set identities and the computational rules that we've just presented. If you mark False, you should be able to present a counterexample.

a) $R = {\sim}B \cup A$
b) $R = ({\sim}B \cup A) \cap B$
c) $R = A - B$

2. Prove that $(A \cap B) \cup {\sim}B = (({\sim}B \cup A) \cap C) \cup (({\sim}B \cup A) \cap {\sim}C)$

3. Prove that: ${\sim}(A \cup {\sim}B) \cap (C \cap {\sim}B) = \varnothing$

Proving Claims about Subsets

It's often useful to be able to reason about subset relationships.

> For example, suppose that we have:
>
> [1] $WeekendGuards \subseteq HourlyEmployees$
> [2] All hourly employees get paid weekly.
>
> We'd like to be able to conclude:
>
> [3] All weekend guards get paid weekly.

Recall the definition of $\subseteq$:

$$A \subseteq B \quad \equiv \quad \forall x\, ((x \in A) \rightarrow (x \in B))$$

This definition makes clear the relationship between the set operator subset ($\subseteq$) and the logical operator implies ($\rightarrow$).

Also recall that one natural way to define a set A is to define a predicate P_A that is true of all and only the elements of A. In that case, we have:

$$x \in A \quad \equiv \quad P_A(x)$$

Putting these two together, we have:

$$A \subseteq B \quad \equiv \quad \forall x\, (P_A(x) \rightarrow P_B(x))$$

> To see this clearly, let's return to the weekend guards example, which we'll now restate using just logical expressions:
>
> [1] $\forall x\, ((WeekendGuard(x)) \rightarrow HourlyEmployee(x))$
> [2] $\forall x\, ((HourlyEmployee(x)) \rightarrow PaidWeekly(x))$
>
> It's now obvious that we can conclude:
>
> [3] $\forall x\, ((WeekendGuard(x)) \rightarrow PaidWeekly(x))$

So we see that one way to reason about subsets is just to bail out to logic. Sometimes, however, if we're in the middle of reasoning about sets, it's useful to be able to reason about subsets directly. So we may want to prove some general claims about them. Of course, the way we do that is much the same as the way we prove other set claims: we convert them to logic, exploit whatever tools of logical reasoning are necessary, and then convert it back to a subset claim.

We can prove the following useful claim about subsets:

For any sets S, A, and B:

$$S \subseteq (A \cap B) \quad \equiv \quad (S \subseteq A) \wedge (S \subseteq B)$$

We'll do the proof in the usual way, by reasoning about an arbitrary element of S. Note that, for step [4], we use a Boolean identity that we haven't used before. We can easily proof its validity with a truth table.

[1]	$S \subseteq (A \cap B)$		
[2]	$(x \in S) \rightarrow (x \in (A \cap B))$	Definition of subset	[1]
[3]	$(x \in S) \rightarrow ((x \in A) \wedge (x \in B))$	Definition of intersection	[2]
[4]	$((x \in S) \rightarrow (x \in A)) \wedge ((x \in S) \rightarrow (x \in B))$	$p \rightarrow (r \wedge s) \equiv (p \rightarrow r) \wedge (p \rightarrow s)$	[3]
[5]	$(S \subseteq A) \wedge (S \subseteq B)$	Definition of subset (twice)	[4]

Often, however, we aren't able to rely exclusively on identity operations. In particular, to prove claims of the form $p \rightarrow q$, we often have to exploit at least some (one-way) inference rules.

Prove that, for any sets A, B, and C:

$$(A \subseteq (C \cup B)) \rightarrow ((A - C) \subseteq B)$$

Because we want to prove a claim of the form $P \rightarrow Q$, we'll use the Conditionalization rule. Our conditional premise is $(A \subseteq (C \cup B))$. We'll convert that to a logical expression, reason with it, and derive $((A - C) \subseteq B)$. We can then conclude that $(A \subseteq (C \cup B)) \rightarrow ((A - C) \subseteq B)$.

[1]	$(A \subseteq (C \cup B))$	(Conditional) Premise	
[2]	$(x \in A) \rightarrow (x \in (C \cup B))$	Def. of subset	[1]
[3]	$(x \in A) \rightarrow ((x \in C) \vee (x \in B))$	Def. of union	[2]
[4]	$\neg(x \in A) \vee ((x \in C) \vee (x \in B))$	Conditional disjunction	[3]
[5]	$(\neg(x \in A) \vee (x \in C)) \vee (x \in B)$	Associativity of *or*	[4]
[6]	$\neg(\neg(x \in A) \vee (x \in C)) \rightarrow (x \in B)$	Conditional disjunction	[5]
[7]	$(\neg\neg(x \in A) \wedge \neg(x \in C)) \rightarrow (x \in B)$	De Morgan	[6]
[8]	$((x \in A) \wedge \neg(x \in C)) \rightarrow (x \in B)$	Double Negation	[7]
[9]	$(x \in A - C) \rightarrow (x \in B)$	Def. of set difference	[8]
[10]	$(A - C) \subseteq B$	Def. of subset	[9]
[11]	$(A \subseteq (C \cup B)) \rightarrow ((A - C) \subseteq B)$	Conditional Discharge	[1], [10]

