

Contents

Lecture 1:	What is MAT1830 about?
Lecture 2:	Divisors and primes
Lecture 3:	Congruences
Lecture 4:	Logic
Lecture 5:	Tautologies and logical equivalence
Lecture 6:	Rules of inference
Lecture 7:	Predicates and quantifiers
Lecture 8:	Predicate logic
Lecture 9:	Mathematical induction
Lecture 10:	Induction and well-ordering
Lecture 11:	Sets
Lecture 12:	Operations on sets
Lecture 13:	Functions
Lecture 14:	Examples of functions
Lecture 15:	Composition and inversion
Lecture 16:	Relations
Lecture 17:	Equivalence relations
Lecture 18:	Order relations
Lecture 19:	Selections and arrangements
Lecture 20:	Pascal's triangle
Lecture 21:	Probability
Lecture 22:	Conditional probability and Bayes' theorem
Lecture 23:	Random variables
Lecture 24:	Expectation and variance
Lecture 25:	Discrete distributions
Lecture 26:	Recursion
Lecture 27:	Recursive algorithms
Lecture 28:	Recursion, lists and sequences
Lecture 29:	Graphs
Lecture 30:	Walks, paths and trails
Lecture 31:	Degree
Lecture 32:	Trees
Lecture 33:	Trees, queues and stacks

Unit Information

MAT1830 – Semester 1 2017

Course coordinator/lecturer (Clayton)

- Name: Dr. Daniel Horsley
- Office: 9Rnf/418
- Phone: 9905 4459
- Email: daniel.horsley@monash.edu

Lecturer (Clayton)

- Name: Dr. Chris Hough
- Office: 9Rnf/417
- Email: chris.hough@monash.edu

Lecturer (Clayton)

- Name: Prof. Ian Wanless
- Office: 9Rnf/448
- Phone: 9905 4442
- Email: ian.wanless@monash.edu

Lecturer (Sunway)

- Name: Dr. Lee Kien Foo
- Email: foo.lee.kien@monash.edu

Moodle Page

The course's moodle page can be accessed through the my.monash portal at <https://my.monash.edu.au/>

Unit Guide

The full unit guide can be found on the moodle page. It contains a topic outline and information on assessment, special consideration, plagiarism etc.

Course Information

- Lectures: Three hours per week (lectures run in weeks 1–12).
Clayton times:
 - Wed 12:00 – 1:00 in 25Exh/C1 & 16Rnf/S12
 - Thu 12:00 – 1:00 in 25Exh/C1 & 16Rnf/S9
 - Fri 4:00 – 5:00 in 25Exh/C1 & 16Rnf/S9
- Support classes: One per week as allocated (support classes run in weeks 2–12).
- Required materials: Course notes booklet – available for less than \$10 from the Clayton campus bookshop or available as a pdf from the course Moodle page. Note that there is no required textbook for the course.

Recordings of the lectures will be available through the moodle page.

Assessment

- Ten assignments worth 3% each (one due each week from week 3 to week 12).
- Final examination worth 70% (held in the examination period).

In order to pass the course you must receive at least 50% overall AND at least 40% for the assignments AND at least 40% for the exam.

The assignments will be issued in lectures and will be available from the course Moodle page. Assignments are to be submitted to your tutor during your support class. They will then be marked and returned to you in your next support class. No calculators or other materials will be allowed in the final exam.

Mathematics Learning Centre

The Mathematics Learning Centre is open weekdays from 11–2 in 9Rnf/G24 (from week 2 right through exams). You can drop in for help anytime during these hours.

Lecture 1: What is MAT1830 about?

Discrete mathematics studies objects which have distinct separated values (e.g. integers), as opposed to objects which vary smoothly (e.g. real numbers). You can think of it as being “digital” mathematics rather than “analogue” mathematics.

Discrete mathematics is particularly important in computer science and the two fields are very closely linked.

This course covers a wide range of topics in discrete mathematics including the following:

- Numbers
- Logic
- Induction and recursion
- Sets, functions and relations
- Probability
- Graph theory

1.1 What to expect

What we do here might be a bit different to a lot of the maths you’ve done in the past. We’ll be concentrating on really understanding the concepts, rather than simply learning how to solve certain types of questions.

For a lot of the questions we ask, there won’t be a single fixed procedure you can apply to get the answer. Instead, you’ll have to think carefully about what the question is asking and try to work out what is really going on. Don’t be afraid to try different things, play around, and look at examples.

We’ll also be emphasising the importance of proving results.

1.2 Proofs

A proof is essentially just a water-tight argument that a certain statement must be true. As

we’ll see, even if you are pretty sure that something is true, it can be really useful to have a proof of it, for a number of reasons.

1.3 Maths in computer science

As we mentioned above, maths and computer science are very closely related. The topics in this course all have many applications to computer science. For example:

- Number theory is used in cryptography to enable secure communication, identity verification, online banking and shopping etc.
- Logic is used in digital circuit design and in program control flow.
- Induction and recursion are used to study algorithms and their effectiveness.
- Functions are important in the theory of programming and relations are vital in database theory and design.
- Probability is vital for understanding randomised algorithms and for creating systems to deal with uncertain situations.
- Graph theory is used in software which solves allocation and scheduling problems.

Questions

- 1.1** What maths that you’ve done in the past would count as discrete? What would count as continuous instead? Are there grey areas?
- 1.2** Why might proofs be important to mathematicians and computer scientists?
- 1.3** Can you think of other links between maths and computer science?

Lecture 2: Divisors and primes

We say that integer a *divides* integer b if
 $b = qa$ for some integer q .

Example. 2 divides 6 because $6 = 3 \times 2$.

This is the same as saying that division with remainder gives remainder 0. Thus a does *not* divide b when the remainder is $\neq 0$.

Example. 3 does not divide 14 because it leaves remainder 2: $14 = 4 \times 3 + 2$.

When a divides b we also say:

- a is a *divisor* of b ,
- a is a *factor* of b ,
- b is *divisible* by a ,
- b is a *multiple* of a .

2.1 Primes

A positive integer $p > 1$ is a prime if its only positive integer divisors are 1 and p . Thus the first few prime numbers are

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, ...

The number 1 is not counted as a prime, as this would spoil the

Fundamental Theorem of Arithmetic.
Each integer > 1 can be expressed in exactly one way, up to order, as a product of primes.

Example. $210 = 2 \times 3 \times 5 \times 7$, and this is the only product of primes which equals 210.

This would not be true if 1 was counted as a prime, because many factorisations involve 1. E.g.

$$210 = 1 \times 2 \times 3 \times 5 \times 7 = 1^2 \times 2 \times 3 \times 5 \times 7 = \dots$$

2.2 Recognising primes

If an integer $n > 1$ has a divisor, it has a divisor $\leq \sqrt{n}$, because for any divisor $a > \sqrt{n}$ we also have the divisor n/a , which is $< \sqrt{n}$.

Thus to test whether 10001 is prime, say, we only have to see whether any of the numbers $2, 3, 4, \dots \leq 100$ divide 10001, since $\sqrt{10001} < 101$. (The least divisor found is in fact 73, because $10001 = 73 \times 137$.)

This explains a common algorithm for recognising whether n is prime: try dividing n by $a = 2, 3, \dots$ while $a \leq \sqrt{n}$.

The algorithm is written with a boolean variable *prime*, and n is prime if *prime* = T (true) when the algorithm terminates.

```
assign  $a$  the value 2.  
assign  $prime$  the value T.  
while  $a \leq \sqrt{n}$  and  $prime = T$   
    if  $a$  divides  $n$   
        give  $prime$  the value F  
    else  
        increase the value of  $a$  by 1.
```

2.3 Finding divisors

This algorithm also finds a prime divisor of n .

Either

the least $a \leq \sqrt{n}$ which divides n ,

or,

if we do not find a divisor among the $a \leq \sqrt{n}$, n itself is prime.

2.4 The greatest common divisor of two numbers

It is remarkable that we can find the greatest common divisor of positive integers m and n , $\gcd(m, n)$, without finding their prime divisors.

This is done by the famous *Euclidean algorithm*, which repeatedly divides the greater number by the smaller, keeping the smaller number and the remainder.

```

assign  $a$  the value  $m$ , and  $b$  the value  $n$ .
assign  $r$  as the remainder when  $a$  is
                                divided by  $b$ .
while  $r \neq 0$ 
    assign  $a$  the value  $b$ 
    assign  $b$  the value  $r$ 
    assign  $r$  as the remainder when  $a$  is
                                divided by  $b$ .
Give  $d$  the value of  $b$ .

```

The final divisor $d = \gcd(m, n)$.

Example. $m = 105, n = 27$

The first values are $a = 105, b = 27$,
so $r = 105 - 3 \times 27 = 24$.

The next values are $a = 27, b = 24$,
so $r = 27 - 1 \times 24 = 3$.

The next values are $a = 24, b = 3$,
so $r = 24 - 8 \times 3 = 0$.

Thus the final value of b is 3, which is $\gcd(105, 27)$.

This can be set out more neatly:

$$\begin{array}{rclclcl}
 105 & = & 3 & \times & 27 & + & 24 \\
 27 & = & 1 & \times & 24 & + & \mathbf{3} \\
 24 & = & 8 & \times & 3 & + & 0
 \end{array}$$

2.5 The Euclidean algorithm works!

We start with the precondition $m \geq n > 0$. Then the division theorem tells us there is a remainder $r < b$ when $a = m$ is divided by $b = n$. Repeating the process gives successively smaller remainders, and hence the algorithm eventually halts.

The first remainder $r = a - qb$ is divisible by any number that divides both $a = m$ and $b = n$, hence the same is true of all the remainders, including the last one, d .

Thus all the divisors of m and n divide d , which means $d = \gcd(m, n)$.

2.6 Extended Euclidean algorithm

If we have used the Euclidean algorithm to find that $\gcd(m, n) = d$, we can “work backwards” through its steps to find integers a and b such that $am + bn = d$.

Example. For our $m = 105, n = 27$ example above:

$$\begin{aligned}
 3 &= 27 - 1 \times 24 \\
 3 &= 27 - 1(105 - 3 \times 27) = -1 \times 105 + 4 \times 27
 \end{aligned}$$

So we see that $a = -1$ and $b = 4$ is a solution in this case.

Our first line above was a rearrangement of the second last line of our original Euclidean algorithm working. In the second line we made a substitution for 24 based on the first line of our original Euclidean algorithm working.

Questions

- 2.1** Write down multiples of 13, and multiples of 21, until you find a multiple of 13 and a multiple of 21 which differ by 1.
- 2.2** Can a multiple of 15 and a multiple of 21 differ by 1? If not, what is the smallest positive difference between such multiples?
- 2.3** Find $\gcd(13, 21)$ and $\gcd(15, 21)$, and suggest how they are related to the results in Questions 2.1 and 2.2.
- 2.4** Work out the prime factorisations of 999 and 1000.
- 2.5** You should find no common prime factor of 999 and 1000. How could you see this *without* factorising the numbers? (Hint: a common divisor of 1000 and 999 is also a common divisor of ... what?)

Lecture 3: Congruences

We're used to classifying the integers as either even or odd. The even integers are those that can be written as $2k$ for some integer k . The odd integers are those that can be written as $2k + 1$ for some integer k .

even	$\dots, -6, -4, -2, 0, 2, 4, 6, \dots$
odd	$\dots, -5, -3, -1, 1, 3, 5, \dots$

This classification is useful because even and odd integers have particular properties. For example, the sum of any two odd integers is even.

Similarly we can split the integers into three classes: those that are $3k$ for some integer k , those that are $3k + 1$ for some integer k , and those that are $3k + 2$ for some integer k .

$3k$	$\dots, -9, -6, -3, 0, 3, 6, 9, \dots$
$3k + 1$	$\dots, -8, -5, -2, 1, 4, 7, 10, \dots$
$3k + 2$	$\dots, -7, -4, -1, 2, 5, 8, 11, \dots$

These classes also have particular properties. For example, the sum of an integer in the second class and an integer in the third class will always be in the first class.

We don't have to stop with 3. We could divide integers into 4 different classes according to their remainders when divided by 4, and so on.

3.1 Congruences

Let $n \geq 2$ be an integer. We say integers a and b are *congruent modulo n* and write

$$a \equiv b \pmod{n}$$

when n divides $a - b$.

Example.

$$\begin{aligned} 19 &\equiv 13 \pmod{6} && \text{because 6 divides } 19-13 \\ 12 &\equiv 20 \pmod{4} && \text{because 4 divides } 20-12 \\ 22 &\equiv 13 \pmod{3} && \text{because 3 divides } 22-13 \end{aligned}$$

3.2 Working with congruences

When working with congruences modulo some fixed integer n , we can "substitute in" just like we can with equalities.

If $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$, then

$$a \equiv c \pmod{n}.$$

Example. Suppose $x \equiv 13 \pmod{7}$. Then $x \equiv 6 \pmod{7}$ because $13 \equiv 6 \pmod{7}$.

We can add, subtract and multiply congruences just like we can with equations.

If $a_1 \equiv b_1 \pmod{n}$ and $a_2 \equiv b_2 \pmod{n}$, then

- $a_1 + a_2 \equiv b_1 + b_2 \pmod{n}$
- $a_1 - a_2 \equiv b_1 - b_2 \pmod{n}$
- $a_1 a_2 \equiv b_1 b_2 \pmod{n}.$

Example. If $x \equiv 3 \pmod{8}$ and $y \equiv 2 \pmod{8}$, then

- $x + y \equiv 5 \pmod{8}$
- $x - y \equiv 1 \pmod{8}$
- $xy \equiv 6 \pmod{8}.$

We can also deduce that $x + 4 \equiv 7 \pmod{8}$, that $4x \equiv 12 \pmod{8}$ and so on, because obviously $4 \equiv 4 \pmod{8}$. Note as well that $4x \equiv 12 \pmod{8}$ can be simplified to $4x \equiv 4 \pmod{8}$.

In some situations we can also “divide through” a congruence by an integer.

If $a \equiv b \pmod{n}$ and d divides a, b and n , then

$$\frac{a}{d} \equiv \frac{b}{d} \pmod{\frac{n}{d}}.$$

3.3 Solving linear congruences

Think of a congruence like $7x \equiv 5 \pmod{9}$. This will hold if 9 divides $7x - 5$ or in other words if there is an integer y such that $7x - 5 = 9y$. So to solve our original congruence we can find an integer solution to $7x - 9y = 5$.

Some congruences don’t have solutions. For example, there is no solution to $10x \equiv 6 \pmod{20}$ because there are no integers x and y such that $10x - 20y = 6$.

We can find an expression for all the integers x that satisfy a congruence like $ax \equiv b \pmod{n}$ in the following way:

1. Find $d = \gcd(a, n)$.
2. If d doesn’t divide b , then there are no solutions.
3. If d divides b , then divide through the congruence by d to get an equivalent congruence $\frac{a}{d}x \equiv \frac{b}{d} \pmod{\frac{n}{d}}$.
4. Find integers x' and y' such that $\frac{a}{d}x' - \frac{n}{d}y' = \frac{b}{d}$. The integers x that satisfy the original congruence are exactly those for which $x \equiv x' \pmod{\frac{n}{d}}$.

Example. Find all integers x such that $36x \equiv 10 \pmod{114}$.

Using the Euclidean algorithm we find $\gcd(36, 114) = 6$. So 6 divides $36x - 114y$ for any integers x and y , and consequently $36x - 114y \neq 10$. This means that there are no integers x such that $36x \equiv 10 \pmod{114}$.

Example. Find all integers x such that $24x \equiv 8 \pmod{44}$.