Problems

1. Prove that, for all sets A, B, and C: $((A \subseteq B) \wedge (B \subseteq C)) \rightarrow (A \subseteq C)$.

2. Prove that, for all sets A and B: $(A - B) \cup (B - A) \subseteq (A \cup B)$.

Proving that Two Sets Are Equal Using Two Subset Relationships

Often, we can prove that two sets A and B are equal using the algebraic (symbol manipulation) techniques that we have just been describing.

But sometimes that doesn't work (at least in any straightforward fashion). This typically happens when A and B are defined in different ways. In this case, we may want to appeal to a different technique for showing that $A = B$. We note that if A and B are both subsets of each other, they must be equal. Formally:

$$(A \subseteq B) \wedge (B \subseteq A) \quad \text{if and only if} \quad A = B$$

This must be so given the definition of set equality: Two sets are equal just in case they contain the same elements. If every element of A is also in B and every element of B is also in A, then A and B do contain the same elements.

So we can prove that $A = B$ in two steps:

- Show that $A \subseteq B$, and
- Show that $B \subseteq A$.

Suppose that we've asked our team to order a plaque for every employee who is celebrating his or her 5^{th} or 10^{th} anniversary with the company. The ceremony is this afternoon. We want to prove:

EmployeesWithPlaques = *AnniversaryEmployees*

We do this in two steps:

- Show that *EmployeesWithPlaques* $\subseteq$ *AnniversaryEmployees*. We do this by going through the pile of plaques and looking up each name in the corporate database to make sure that the employee is having an anniversary.
- Show that *AnniversaryEmployees* $\subseteq$ *EmployeesWithPlaques*. We do this by asking the corporate database for a list of anniversary employees. For each of them, we check to make sure that there's a plaque.

In this simple example, we could actually combine the steps by printing out the list of anniversary employees. We could pick up the plaques one at a time and cross out the matching name. If we end up with no stray plaques and no stray names, we're done. So, while there's an efficient way to combine the steps, we observe that what we're doing is to check the two-way subset relationship.

As we build more interesting formalisms, we may find that we want to show that two sets that are defined in very different ways are, in fact, the same. In these cases, showing a two-way subset relationship is often the way to go.

Proving Claims about Powersets

We prove claims about powersets in much the same way we prove claims about other sets: we reason about typical elements. But now we note that each such typical element is itself a set.

Recall the definition of the powerset $\wp(A)$ of a set A:

For any sets A and S:

$$S \in \wp(A) \quad \text{if and only if} \quad S \subseteq A$$

We'll first prove a very straightforward and useful claim. We won't need much logical reasoning to do it.

Prove that, for any sets A and B, $\wp(A \cap B) = \wp(A) \cap \wp(B)$.

To do this, we'll use as a Lemma a claim that we proved back in the section on subsets:

Subset Intersection Lemma: $S \subseteq (A \cap B) \equiv (S \subseteq A) \wedge (S \subseteq B)$

This is a good example of the use of lemmas to build new claims on ones we've already proved.

For any sets A, B and S:

[1]	$S \in \wp(A \cap B)$		
[2]	$S \subseteq (A \cap B)$	Def. of power set	[1]
[3]	$(S \subseteq A) \wedge (S \subseteq B)$	Subset Intersection Lemma	[2]
[4]	$(S \in \wp(A)) \wedge (S \subseteq B)$	Def. of powerset	[3]
[5]	$(S \in \wp(A)) \wedge (S \in \wp(B))$	Def. of powerset	[4]
[6]	$S \in \wp(A) \cap \wp(B)$	Def. of $\cap$	[5]

We've now shown that an arbitrary element S is in $\wp(A \cap B)$ if and only if it is in $\wp(A) \cap \wp(B)$. So, since the two sets have the same elements, we have that $\wp(A \cap B) = \wp(A) \cap \wp(B)$.

Problems

1. We've just proved that for any sets A and B, $\wp(A \cap B) = \wp(A) \cap \wp(B)$. But what about union? Consider the following claim:

For any sets A and B, $\wp(A \cup B) = \wp(A) \cup \wp(B)$.