Using the Euclidean algorithm we find $\gcd(24, 44) = 4$. So we divide through by 4 to get the equivalent congruence $6x \equiv 2 \pmod{11}$. Using the extended euclidean algorithm we see that $2 \times 6 - 1 \times 11 = 1$, and hence $4 \times 6 - 2 \times 11 = 2$. Thus the integers x such that $24x \equiv 8 \pmod{44}$ are exactly the integers $x \equiv 4 \pmod{11}$.

3.4 Modular inverses

A modular multiplicative inverse of an integer a modulo n is an integer x such that

$$ax \equiv 1 \pmod{n}.$$

From the last section we know that such an inverse will exist if and only if $\gcd(a, n) = 1$. If inverses do exist then we can find them using the extended Euclidean algorithm (there will be lots of inverses, but they will all be in one congruence class modulo n). These inverses have important applications to cryptography and random number generation.

Example. 8 should have a multiplicative inverse modulo 45 because $\gcd(8, 45) = 1$. Using the extended Euclidean algorithm we see that $-3 \times 45 + 17 \times 8 = 1$. So $8 \times 17 \equiv 1 \pmod{45}$. This means that 17 is a multiplicative inverse of 8 modulo 45.

Questions

3.1 Are the following true or false?

- $6 \equiv 3 \pmod{3}$
- $9 \equiv 18 \pmod{8}$
- $5x + 6 \equiv 2x \pmod{3}$

3.2 Prove all of the facts about congruences that were stated in this lecture (use the definition of congruence modulo n and the definition of divides).

3.3 Find an expression for all the integers x that satisfy $9x \equiv 36 \pmod{60}$.

Lecture 4: Logic

The simplest and most commonly used part of logic is the logic of “and”, “or” and “not”, which is known as *propositional logic*.

A proposition is any sentence which has a definite truth value (true= T or false= F), such as

$1 + 1 = 2$, or
11 is a prime number.

but not

What is your name? or
This sentence is false.

Propositions are denoted by letters such as p, q, r, \dots , and they are combined into compound propositions by *connectives* such as \wedge (and), \vee (or) and \neg (not).

4.1 Connectives \wedge, \vee and \neg

\wedge, \vee and \neg are called “connectives” because they can be used to connect two sentences p and q into one. These particular connectives are defined so that they agree with the most common interpretations of the words “and”, “or” and “not.”

To define $p \wedge q$, for example, we only have to say that $p \wedge q$ is true only when p is true and q is true.

We define \wedge by the following *truth table*:

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

Similarly, $p \vee q$ is true when p is true or q is true, but now we have to be more precise, because “or” has at least two meanings in ordinary speech.

We define \vee by the truth table

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

This is the inclusive sense of “ p or q ” (often written “ p and/or q ” and meaning at least one of p , q is true).

Finally, “not” \neg (also called negation) is defined as follows.

We define \neg by the truth table

p	$\neg p$
T	F
F	T

The connectives \wedge, \vee and \neg , are functions of the propositional variables p and q , which can take the two values T and F. For this reason, \wedge, \vee and \neg are also called *truth functions*.

4.2 Implication

Another important truth function is $p \rightarrow q$, which corresponds to “if p then q ” or “ p implies q ” in ordinary speech.

In ordinary speech the value of $p \rightarrow q$ depends only on what happens when p is true. For example to decide whether

MCG flooded \rightarrow the cricket is off

it is enough to see what happens when the MCG is flooded. *Thus we agree that $p \rightarrow q$ is true when p is false.*

We define \rightarrow by the truth table

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

4.3 Other connectives

Two other important connectives are \leftrightarrow (“if and only if”) and $\underline{\vee}$ (“exclusive or”).

The sentence $p \leftrightarrow q$ is true exactly when the truth values of p and q agree.

We define \leftrightarrow by the truth table

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

We could also write $p \leftrightarrow q$ as $(p \rightarrow q) \wedge (q \rightarrow p)$. We’ll see how to prove this in the next lecture.

The sentence $p \underline{\vee} q$ is true exactly when the truth values of p and q disagree.

We define $\underline{\vee}$ by the truth table

p	q	$p \underline{\vee} q$
T	T	F
T	F	T
F	T	T
F	F	F

4.4 Remarks

1. The symbols \wedge and \vee are intentionally similar to the symbols \cap and \cup for set intersection and union because

$$x \in A \cap B \Leftrightarrow (x \in A) \wedge (x \in B)$$

$$x \in A \cup B \Leftrightarrow (x \in A) \vee (x \in B)$$

(We study sets later.)

2. The “exclusive or” function $\underline{\vee}$ is written XOR in some programming languages.

3. If we write 0 for F and 1 for T then $\underline{\vee}$ becomes the function

p	q	$p \underline{\vee} q$
1	1	0
1	0	1
0	1	1
0	0	0

This is also known as the “mod 2 sum”, because $1 + 1 = 2 \equiv 0 \pmod{2}$. (It could also be called the “mod 2 difference” because $a + b$ is the same as $a - b \pmod{2}$).

4. The mod 2 sum occurs in many homes where two switches p, q control the same light. The truth value of $p \underline{\vee} q$ tells whether the light is on or not, and the light can be switched to the opposite state by switching the value of either p or q .

Questions

4.1 Which of the following are propositions?

$$1 + 1 = 3, \quad 1 + 1, \quad 3 \text{ divides } 7, \quad 3 \div 7$$

4.2 Let f be the proposition “foo” and let b be the proposition “bar”. Write the following propositions in symbols, using f, b, \rightarrow and \neg .

- if foo, then bar.
- bar if foo.
- bar only if foo.
- foo implies not bar.
- foo is sufficient for bar.
- foo is necessary for bar.

4.3 In the following examples, is the “or” intended to be inclusive or exclusive?

- Would you like coffee or tea?
- Oranges or lemons are a good source of vitamin C.
- He will arrive in a minute or two.

Lecture 5: Tautologies and logical equivalence

A major problem in logic is to recognise statements that are “always true” or “always false”.

5.1 Tautologies and contradictions

A *sentence* ϕ in propositional logic is a formula with variables p, q, r, \dots which can take the values T and F. The possible *interpretations* of ϕ are all possible assignments of values to its variables.

A sentence in propositional logic is

- a *tautology* if it has value T under all interpretations;
- a *contradiction* if it has value F under all interpretations.

We can check whether ϕ is a tautology, a contradiction, or neither by computing its value for all possible values of its variables.

Example. $(\neg p) \vee p$

For $p = \text{T}$ we have

$$(\neg \text{T}) \vee \text{T} = \text{F} \vee \text{T} = \text{T},$$

using the known values of the \neg and \vee functions.

For $p = \text{F}$ we have

$$(\neg \text{F}) \vee \text{F} = \text{T} \vee \text{F} = \text{T},$$

hence $(\neg p) \vee p$ is a tautology. (It is known as the *law of the excluded middle*).

We can similarly compute the values of any truth function ϕ , so this is an algorithm for recognising tautologies. However, if ϕ has n variables, they have 2^n sets of values, so the amount of computation grows rapidly with n . One of the biggest unsolved problems of logic and computer science is to find an efficient algorithm for recognising tautologies.

5.2 Logical equivalence

Sentences ϕ and ψ are *logically equivalent* if they are the same truth function, which also means $\phi \leftrightarrow \psi$ is a tautology. This relation between sentences is written $\phi \Leftrightarrow \psi$ or $\phi \equiv \psi$.

Example. $p \rightarrow q \equiv (\neg p) \vee q$

We know $p \rightarrow q$ has the truth table

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Now $(\neg p) \vee q$ has the truth table

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

So $p \rightarrow q$ and $(\neg p) \vee q$ have the same truth table (looking just at their columns). It follows from this that $p \rightarrow q$ can always be rewritten as $(\neg p) \vee q$. In fact, all truth functions can be expressed in terms of \wedge , \vee , and \neg .

This is like finding identities in algebra – one uses known equivalences to rearrange, expand and simplify.

5.3 Useful equivalences

The following equivalences are the most frequently used in this “algebra of logic”.

Equivalence law

$$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$$

Implication law

$$p \rightarrow q \equiv (\neg p) \vee q$$

Double Negation law

$$\neg \neg p \equiv p$$

Idempotent laws

$$p \wedge p \equiv p$$

$$p \vee p \equiv p$$

Commutative laws

$$p \wedge q \equiv q \wedge p$$

$$p \vee q \equiv q \vee p$$

Associative laws

$$p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$$

$$p \vee (q \vee r) \equiv (p \vee q) \vee r$$

Distributive laws

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

De Morgan's laws

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

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

Identity laws

$$p \wedge \top \equiv p$$

$$p \vee \text{F} \equiv p$$

Annihilation laws

$$p \wedge \text{F} \equiv \text{F}$$

$$p \vee \top \equiv \top$$

Inverse laws

$$p \wedge (\neg p) \equiv \text{F}$$

$$p \vee (\neg p) \equiv \top$$

Absorption laws

$$p \wedge (p \vee q) \equiv p$$

$$p \vee (p \wedge q) \equiv p$$

Remarks

1. The commutative laws are used to rearrange terms, as in ordinary algebra. The law $p \vee q \equiv q \vee p$ is like $p + q = q + p$ in ordinary algebra, and $p \wedge q \equiv q \wedge p$ is like $pq = qp$.

2. The associative laws are used to remove

brackets. Since $p \vee (q \vee r) \equiv (p \vee q) \vee r$, we can write either side as $p \vee q \vee r$. This is like $p + (q + r) = (p + q) + r = p + q + r$ in ordinary algebra.

3. The distributive laws are used to “expand” combinations of \wedge and \vee .

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

is like

$$p(q + r) = pq + pr.$$

The other distributive law is *not* like anything in ordinary algebra.

4. Some of these laws are redundant, in the sense that other laws imply them. For example, the absorption law

$$p \wedge (p \vee q) \equiv p$$

follows from the distributive, idempotent, identity and annihilation laws:

$$\begin{aligned} p \wedge (p \vee q) &\equiv (p \wedge p) \vee (p \wedge q) \\ &\quad \text{by distributive law} \\ &\equiv p \vee (p \wedge q) \\ &\quad \text{by idempotent law} \\ &\equiv (p \wedge \top) \vee (p \wedge q) \\ &\quad \text{by identity law} \\ &\equiv p \wedge (\top \vee q) \\ &\quad \text{by distributive law} \\ &\equiv p \wedge \top \\ &\quad \text{by annihilation law} \\ &\equiv p \\ &\quad \text{by identity law} \end{aligned}$$

Questions

5.1 Explain why there are 8 ways to assign truth values to variables p, q, r ; 16 ways to assign truth values to variables p, q, r, s ; and in general 2^n ways to assign truth values to n variables.

5.2 Use truth tables to verify the de Morgan's laws and absorption laws.

5.3 If $p \vee\vee q$ is the “exclusive or” discussed last lecture, see whether it satisfies the distributive laws

$$p \wedge (q \vee\vee r) \equiv (p \wedge q) \vee\vee (p \wedge r)$$

$$p \vee\vee (q \wedge r) \equiv (p \vee\vee q) \wedge (p \vee\vee r)$$

Lecture 6: Rules of inference

Last time we saw how to recognise tautologies and logically equivalent sentences by computing their truth tables. Another way is to *infer* new sentences from old by *rules of inference*.

6.1 The rule of replacement

This rule says that any sentence may be replaced by an equivalent sentence. Any series of such replacements therefore leads to a sentence equivalent to the one we started with.

Using replacement is like the usual method of proving identities in algebra – make a series of replacements until the left hand side is found equal to the right hand side.

Examples

1. $x \rightarrow y \equiv (\neg y) \rightarrow (\neg x)$

Proof.

$$\begin{aligned} x \rightarrow y &\equiv (\neg x) \vee y && \text{by implication law} \\ &\equiv y \vee (\neg x) && \text{by commutative law} \\ &\equiv (\neg \neg y) \vee (\neg x) && \text{by law of double negation} \\ &\equiv (\neg y) \rightarrow (\neg x) && \text{by implication law} \quad \square \end{aligned}$$

$$x \rightarrow y \equiv (\neg y) \rightarrow (\neg x)$$

$(\neg y) \rightarrow (\neg x)$ is the *contrapositive* of $x \rightarrow y$.

Example. The contrapositive of

MCG flooded \rightarrow cricket is off

is

Cricket is on \rightarrow MCG not flooded.

An implication and its contrapositive are equivalent: they mean the same thing!

2. $p \rightarrow (q \rightarrow p) \equiv \top$

Proof.

$$\begin{aligned} p \rightarrow (q \rightarrow p) &\equiv (\neg p) \vee (q \rightarrow p) && \text{by implication law} \\ &\equiv (\neg p) \vee ((\neg q) \vee p) && \text{by implication law} \\ &\equiv (\neg p) \vee (p \vee (\neg q)) && \text{by commutative law} \\ &\equiv ((\neg p) \vee p) \vee (\neg q) && \text{by associative law} \\ &\equiv (p \vee (\neg p)) \vee (\neg q) && \text{by commutative law} \\ &\equiv \top \vee (\neg q) && \text{by inverse law} \\ &\equiv \top && \text{by annihilation law} \quad \square \end{aligned}$$

3. $((p \rightarrow q) \wedge p) \rightarrow q \equiv \top$

Proof.

$$\begin{aligned} ((p \rightarrow q) \wedge p) \rightarrow q &\equiv \neg((p \rightarrow q) \wedge p) \vee q && \text{by implication law} \\ &\equiv (\neg(p \rightarrow q) \vee (\neg p)) \vee q && \text{by de Morgan's law} \\ &\equiv \neg(p \rightarrow q) \vee ((\neg p) \vee q) && \text{by associative law} \\ &\equiv \neg(p \rightarrow q) \vee (p \rightarrow q) && \text{by implication law} \\ &\equiv (p \rightarrow q) \vee \neg(p \rightarrow q) && \text{by commutative law} \\ &\equiv \top && \text{by inverse law} \quad \square \end{aligned}$$

This tautology says that “if p implies q and p is true then q is true”.

6.2 Modus ponens

The tautology $((p \rightarrow q) \wedge p) \rightarrow q$ also translates into a rule of inference known as *modus ponens*: from sentences $p \rightarrow q$ and p we can infer the sentence q .

6.3 Logical consequence

A sentence ψ is a *logical consequence* of a sentence ϕ , if $\psi = \text{T}$ whenever $\phi = \text{T}$. We write this as $\phi \Rightarrow \psi$.

It is the same to say that $\phi \rightarrow \psi$ is a tautology, but $\phi \Rightarrow \psi$ makes it clearer that we are discussing a relation between the sentences ϕ and ψ .

Any sentence ψ logically *equivalent* to ϕ is a logical consequence of ϕ , but not all consequences of ψ are equivalent to it.

Example. $p \wedge q \Rightarrow p$

p is a logical consequence of $p \wedge q$, because $p = \text{T}$ whenever $p \wedge q = \text{T}$. However, we can have $p \wedge q = \text{F}$ when $p = \text{T}$ (namely, when $q = \text{F}$). Hence $p \wedge q$ and p are not equivalent.

This example shows that \Rightarrow is not symmetric:

$$(p \wedge q) \Rightarrow p \quad \text{but} \quad p \not\Rightarrow (p \wedge q)$$

This is where \Rightarrow differs from \equiv , because if $\phi \equiv \psi$ then $\psi \equiv \phi$.

In fact, we build the relation \equiv from \Rightarrow the same way \leftrightarrow is built from \rightarrow :

$$\phi \equiv \psi \quad \text{means} \quad (\phi \Rightarrow \psi) \text{ and } (\psi \Rightarrow \phi).$$

Questions

- 6.1** The slogan “no pain, no gain” stands for an implication. What is it?
- 6.2** What is the contrapositive of “no pain, no gain”?
- 6.3** Write the following sentences as implications, and then write their contrapositives.
- You can’t make an omelette without breaking eggs.
 - If n is even, so is n^2
 - Haste makes waste.
- 6.4** Show that $p \rightarrow (q \rightarrow (r \rightarrow p))$ is a tautology using the laws of logic.