It turns out that this claim is false. We can prove it false by exhibiting a counterexample. Consider each of the following proposed counterexamples. Mark each one Counterexample if it is, in fact, a counterexample to our claim. Mark it Not Counterexample otherwise.

a) Let $A = \varnothing$ and $B = \varnothing$.
b) Let $A = \{1\}$ and $B = \{2\}$.
c) Let $A = \{1\}$ and $B = \{1, 2\}$.
d) Let $A = \varnothing$ and $B = \{1, 2\}$.

2. Prove that, for any sets B and C:

$$(B \subseteq C) \rightarrow \wp(B) \subseteq \wp(C)$$

We will assume that we've already proven the following theorem about the transitivity of subset:

$$\forall X, Y, Z\ (\ ((X \subseteq Y) \wedge (Y \subseteq Z)) \rightarrow (X \subseteq Z)\ )$$

We'll use it as a premise.

Proof by Counterexample

When we're working with sets, there are two common kinds of universal claims:

- $\forall x \in S\ (P(x))$ In other words, P is true of every element of the set S.
- $S = T$ In other words, every element of S is also in T and vice versa.

The first form is obviously a universal claim. But the second one is a universal claim too.

So we can prove things about both of them using our general strategies for working with universal claims.

In particular, we can refute any universal claim by exhibiting a single counterexample.

> Consider the following claim:
>
> Let A, B, and C be any sets. If $A - C = A - B$ then $B = C$.
>
> We show that this claim is false with a counterexample: Let $A = \varnothing$, $B = \{1\}$, and $C = \{2\}$. $A - C = A - B = \varnothing$. But $B \neq C$.

Problems

1. Consider the following claim:

For any sets, A, B, and C, if $A \neq B$ then $A - C \neq B - C$.

We want to show that this claim is false by exhibiting a single counterexample. For each of the following, indicate whether it is such a counterexample:

a) $A = \{1, 2\}$. $B = \{1, 3\}$. $C = \mathbf{N}$.
b) $A = \{1, 2\}$. $B = \{2, 1\}$. $C = \mathbf{N}$.
c) $A = \{1, 2\}$. $B = \{1, 3\}$. $C = \varnothing$.
d) $A = \{1, 2\}$. $B = \{1, 3\}$. $C = \{2, 3\}$.

Inclusion/Exclusion Exploration

In many applications, we work with finite sets.

For example, we work with sets of students, sets of employees, sets of customers, etc.

So it's worth considering some interesting properties of finite sets.

Let's look at what happens when we union two finite sets A and B. What can we say about $|A \cup B|$ (the number of elements in $A \cup B$)?

Problems

1. Let: $A = \{1, 3\}$. So: $|A| = 2$
   $B = \{2, 4, 8\}$. So: $|B| = 3$

First, determine the value of $A \cup B$. Now answer the question: What is $|A \cup B|$?

2. Let: $A = \{1, 2, 3\}$ So: $|A| = 3$
   $B = \{1, 2, 4, 8\}$ So: $|B| = 4$

What is $|A \cup B|$?

3. Let: $A = \{1, 2, 4, 8\}$ So: $|A| = 4$
   $B = \{1, 2, 4, 8\}$ So: $|B| = 4$

What is $|A \cup B|$?

4. Let A be an arbitrary set.
   B be an arbitrary set.

What is $|A \cup B|$? You should be able to figure this out from your answers to the last few problems.

a) $|A| + |B|$
b) $|A| + |B| - |A \cap B|$
c) $|A - B| + |B - A|$
d) $|A| + |B| - |A - B| - |B - A|$
e) $|A \cap B| + |A| + |B|$

The Inclusion/Exclusion Principle

In the last few problems, we saw that, for any sets A and B:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

This observation is called the ***Inclusion/Exclusion Principle***.

But just doing a few examples, while good for intuition building, isn't a proof. We need to prove this claim.

This proof is a bit long. If you skip it, you can still take the Inclusion/Exclusion Principle as a theorem.

We'll do the proof by induction on the number of elements in B. (We chose B arbitrarily. We could equally well have done it by induction on the number of elements in A.)

Base case: $B = \emptyset$. So $|B| = 0$.

$A \cup B = A \cup \emptyset = A$ $\qquad$ $A \cap B = A \cap \emptyset = \emptyset$

$|A \cup B| = |A|$ $\qquad$ $|A \cap B| = 0$

So we have:

$$|A \cup B| = |A| = |A| + 0 = |A| + |B| = |A| + |B| - 0 = |A| + |B| - |A \cap B|$$

Induction step: We must prove that, for $n \geq 0$:

If: $|A \cup B| = |A| + |B| - |A \cap B|$ for all sets A and B where $|B| = n$,

Then: $|A \cup B| = |A| + |B| - |A \cap B|$ for all sets A and B where $|B| = n + 1$.