- 6.5** Find a tautology with n variables which is $p \rightarrow (q \rightarrow p)$ for $n = 2$ and $p \rightarrow (q \rightarrow (r \rightarrow p))$ for $n = 3$.

Lecture 7: Predicates and quantifiers

We get a more expressive language than propositional logic by admitting *predicates* like

$$P(n), \quad Q(x, y), \quad R(a, b, c)$$

These stand for properties or relations such as

$$\begin{aligned} P(n) &: n \text{ is prime} \\ Q(x, y) &: x \leq y \\ R(a, b, c) &: a + b = c. \end{aligned}$$

Those with one variable, such as “ n is prime,” are usually called *properties*, while those with two or more variables, such as “ $x \leq y$,” are usually called *relations*.

7.1 Predicates

A predicate such as “ n is prime” is not a sentence because it is neither true nor false. Rather, it is a function $P(n)$ of n with the Boolean values T (true) or F (false). In this case, $P(n)$ is a function of natural numbers defined by

$$P(n) = \begin{cases} \text{T} & \text{if } n \text{ is prime} \\ \text{F} & \text{otherwise.} \end{cases}$$

Similarly, the “ $x \leq y$ ” predicate is a function of pairs of real numbers, defined by

$$R(x, y) = \begin{cases} \text{T} & \text{if } x \leq y \\ \text{F} & \text{otherwise.} \end{cases}$$

Since most of mathematics involves properties and relations such as these, only a language with predicates is adequate for mathematics (and computer science).

7.2 Building sentences from predicates

One way to create a sentence from a predicate is to replace its variables by constants. For example, when $P(n)$ is the predicate “ n is prime,” $P(3)$ is the sentence “3 is prime.”

Another way is to use *quantifiers*:

- \forall (meaning “for all”) and
- \exists (meaning “there exists” or “there is”).

Example. $\exists n P(n)$ is the (true) sentence

there exists an n such that n is prime.

$\forall n P(n)$ is the (false) sentence

for all n , n is prime.

Note that when $\exists n$ is read “there exists an n ” we also add a “such that.”

7.3 Quantifiers and connectives

We can also combine quantifiers with connectives from propositional logic.

Example. Let $Sq(n)$ be the predicate “ n is a square,” and let $Pos(n)$ be the predicate “ n is positive” as above. Then we can symbolise the following sentences:

There is a positive square:

$$\exists n (Pos(n) \wedge Sq(n)).$$

There is a positive integer which is not a square:

$$\exists n (Pos(n) \wedge \neg Sq(n))$$

All squares are positive:

$$\forall n (Sq(n) \rightarrow Pos(n))$$

Notice that the “All...are” combination in English actually involves an implication. This is needed because we are making a claim only about squares and the implication serves to “narrow down” the set we are examining.

7.4 Alternating quantifiers

Combinations of quantifiers like $\forall x \exists y \dots$, “for all x there is a $y \dots$ ” are common in mathematics, and can be confusing. It helps to have some examples in mind to recall the difference between $\forall x \exists y \dots$ and $\exists y \forall x \dots$.

The relation $x < y$ is convenient to illustrate such combinations; we write $x < y$ as the predicate $L(x, y)$

Then

$$\forall x \exists y L(x, y)$$

is the (true) sentence

for all x there is a y such that $x < y$,

which says that there is no greatest number.

But with the opposite combination of quantifiers we have

$$\exists y \forall x L(x, y)$$

is the false sentence

there is a y such that for all x , $x < y$,

which says there is a number greater than all numbers.

Even though these statements are usually written without brackets they are effectively bracketed “from the centre”. So $\forall x \exists y L(x, y)$ means $\forall x (\exists y L(x, y))$ and $\exists y \forall x L(x, y)$ means $\exists y (\forall x L(x, y))$.

7.5 An example from Abraham Lincoln

You can fool all of the people some of the time
and

you can fool some of the people all of the time
but

you can't fool all of the people all of the time.

Let $F(p, t)$ be the predicate:

person p can be fooled at time t .

Then

$\forall p \exists t F(p, t)$ says

you can fool all of the people some of the time,

$\exists p \forall t F(p, t)$ says

you can fool some of the people all of the time,

$\neg \forall p \exists t F(p, t)$ says

you can't fool all of the people all of the time.

Hence Lincoln's sentence in symbols is:

$$\forall p \exists t F(p, t) \wedge \exists p \forall t F(p, t) \wedge \neg \forall p \exists t F(p, t)$$

Remark. Another way to say “you can't fool all of the people all of the time” is

$$\exists p \exists t \neg F(p, t).$$

Questions

7.1 Write “roses are red” in the language of predicate logic, using

$\text{rose}(x)$ for “ x is a rose”

$\text{red}(x)$ for “ x is red.”

7.2 If $P(n)$ stands for “ n is prime” and $E(n)$ stands for “ n is even,” what does $P(n) \wedge (\neg E(n))$ say about n ?

7.3 Using the predicates

$\text{pol}(x)$ for “ x is a politician”

$\text{liar}(x)$ for “ x is a liar”

- all politicians are liars
- some politicians are liars
- no politicians are liars
- some politicians are not liars.

Are any of these sentences logically equivalent?

Lecture 8: Predicate logic

8.1 Valid sentences

The language of predicate logic is based on predicate symbols, variables, constants, brackets, \forall, \exists and connectives. The examples from last lecture illustrate how these ingredients are used to form sentences.

A sentence in predicate logic is *valid* if it has value \top under all interpretations.

This is similar to the definition of a tautology in propositional logic. But now “all interpretations” means all interpretations of the predicate symbols, which is more complicated. The interpretation of a symbol $P(n)$, say, must include both the range of the variable n , as well as saying those n for which $P(n)$ is true.

8.2 Interpretations

For example, one interpretation of $P(n)$ is “ n is positive,” where n ranges over the real numbers. Under this interpretation, $\forall n P(n)$ is false.

A different interpretation of $P(n)$ is “ n is positive,” where n ranges over the numbers > 2 . Under this interpretation, $\forall n P(n)$ is true.

Unlike in propositional logic, there are infinitely many different interpretations of each formula. Thus there is no truth table method for predicate logic. We cannot decide whether a formula is valid by testing all interpretations.

8.3 Recognising valid sentences

Nevertheless, in some cases, we can see that a sentence is true for all interpretations.

Example. $\forall x P(x) \rightarrow \exists x P(x)$ is true for all properties P , and hence is valid.

Likewise, we can sometimes see that a sentence is *not* valid by finding an interpretation which makes it false.

Example. The sentence

$$\forall x \exists y Q(x, y) \rightarrow \exists x \forall y Q(x, y)$$

is false if we interpret $Q(x, y)$ as $x \leq y$ on the real numbers. With this interpretation

$$\forall x \exists y Q(x, y) \text{ is true}$$

(for any number there is a larger number), but

$$\exists x \forall y Q(x, y) \text{ is false}$$

(there is no number \leq all numbers). Hence the implication is false.

8.4 Consequence and equivalence

As in propositional logic, a sentence ψ is a *logical consequence* of a sentence ϕ if any interpretation which makes ϕ true makes ψ true. Again we write $\phi \Rightarrow \psi$ if ψ is a consequence of ϕ , and this is the same as saying $\phi \rightarrow \psi$ is valid.

Example. Any interpretation which makes $\forall n P(n)$ true makes $\exists n P(n)$ true, and this is why $\forall x P(x) \rightarrow \exists x P(x)$ is valid.

Similarly, sentences ψ and ϕ are equivalent, written $\psi \equiv \phi$, if each is a consequence of the other. Some sentences are equivalent for “propositional logic reasons.”

Example. We have

$$\forall x (P(x) \wedge Q(x)) \equiv \forall x (Q(x) \wedge P(x))$$

simply because

$$P(x) \wedge Q(x) \equiv Q(x) \wedge P(x)$$

for any x .

However there are also equivalences that genuinely involve quantifiers.