Let B be any set such that $|B| = n + 1$. Since n is at least 0, $n + 1$ is at least 1. So B contains at least one element. Pick any one such element. Call it x. Now consider $B' = B - \{x\}$. Note that $|B'| = |B| - 1 = n$.

We now consider two cases:

(1) $x \in A$: $A \cup B = A \cup B'$ (since x is in $A \cup B$ regardless of whether it was in B)

$|A \cup B| = |A \cup B'|$

Since $|B'| = n$, we can use the induction hypothesis to rewrite the right hand side, giving:

[1] $|A \cup B| = |A| + |B'| - |A \cap B'|$

Now observe: $x \in A$. $x \in B$. So $x \in A \cap B$. But, since $x \notin B'$, $x \notin A \cap B'$. For every other element y, y must be in both $A \cap B$ and $A \cap B'$ or neither. So:

$A \cap B = (A \cap B') \cup \{x\}$
$|A \cap B| = |A \cap B'| + 1$ (since x was not already in $A \cap B'$)
$|A \cap B| - 1 = |A \cap B'|$

Substituting this value for $|A \cap B'|$ into [1], we get:

$|A \cup B| = |A| + |B'| - (|A \cap B| - 1)$

Since $|B'| = |B| - 1$, we have:

$|A \cup B| = |A| + (|B| - 1) - (|A \cap B| - 1)$

Simplifying, we have:

$|A \cup B| = |A| + |B| - |A \cap B|$

(2) $x \notin A$: Since $x \in B$, $x \in A \cup B$.
However, since $x \notin B'$, $x \notin A \cup B'$.
For every other element y, y must be in both $A \cup B$ and $A \cup B'$ or neither. So:

$A \cup B = (A \cup B') \cup \{x\}$
$| A \cup B | = |A \cup B'| + 1$ (since x was not already in $A \cup B'$)

Since $|B'| = n$, we can use the induction hypothesis to rewrite the right hand side, giving:

[2] $|A \cup B| = |A| + |B'| - |A \cap B'| + 1$

Since $x \notin A$, x is not an element of either $|A \cap B|$ or $|A \cap B'|$. For every other element y, y must be in both $A \cap B$ and $A \cap B'$ or neither. So:

$A \cap B = A \cap B'$
$|A \cap B| = |A \cap B'|$

Substituting this value of $|A \cap B'|$ into [2], we get:

[3] $|A \cup B| = |A| + |B'| - |A \cap B| + 1$

The last fact we need is:

$|B'| = |B| - 1$

Substituting this into [3], we get:

$|A \cup B| = |A| + (|B| - 1) - |A \cap B| + 1$

Simplifying we get:

$|A \cup B| = |A| + |B| - |A \cap B|$

Problems

1. If A and B are disjoint, what is $|A \cup B|$? Mark true for each expression that must be true for any disjoint sets A and B. Mark False for all others.

a) $|A \cup B| = |A| + |B| - |A \cap B|$
b) $|A \cup B| = |A| + |B| + |A \cap B|$
c) $|A \cup B| = |A| + |B|$
d) $|A \cup B| = |A \cap B|$

2. How many integers in {1 . . . 1000} are divisible by 2 or 3? (Hint: Use the Inclusion/Exclusion principle. Consider the sets *DivisibleBy2* and *DivisibleBy3*.)

Computer Representation of Sets

Concrete Representations of Sets

In mathematics, we can reason about sets without concrete representations of them. But when we want computers to do the reasoning, we must begin with computationally effective representations.

We've already considered two such representational techniques:

- We can write a ***generator*** (also called an ***enumerator***). Its job is to output (in some unspecified order) all and only the elements of the set.

- We can write a ***recognizer***. Its job is to examine a candidate and return *True* if the candidate is in the set and *False* otherwise.

When we use either of these techniques, we don't need to write out explicitly all of the elements of the set that we're working with. This means that both of these techniques can be used not just to define finite sets (where we could have written out the elements if we'd wanted to) but also sets that aren't finite.

Generators and recognizers are particularly useful when there's a pattern that describes the elements of the set in question. Short programs can describe large sets.

Here's a recognizer (written in Python) for the set of integers that are evenly divisible by 3. (To make it simple, we've assumed a universe of integers, so the program assumes that its input, n, is an integer. Also, note that the expression n%3 returns the remainder when n is divided by 3.)

```
def div_3(n):
    if n%3 == 0:
        return True
    else:
        return False
```

This program is a lot shorter than a list of even the first couple of thousand integers that are divisible by 3.