8.5 Useful equivalences

Two important equivalences involving quantifiers are

$$\neg\forall xP(x) \equiv \exists x\neg P(x)$$

$$\neg\exists xP(x) \equiv \forall x\neg P(x)$$

These make sense intuitively. For example, $\neg\forall xP(x)$ means $P(x)$ is not true for all x , hence there is an x for which $P(x)$ is false, that is, $\exists x\neg P(x)$.

They can also be viewed as “infinite De Morgan’s laws.” If x ranges over $\{1, 2, 3, \dots\}$ for example, then

$$\forall xP(x) \equiv P(1) \wedge P(2) \wedge P(3) \wedge \dots$$

and

$$\exists xP(x) \equiv P(1) \vee P(2) \vee P(3) \vee \dots$$

Hence

$$\begin{aligned}\neg\forall xP(x) &\equiv \neg(P(1) \wedge P(2) \wedge P(3) \wedge \dots) \\ &\equiv (\neg P(1)) \vee (\neg P(2)) \vee (\neg P(3)) \vee \dots \\ &\quad \text{by de Morgan's law} \\ &\equiv \exists x\neg P(x).\end{aligned}$$

And similarly

$$\begin{aligned}\neg\exists xP(x) &\equiv \neg(P(1) \vee P(2) \vee P(3) \vee \dots) \\ &\equiv (\neg P(1)) \wedge (\neg P(2)) \wedge (\neg P(3)) \wedge \dots \\ &\quad \text{by de Morgan's law} \\ &\equiv \forall x\neg P(x).\end{aligned}$$

8.6 Simplification

The infinite de Morgan’s laws allow a certain simplification of predicate formulas by “pushing \neg inside quantifiers.”

Example.

$$\begin{aligned}\neg\forall x\exists yQ(x, y) &\equiv \exists x\neg\exists yQ(x, y) \\ &\equiv \exists x\forall y\neg Q(x, y).\end{aligned}$$

It is in fact possible to transform any quantified statement in predicate logic to an equivalent with all quantifiers at the front.

8.7* Completeness and undecidability

In 1930, Gödel proved that there is a complete set of rules of inference for predicate logic. This means, in particular, that there is an algorithm to list the valid sentences.

However, in 1936, Church and Turing proved that there is no algorithm to list the logically false sentences. This means, in particular, that predicate logic is *undecidable*: there is no algorithm which, for any sentence ϕ , decides whether ϕ is valid or not.

This negative result is due to the power of predicate logic: it can express all mathematical or computational problems, and it is known that some of these problems cannot be solved by algorithm.

Questions

8.1 Give interpretations which make the following sentences false.

$$\begin{aligned}\exists nP(n) \rightarrow \forall nP(n) \\ \forall x\forall y (R(x, y) \rightarrow R(y, x)) \\ \forall m\exists nS(m, n)\end{aligned}$$

8.2 Give interpretations which show that the sentences

$$\exists x (P(x) \wedge L(x))$$

and

$$\exists x (P(x) \wedge \neg L(x))$$

are not equivalent.

8.3 Is $\exists y\forall xR(x, y)$ a logical consequence of $\forall x\exists yR(x, y)$?

If so, explain why. If not, give an interpretation which makes $\forall x\exists yR(x, y)$ true and $\exists y\forall xR(x, y)$ false.

8.4 Is $\forall x\exists yR(x, y)$ a logical consequence of $\exists y\forall xR(x, y)$?

If so, explain why. If not, give an interpretation which makes $\exists y\forall xR(x, y)$ true and $\forall x\exists yR(x, y)$ false.

8.5 Explain why $\neg\forall p\forall tF(p, t) \equiv \exists p\exists t\neg F(p, t)$.

Lecture 9: Mathematical induction

Since the natural numbers $0, 1, 2, 3, \dots$ are generated by a process which begins with 0 and repeatedly adds 1, we have the following.

Property P is true for all natural numbers if

1. $P(0)$ is true.
2. $P(k) \Rightarrow P(k+1)$ for all $k \in \mathbb{N}$.

This is called the *principle of mathematical induction*.

It is used in a style of proof called *proof by induction*, which consists of two steps.

Base step: Proof that the required property P is true for 0.

Induction step: Proof that **if** $P(k)$ is true **then** $P(k+1)$ is true, for each $k \in \mathbb{N}$.

9.1 Examples

Example 1. 3 divides $n^3 + 2n$

Base step. 3 divides $0^3 + 2 \times 0 = 0$, so the required property is true when $n = 0$.

Induction step. We want to prove

$$\begin{aligned} & 3 \text{ divides } k^3 + 2k \\ \Rightarrow & 3 \text{ divides } (k+1)^3 + 2(k+1). \end{aligned}$$

Well,

$$\begin{aligned} & (k+1)^3 + 2(k+1) \\ = & k^3 + 3k^2 + 3k + 1 + 2k + 2 \\ = & k^3 + 2k + 3k^2 + 3k + 3 \\ = & k^3 + 2k + 3(k^2 + k + 1). \end{aligned}$$

Therefore,

$$\begin{aligned} & 3 \text{ divides } k^3 + 2k \\ \Rightarrow & 3 \text{ divides } k^3 + 2k + 3(k^2 + k + 1) \\ \Rightarrow & 3 \text{ divides } (k+1)^3 + 2(k+1) \end{aligned}$$

as required. This completes the induction step, and hence completes the proof.

Example 2. There are 2^n n -letter words using the letters A and B .

Base step. There is $2^0 = 1$ 0-letter word, namely the empty word, so the property is true for $n = 0$.

Induction step. We want to prove that

$$\begin{aligned} & 2^k \text{ } k\text{-letter words using } A \text{ and } B \\ \Rightarrow & 2^{k+1} \text{ } (k+1)\text{-letter words} \\ & \text{using } A \text{ and } B \end{aligned}$$

Well, a $(k+1)$ -letter word is either WA or WB , where W is any k -letter word. Thus if there are 2^k k -letter words W , there are $2 \times 2^k = 2^{k+1}$ $(k+1)$ -letter words.

This completes the induction step, and hence completes the proof.

9.2 Starting the base step higher

It is not always appropriate to start the induction at 0. Some properties are true only from a certain positive integer upwards, in which case the induction starts at that integer.

Example 3. $n! > 2^n$ for $n \geq 4$

Base step. $4! = 4 \times 3 \times 2 = 24 > 16 = 2^4$, so $n! > 2^n$ is true for $n = 4$.

Induction step.

$$\begin{aligned} (k+1)! &= (k+1) \times k! \\ &> 2 \times k! \quad \text{if } k \geq 4. \end{aligned}$$

Hence if $k! > 2^k$ then

$$(k+1)! > 2 \times k! > 2 \times 2^k = 2^{k+1}.$$

Thus $k! > 2^k \Rightarrow (k+1)! > 2^{k+1}$, as required to complete the induction.

Hence $n! > 2^n$ for all $n \geq 4$.

Example 4. Any integer value $n \geq 8$ (in cents) is obtainable with 3c and 5c stamps.

Base step. The value 8c is obtained by a 3c plus a 5c stamp.

Induction step. We have to show that if k cents is obtainable, so is $(k + 1)$ cents, when $k \geq 8$.

Case 1. The k cents is obtained using a 5c stamp (among others). Replace the 5c stamp by two 3c stamps, thus obtaining $k + 1$ cents.

Case 2. If the k cents is obtained using only 3c stamps, there are at least three of them (since $k \geq 8$). In this case, replace three 3c stamps by two 5c stamps, again obtaining $k + 1$ cents.

Thus in either case, when $k \geq 8$,

k cents obtainable $\Rightarrow k + 1$ cents obtainable.

This completes the induction proof that n cents are obtainable from 3c and 5c stamps, for all natural numbers $n \geq 8$.

9.3 Sums of series

Induction is commonly used to prove that sum formulas are correct.

Example 5. $1 + 2 + 3 + \dots + n = n(n + 1)/2$

Base step. When $n = 1$, the left hand side is 1, and the right hand side is $1(1 + 1)/2 = 2/2 = 1$, so the formula is true for $n = 1$.

Induction step. We have to prove that

$$\begin{aligned} 1 + 2 + \dots + k &= \frac{k(k+1)}{2} \\ \Rightarrow 1 + 2 + \dots + k + (k + 1) &= \frac{(k+1)(k+2)}{2}. \end{aligned}$$

In fact

$$\begin{aligned} 1 + 2 + \dots + k &= \frac{k(k+1)}{2} \\ \Rightarrow 1 + 2 + \dots + k + (k + 1) &= \frac{k(k+1)}{2} + (k + 1), \\ \text{adding } (k + 1) \text{ to both sides,} \end{aligned}$$

So we automatically get the correct LHS, and

$$\begin{aligned} \text{RHS} &= (k + 1)\left(\frac{k}{2} + 1\right), \\ &\text{factorising the right hand side,} \\ &= (k + 1)\frac{k+2}{2} = \frac{(k+1)(k+2)}{2}, \end{aligned}$$

as required.

This completes the induction proof that $1 + 2 + 3 + \dots + n = n(n + 1)/2$.

Remark. Another proof is to write down

$$\begin{array}{r} 1 + 2 + 3 + \dots + (n - 1) + n \\ n + (n - 1) + \dots + 3 + 2 + 1 \end{array}$$

and observe that each of the n columns sums to $n + 1$. Thus the sum of twice the series is $n(n + 1)$, and hence the sum of the series itself is $n(n + 1)/2$. This proof uses induction unconsciously, to prove that the sum of each column is the same.

Questions

In most induction problems set for students we skip the experimental part, which is *finding what to prove*. Before trying to prove that 3 divides $n^3 + 2n$, for example, someone needs to guess that it is true, perhaps by trying $n = 1, 2, 3, 4$.

9.1 In this question, try to guess what ? stands for, by trying a few values of n .

- ? divides $n^2 + n$
- The sum of the first n odd numbers is ?
- $\frac{1}{1 \times 2} + \frac{1}{2 \times 3} + \frac{1}{3 \times 4} + \dots + \frac{1}{n(n+1)} = 1 - ?$

9.2 If you correctly guessed the sum

$$\frac{1}{1 \times 2} + \frac{1}{2 \times 3} + \frac{1}{3 \times 4} + \dots + \frac{1}{n(n+1)},$$

you might wonder why it is so simple. Here is a clue:

$$\frac{1}{1 \times 2} = \frac{1}{1} - \frac{1}{2}.$$

What is $\frac{1}{2 \times 3}$? $\frac{1}{3 \times 4}$?

How does this lead to a simple formula for

$$\frac{1}{1 \times 2} + \frac{1}{2 \times 3} + \frac{1}{3 \times 4} + \dots + \frac{1}{n(n+1)}?$$

OK, if we can guess formulas correctly, why bother proving them by induction? The reason is that a statement which fits many values of n can still be wrong.

9.3 Show that $n^2 + n + 41$ is a prime number for $n = 1, 2, 3, 4$ (and go further, if you like). Do you think $n^2 + n + 41$ is prime for all natural numbers n ?

Lecture 10: Induction and well-ordering

In the previous lecture we were able to prove a property P holds for $0, 1, 2, \dots$ as follows:

Base step. Prove $P(0)$

Induction step. Prove $P(k) \Rightarrow P(k+1)$ for each natural number k .

This is sufficient to prove that $P(n)$ holds for all natural numbers n , but it may be difficult to prove that $P(k+1)$ follows from $P(k)$. It may in fact be easier to prove the induction step

$$P(0) \wedge P(1) \wedge \dots \wedge P(k) \Rightarrow P(k+1).$$

That is, it may help to assume P holds for *all* numbers before $k+1$. Induction with this style of induction step is sometimes called the *strong form* of mathematical induction.

10.1 Examples of strong induction

Example 1. Every integer ≥ 2 is a product of primes.

Base step. 2 is a prime, hence a product of (one) prime.

Induction step. Suppose $2, 3, \dots, k$ are products of primes. We wish to prove that $k+1$ is a product of primes.

This is certainly true if $k+1$ is a prime. If not

$$k+1 = i \times j,$$

for some natural numbers i and j less than $k+1$. But then i and j are products of primes by our assumption, hence so is $i \times j = k+1$.

Example 2. Every integer ≥ 1 is a sum of distinct powers of 2.

The idea behind this proof is to repeatedly subtract the largest possible power of 2. We illus-

trate with the number 27.

$$27 - \text{largest power of 2 less than 27}$$

$$= 27 - 16 = 11$$

$$11 - \text{largest power of 2 less than 11}$$

$$= 11 - 8 = 3$$

$$3 - \text{largest power of 2 less than 3}$$

$$= 3 - 2 = 1$$

Hence $27 = 16 + 8 + 2 + 1 = 2^4 + 2^3 + 2^1 + 2^0$.

(It is only interesting to find *distinct* powers of 2, because of course each integer ≥ 1 is a sum of 1s, and $1 = 2^0$.)

The strong induction proof goes as follows.

Base step. $1 = 2^0$, so 1 is a sum of (one) power of 2.

Induction step. Suppose each of the numbers $1, 2, 3, \dots, k$ is a sum of distinct powers of 2. We wish to prove that $k+1$ is a sum of distinct powers of 2.

This is certainly true if $k+1$ is a power of 2. If not, let 2^j be the greatest power of 2 less than $k+1$. Then

$$i = k+1 - 2^j$$

is one of the numbers $1, 2, 3, \dots, k$, and hence it is a sum of distinct powers of 2.

Also, the powers of 2 that sum to i are all less than 2^j , otherwise 2^j is less than half $k+1$, contrary to the choice of 2^j as the largest power of 2 less than $k+1$.

Hence $k+1 = 2^j +$ powers of 2 that sum to i is a sum of distinct powers of 2.

This completes the induction.

10.2 Well-ordering and descent

Induction expresses the fact that each natural number n can be reached by starting at 0 and going upwards (e.g. adding 1) a finite number of times.

Equivalent facts are that it is only a finite number of steps *downwards* from any natural

number to 0, that *any descending sequence of natural numbers is finite*, and that *any set of natural numbers has a least member*.

This property is called *well-ordering* of the natural numbers. It is often convenient to arrange a proof to “work downwards” and appeal to well-ordering by saying that the process of working downwards must eventually stop.

Such proofs are equivalent to induction, though they may be called “infinite descent” or some such name.

10.3 Proofs by descent

Example 1. Existence of a prime divisor

If n is any natural number ≥ 2 , then n has a prime divisor.

Proof. If n is prime, then it is a prime divisor of itself. If not, let $n_1 < n$ be a divisor of n .

If n_1 is prime, it is a prime divisor of n . If not, let $n_2 < n_1$ be a divisor of n_1 (and hence of n).

If n_2 is prime, it is a prime divisor of n . If not, let $n_3 < n_2$ be a divisor of n_2 , etc.

The sequence $n > n_1 > n_2 > n_3 > \dots$ must eventually terminate, and this means we find a prime divisor of n .

Example 2. Irrationality of $\sqrt{2}$

Suppose that $\sqrt{2} = m/n$ for natural numbers m and n . Since the square of an odd number is odd, we can argue as follows

$$\begin{aligned} \sqrt{2} &= m/n \\ \Rightarrow 2 &= m^2/n^2 \\ &\text{squaring both sides} \\ \Rightarrow m^2 &= 2n^2 \\ \Rightarrow m^2 &\text{ is even} \\ \Rightarrow m &\text{ is even} \\ &\text{since the square of an odd number is odd} \\ \Rightarrow m &= 2m_1 \text{ say} \\ \Rightarrow 2n^2 &= m^2 = 4m_1^2 \\ \Rightarrow n^2 &= 2m_1^2 \\ \Rightarrow n &\text{ is even, } = 2n_1 \text{ say} \end{aligned}$$

But then $\sqrt{2} = m_1/n_1$, and we can repeat the argument to show that m_1 and n_1 are both even, so $m_1 = 2m_2$ and $n_1 = 2n_2$, and so on.