But sometimes there's not a pattern. There's just a collection of elements that have occurred in some way in a real problem.

In our factory, we might need to work with the following set that contains the numbers of the parts that are to be fabricated this week:

{46454, 363539, 84747, 4242, 96579, 353535, 513243}

In cases like this, there really isn't any good way to represent the set except by explicit mention of the elements that it contains.

We can do that in code.

For example, here's a Python program to recognize the set of part numbers (again assume that the universe is the integers. Also note that \ just means that we're continuing a long expression onto the next line.)

```
def part(n):
    if  n == 46454 or n == 363539 or n == 84747 or \
        n == 4242 or n == 96579 or n == 353535 or \
        n == 513243:
        return True
    else:
        return False
```

But there's got to be a better way. This program is not compact. And, worse, each week, we have to go in and change the code to look for new part numbers.

There are several better ways. In general, the idea is to get set definitions out of code and into some declarative representation that is both more flexible to use and more reliable to update. Two common ways to do that are:

- Store the instances in an external database. If we do this, many programs can use the same data.
- Store the instances in an internal data structure that has been specifically designed to represent sets. We'll look at one way to do this on the next slide.

Bit Vector Representations of Sets

Let *U* be a finite universe from which the elements of a set *S* will be taken. Imagine a one-row table *R* that has one column for each element of *U*. We'll refer to the elements of *R* as:

> $R[k]$ is the k^{th} element of *R*.

It is common in the programming world to start counting from 0 when we do this. We'll stick with the more everyday convention in which we start counting from 1, but it really doesn't matter as long as we're consistent.

Suppose that *U* contains 10 elements. Then *R* will look like this (where we've shown the names of the elements of *U* above their respective entries):

U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}

To represent a set *S*, drawn from *U*, we'll fill this table with Boolean values as follows:

- If $U_k \in S$: $R[k] = True$
- If $U_k \notin S$: $R[k] = False$

We wouldn't have to think of the values in *R* as Booleans. They could be drawn from any two-valued set. So they could be either % or #. Or they could be • or blank. Or they could be 0 and 1. The set {0, 1} is exactly the set of digits that occur when we represent numbers in binary (base 2). When we do that, a single digit is called a *bit* (short for binary digit). For this reason, the representation that we're describing is often called a ***bit vector representation***. In this representation, 0 corresponds to *False* and 1 corresponds to *True*.

Let *U* be {11, 22, 33, 44, 55, 66, 77, 88, 99}. Let *S* = {99, 55}. Then we can represent *S* with the following bit vector (shown in a couple of standard ways):

11	22	33	44	55	66	77	88	99
				•				•

11	22	33	44	55	66	77	88	99
False	False	False	False	True	False	False	False	True

11	22	33	44	55	66	77	88	99
0	0	0	0	1	0	0	0	1

Problems

1. Define the following two sets:

A =

11	22	33	44	55	66	77	88	99
False	True	False	False	True	False	False	False	True

B =

11	22	33	44	55	66	77	88	99
False	True	True	False	False	False	False	False	True

Show the bit vector that describes $A \cup B$:

11	22	33	44	55	66	77	88	99
False	True	True	False	True	False	False	False	True

Set Operations Using Bit Vector Representations

The last problem that we did suggests one of the main advantages of bit vector representations:

To perform operations on sets, all we need to do is to perform Boolean operations on their bit vector representations (again thinking of 0 as *False* and 1 as *True*).

Problems

1. Define the following two sets:

A =

11	22	33	44	55	66	77	88	99
False	True	False	False	True	False	False	False	True

B =

11	22	33	44	55	66	77	88	99
False	True	True	False	False	False	False	False	True

Show the bit vector that describes $A \cap B$:

2. Define the following set:

A =

11	22	33	44	55	66	77	88	99
False	True	False	False	True	False	False	False	True

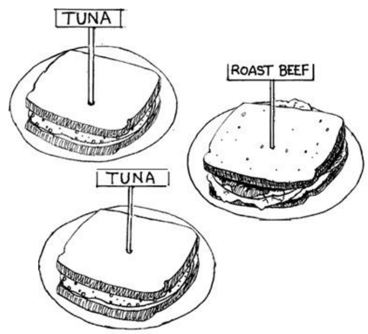
Show the bit vector that describes ~*A*:

Multisets

The Key Idea

Sets don't have duplicate elements. But sometimes duplicates matter.

Recall our sandwich order example. Suppose that a group of you are together at a meeting and one of you has offered to run to the deli for sandwiches. Everyone is asked to submit an order. If two people ask for tuna, it's important that tuna get entered twice.



A ***multiset*** is an unordered collection of (not necessarily distinct) elements.

Suppose that we want to find the average score on last week's test. Then we need to start with a multiset that contains the individual scores. We might, for example, have:

S = {90, 57, 89, 95, 68, 90, 78, 95, 87, 92, 98, 88, 69, 100, 92, 84, 86, 100}

If A is a multiset and x is some object, then the ***multiplicity*** of x in A is the number of occurrences of x in A. We'll write that as:

$\#_x(A)$ = the number of occurrences of x in A.

If S is the multiset of exam scores given above, then $\#_{90}(S) = 2$. $\#_{45}(S) = 0$.

Notice that every set S is also a multiset. It simply has the property that, for every element x in the universe, either $\#_x(S) = 0$ (if $x \notin S$) or $\#_x(S) = 1$ (if $x \in S$).

The standard set relationships and operations, plus one new and very useful one, can be defined for multisets. For example:

Union: For any multisets A and B and element x:

$$\#_x(A \cup B) = \text{maximum}(\#_x(A), \#_x(B))$$

Intersection: For any multisets A and B and element x:

$$\#_x(A \cap B) = \text{minimum}(\#_x(A), \#_x(B))$$

Difference: For any multisets A and B and element x, we want to subtract the elements of B from the elements of A, but we must assure that the result never goes below 0, even if there are more x's in B than in A. So we have:

$$\#_x(A - B) = \text{maximum}(0, \#_x(A) - \#_x(B))$$

Subset: For any multisets A and B:

$$A \subseteq B \text{ if and only if } \quad \forall x\ (\#_x(A) \leq \#_x(B))$$

Sum: This is a new operator, which we define as: For any multisets A and B and element x:

$$\#_x(A + B) = \#_x(A) + \#_x(B)$$

Multisets are important in many practical applications (not just sandwich ordering).

Define the following sets:

$Chem_1$, the equipment that is needed for the Chemistry 1 lab:

{beaker, beaker, beaker, Bunsen, Bunsen, tube, tube, tube, tube, tube, tube}

$Chem_2$, the equipment that is needed for the Chemistry 2 lab:

{beaker, beaker, Bunsen, tube, tube, tube, pipette, scale, pHmeter}

The equipment that we need for the two labs if we schedule them at the same time is:

$Chem_1 + Chem_2$ =

{beaker, beaker, beaker, beaker, beaker, Bunsen, Bunsen, Bunsen, tube, tube, tube, tube, tube, tube, tube, tube, tube, pipette, scale, pHmeter}

The equipment that we need for the two labs if we *don't* schedule them at the same time (and thus they can share equipment) is:

$Chem_1 \cup Chem_2$ =

{beaker, beaker, beaker, Bunsen, Bunsen, tube, tube, tube, tube, tube, tube, pipette, scale, pHmeter}

Problems

1. Sometimes, when we analyze texts, we are interested in the words they contain but not in the order in which the words occur. This is true when we do key word searches to find texts that talk about specific things. It's also true, for example, if we want to rate texts based on the difficulty of the vocabulary that they use. In these cases, it's useful to think of a text T as a multiset of words.

Let T_1 be a multiset that contains the words of a text we want to work with. For example, if the text were just, "The Internet is the most powerful tool for changing the Internet." Then T_1 would be:

{the, the, the, Internet, Internet, is, most, powerful, tool, for, changing}

Just in this simple case we notice a couple of things about T_1. Yes, "the" occurs often. So we may want to define the universe so that we throw away very common words. But "Internet" occurs more than "tool". That suggests that this text is more about the Internet than it is about tools. Of course, in this tiny example, we can't conclude much. But with a longer text, observations like this could be useful.

In some of the analyses that we're about to do, we'll want to consider the universe of possible words. So define:

U = the *set* that contains all words that might occur in any text.

(Part 1) Suppose that we want to build a model of the word use of a particular author. Then we will want to combine word counts from many texts that she's written. So, consider a second text, "The Internet is changing the world every day." Let T_2 be the multiset that contains its words:

{the, the, Internet, is, changing, world, every, day}

We want to analyze the author's style by working with a large corpus that contains many texts, including these. So we want to build the following multiset, which we'll call T_3, which combines T_1 and T_2:

{the, the, the, the, the, Internet, Internet, Internet, is, is, most, powerful, tool, for, changing, changing, world, every, day}

Which of the following multiset expressions correctly describes T_3:

a) $T_3 = T_1 \cup T_2$
b) $T_3 = T_1 \cap T_2$
c) $T_3 = T_1 + T_2$
d) $T_3 = U \cup (T_1 \cap T_2)$

(Part 2) In order to focus our analysis on the words that really matter, we'd like to get rid of all the words that occur only once in the text corpus. (Just as an aside: There's a word for such words. They're called ***hapax legomena***.) We're willing to alter the other counts as long as it isn't by much and we preserve information about the relative frequencies of words. We'll define T_4 to do that. Which of the following multiset expressions correctly describes T_4:

a) $T_4 = T_3 \cup U$
b) $T_4 = T_3 \cap U$
c) $T_4 = T_3 + U$
d) $T_4 = T_3 - U$

(Part 3) We want to use textual analysis to try to figure out who the author of our large corpus was. There are several candidates. For each candidate, we have a set that contains the words that that author has been known to use. Our idea is to look at our text to see if it contains words that a particular author would never have used. Let A be the set of words that our first author ever uses. Which of the following multiset expressions correctly describes T_5, the set of all words in T_3 that this author has never before been known to use:

a) $T_5 = T_3 \cup (U - A)$
b) $T_5 = T_3 \cap (U - A)$
c) $T_5 = (U - A) - T_3$
d) $T_5 = T_3 \cap (U \cap A)$

2. Let A be the empty multiset ({}). Is it true that for all multisets B, $A \subseteq B$? Try to prove your answer.

a) It is true and there's a straightforward proof based on the definition of $\subseteq$.
b) It is true but we need to use induction to prove it.
c) It isn't true and I've found a counterexample.
d) I don't think it's true but there's no simple counterexample.

The Fundamental Theorem of Arithmetic

Multisets are also important in mathematics. We'll look here at one important example.

The ***Fundamental Theorem of Arithmetic*** (also called the ***Unique Prime Factorization Theorem***) tells us that every integer greater than 1 can be uniquely described as a multiset of prime factors.

For example:

5	=	5	So the prime factorization of 5 is {5}.
12	=	2 · 2 · 3.	So the prime factorization of 12 is {2, 2, 3}.
270	=	2 · 3 · 3 · 3 · 5.	So the prime factorization of 270 is {2, 3, 3, 3, 5}.

Define the ***greatest common divisor*** (or ***gcd***) of two integers n and m to be the largest integer that divides both of them.

For example, *gcd*(12, 270) = 6.

A straightforward way to determine the greatest common divisor of two integers n and m is to intersect their prime factorizations to create a new multiset of prime factors. Then $gcd(n, m)$ is the number whose prime factorization we've just computed.

For example, suppose that we want to compute *gcd*(12, 270). We can use the prime factorizations of 12 and 270:

$$\{2, 2, 3\} \cap \{2, 3, 3, 3, 5\} = \{2, 3\}$$

The integer whose prime factorization is {2, 3} is 6.

Nifty Aside

Thinking of integers (particularly very large ones) as products of prime factors is important in modern cryptography. For example it plays a key role in the RSA encryption algorithm, which depends on two important facts:

- No *efficient* technique for finding prime factorizations is known. So if the only way for an unauthorized person to decrypt a message that I've sent you is to factor a very large number, our communication is secure.

- But, interestingly, an efficient technique for finding greatest common divisors (without finding full prime factorizations first) has been known since Euclid (in about 300 BC). This is important since agents that want to encrypt their communications using RSA must first compute *gcd*.

Problems

1. For each of the following multisets, mark True if it corresponds to the prime factorization of 3060. Mark False otherwise.

a) {2, 2, 3, 3, 17, 5}
b) {2, 2, 3, 3, 5, 17}
c) {2, 3, 5, 17}
d) {2, 3, 5, 6, 17}
e) {1, 2, 2, 3, 3, 17, 5}

2. The prime factorization of 116,280 is {2, 2, 3, 3, 17, 5, 19, 2}. The prime factorization of 151,620 is {2, 2, 3, 7, 5, 19, 19}. What is *gcd*(116280, 151620)?

3. We stated the Fundamental Theorem of Arithmetic without proof. Its proof has two parts:

i. Prove that every integer greater than 1 can be described as a multiset of prime factors.
ii. Prove that that multiset is unique.

Step ii is beyond the scope of this discussion. But Step i can be done using techniques that we have at our disposal. Let's do it.

We'll start by using Case Enumeration. We'll consider two cases. What are they? (Hint: Consider the first several integers greater than 1: 2, 3, 4, 5, 6, 7, 8, . . . How would you show that this claim holds for each of them? Have you used exactly two techniques?)

We'll prove the second case by strong induction.

Start by writing an explicit statement of the claim we need to prove. Write it as a predicate $P(n)$ where n is an integer greater than 1. (Hint: It's fine to use straightforward English to do this.)

Next we need to prove the base case(s). What is/are they? (Hint: This is a little bit tricky. There are more than a couple

Induction step: We must prove that $P(n)$ holds for all composite integers greater than 1. We are going to use strong induction. So we must prove that if P holds for all values less than n, it must also hold for n. So we must prove:

$$(\forall k<n\ (P(k)) \rightarrow P(n)$$

Since n is composite, what do we know about it? Write down a simple fact

Now use the induction hypothesis. What can we say about a prime factorization of a and b? And can we go from there to the required claim, $P(n)$? Write down the rest of the proof.

Appendices

Set Identities

Double Negation

$p \equiv \neg(\neg p)$ $\qquad$ $A = \sim(\sim A)$

Equivalence

$(p \equiv q) \equiv (p \rightarrow q) \wedge (q \rightarrow p)$ $\qquad$ $(A = B) \equiv (A \subseteq B) \wedge (B \subseteq A)$

Idempotence

$(p \wedge p) \equiv p$ $\qquad$ $(A \cap A) = A$

$(p \vee p) \equiv p$ $\qquad$ $(A \cup A) = A$

De Morgan

$(\neg(p \wedge q)) \equiv (\neg p \vee \neg q)$ $\qquad$ $\sim(A \cap B) = \sim A \cup \sim B$

$\neg(p \vee q) \equiv (\neg p \wedge \neg q)$ $\qquad$ $\sim(A \cup B) = \sim A \cap \sim B$

Commutativity of *or* $\qquad$ **Commutativity of Union**

$(p \vee q) \equiv (q \vee p)$ $\qquad$ $(A \cup B) = (B \cup A)$

Commutativity of *and*: $\qquad$ **Commutativity of Intersection**

$(p \wedge q) \equiv (q \wedge p)$ $\qquad$ $(A \cap B) = (B \cap A)$

Associativity of *or*: $\qquad$ **Associativity of Union**

$(p \vee (q \vee r)) \equiv ((p \vee q) \vee r)$ $\qquad$ $A \cup (B \cup C) = (A \cup B) \cup C$

Associativity of *and*: $\qquad$ **Associativity of Intersection**

$(p \wedge (q \wedge r)) \equiv ((p \wedge q) \wedge r)$ $\qquad$ $A \cap (B \cap C) = (A \cap B) \cap C$

Distributivity of *and* over *or*: $\qquad$ **Distributivity of Intersection over Union**

$(p \wedge (q \vee r)) \equiv ((p \wedge q) \vee (p \wedge r))$ $\qquad$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Distributivity of *or* over *and*: $\qquad$ **Distributivity of Union over Intersection**

$(p \vee (q \wedge r)) \equiv ((p \vee q) \wedge (p \vee r))$ $\qquad$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Conditional Disjunction:

$(p \rightarrow q) \equiv (\neg p \vee q)$ $\qquad$ $(A \subseteq B) \equiv ((\sim A \cup B) = U)$

Contrapositive:

$(p \rightarrow q) \equiv (\neg q \rightarrow \neg p)$ $\qquad$ $(A \subseteq B) \equiv (\sim B \subseteq \sim A)$

Law of the Excluded Middle:

$p \vee \neg p$ $\qquad$ $A \cup \sim A = U$

Set Inference Rules

Modus Ponens:

From p and $p \rightarrow q$, infer q $\qquad$ From $(x \in A) \wedge (A \subseteq B)$, infer $x \in B$

Modus Tollens:

From $p \rightarrow q$ and $\neg q$, infer $\neg p$ $\qquad$ From $(A \subseteq B) \wedge (x \notin B)$, infer $x \notin A$

Disjunctive Syllogism:

From $p \vee q$ and $\neg q$, infer p $\qquad$ From $(x \in A \cup B) \wedge (x \notin B)$, infer $x \in A$

Simplification:

From $p \wedge q$, infer p $\qquad$ From $(x \in A \cap B)$, infer $x \in A$

Addition:

From p, infer $p \vee q$ $\qquad$ From $(x \in A)$, infer $x \in A \cup B$

Conjunction:

From p and q, infer $p \wedge q$ $\qquad$ From $(x \in A) \wedge (x \in B)$, infer $x \in A \cap B$

Hypothetical Syllogism:

From $p \rightarrow q$ and $q \rightarrow r$, infer $p \rightarrow r$ $\qquad$ From $(A \subseteq B) \wedge (B \subseteq C)$, infer $A \subseteq C$

Contradictory Premises:

From p and $\neg p$, infer q $\qquad$ $A \cap \sim A = \varnothing$

Resolution:

From $p \vee q$ and $\neg p \vee r$, infer $q \vee r$ $\qquad$ From $(x \in A \cup B) \wedge (x \in \sim B \cup C)$, infer $x \in A \cup C$