Since the argument can be repeated indefinitely, we get an *infinite* descending sequence of

natural numbers

$$m > m_1 > m_2 > \dots$$

which is impossible.

Hence there are no natural numbers m and n with $\sqrt{2} = m/n$.

Questions

10.1 For each of the following statements, say which is likely to require strong induction for its proof.

- $1 + a + a^2 + \dots + a^n = \frac{a^{n+1}-1}{a-1}$
- $\neg(p_1 \vee p_2 \vee p_3 \vee \dots \vee p_n) \equiv (\neg p_1) \wedge (\neg p_2) \wedge (\neg p_3) \wedge \dots \wedge (\neg p_n)$
- Each fraction $\frac{n}{m} < 1$ is a sum of distinct fractions with numerator 1 (for example, $\frac{11}{12} = \frac{1}{2} + \frac{1}{3} + \frac{1}{12}$).

10.2 There is something else which tells you every integer ≥ 1 is a sum of distinct powers of 2. What is it?

10.3 Is every integer ≥ 1 a sum of distinct powers of 3?

Lecture 11: Sets

Sets are vital in expressing mathematics formally and are also very important data structures in computer science.

A set is basically just an unordered collection of distinct objects, which we call its *elements* or *members*. Note that there is no notion of order for a set, even though we often write down its elements in some order for convenience. Also, there is no notion of multiplicity: an object is either in a set or not – it cannot be in the set multiple times.

Sets A and B are equal when every element of A is an element of B and vice-versa.

11.1 Set notation

- $x \in S$ means x is an element of set S .
- $\{x_1, x_2, x_3, \dots\}$ is the set with elements x_1, x_2, x_3, \dots
- $\{x : P(x)\}$ is the set of all x with property P .

Example.

$$17 \in \{x : x \text{ is prime}\} = \{2, 3, 5, 7, 11, 13, \dots\}$$

$$\{1, 2, 3\} = \{3, 1, 2\}$$

$$\{1, 1, 1\} = \{1\}$$

For a finite set S , we write $|S|$ for the number of elements of S .

11.2 Universal set

The idea of a “set of all sets” leads to logical difficulties. Difficulties are avoided by always working within a local “universal set” which includes only those objects under consideration.

For example, when discussing arithmetic it might be sufficient to work just with the numbers $0, 1, 2, 3, \dots$. Our universal set could then be taken as

$$\mathbb{N} = \{0, 1, 2, 3, \dots\},$$

and other sets of interest, e.g. $\{x : x \text{ is prime}\}$, are parts of \mathbb{N} .

11.3 Subsets

We say that A is a *subset* of B and write $A \subseteq B$ when each element of A is an element of B .

Example. The set of primes forms a *subset* of \mathbb{N} , that is $\{x : x \text{ is prime}\} \subseteq \mathbb{N}$.

11.4 Characteristic functions

A subset A of B can be specified by its *characteristic function* χ_A , which tells which elements of B are in A and which are not.

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

Example. The subset $A = \{a, c\}$ of $B = \{a, b, c\}$ has the characteristic function χ_A with

$$\chi_A(a) = 1, \quad \chi_A(b) = 0, \quad \chi_A(c) = 1.$$

We also write this function more simply as

$$\begin{array}{ccc} a & b & c \\ 1 & 0 & 1 \end{array}$$

In fact we can list all characteristic functions on $\{a, b, c\}$, and hence all subsets of $\{a, b, c\}$, by listing all sequences of three binary digits:

characteristic function			subset
a	b	c	
0	0	0	$\{\}$
0	0	1	$\{c\}$
0	1	0	$\{b\}$
0	1	1	$\{b, c\}$
1	0	0	$\{a\}$
1	0	1	$\{a, c\}$
1	1	0	$\{a, b\}$
1	1	1	$\{a, b, c\}$

We could similarly list all the subsets of a four-element set, and there would be $2^4 = 16$ of them, corresponding to the 2^4 sequences of 0s and 1s.

In the same way, we find that an n -element set has 2^n subsets, because there are 2^n binary sequences of length n . (Each of the n places in the sequence can be filled in two ways.)

11.5 Power set

The set of all subsets of a set U is called the **power set** $\mathcal{P}(U)$ of U .

Example. We see from the previous table that $\mathcal{P}(\{a, b, c\})$ is the set

$\{\{\}, \{c\}, \{b\}, \{b, c\}, \{a\}, \{a, c\}, \{a, b\}, \{a, b, c\}\}$.

If U has n elements, then $\mathcal{P}(U)$ has 2^n elements.

(The reason $\mathcal{P}(U)$ is called the “power” set is probably that the number of its elements is this power of 2. In fact, the power set of U is sometimes written 2^U .)

11.6 Sets and properties

We mentioned at the beginning that $\{x : P(x)\}$ stands for the set of objects x with property P . Thus sets correspond to properties.

Properties of the natural numbers $0, 1, 2, 3, \dots$, for example, correspond to subsets of the set $\mathbb{N} = \{0, 1, 2, 3, \dots\}$. Thus the subset

$$\{0, 2, 4, 6, \dots\} = \{n \in \mathbb{N} : n \text{ is even}\},$$

so it corresponds to the property of being even. Similarly, the set

$$\{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, \dots\}$$

corresponds to the property of being prime. The power set $\mathcal{P}(\mathbb{N})$ corresponds to all possible properties of natural numbers.

11.7* What are numbers?

“Everything is a set” in mathematics. This claim can be illustrated by defining the numbers $0, 1, 2, 3, \dots$ as particular sets, starting with the empty set. This definition is due to von Neumann.

$$\begin{aligned} 0 &= \{\} \\ 1 &= \{0\} \\ 2 &= \{0, 1\} \\ &\vdots \\ n+1 &= \{0, 1, 2, \dots, n\} \end{aligned}$$

We are not going to use this definition in this course. Still, it is interesting that numbers *can* be defined in such a simple way.

Questions

11.1 Suppose $E(x)$ stands for “ x is even” and $F(x)$ stands for “5 divides x .”

- What is the set $\{x : E(x) \wedge F(x)\}$?
- Write a formula using $E(x)$ and $F(x)$ which describes the set $\{5, 15, 25, 35, \dots\}$.

11.2 How many subsets does the set $\{2, 5, 10, 20\}$ have?

11.3 Consider the infinitely many sets

$$\begin{aligned} \{x : 0 < x < 1\} \\ \{x : 0 < x < \tfrac{1}{2}\} \\ \{x : 0 < x < \tfrac{1}{3}\} \\ \{x : 0 < x < \tfrac{1}{4}\} \\ &\vdots \end{aligned}$$

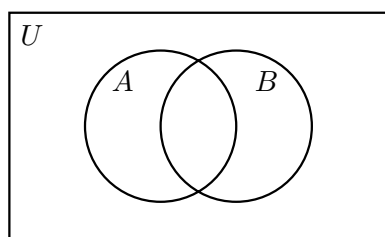
Do they have any element in common?

Lecture 12: Operations on sets

There is an “arithmetic” of sets similar to ordinary arithmetic. There are operations similar to addition, subtraction and multiplication.

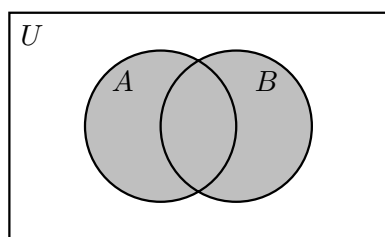
12.1 Venn diagrams

The simple operations on sets can be visualised with the help of *Venn diagrams*, which show sets A, B, C, \dots as disks within a rectangle representing the universal set U .



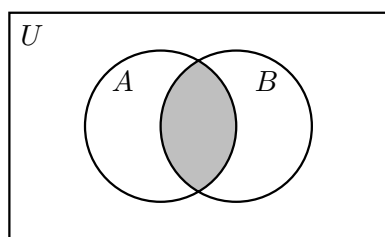
12.2 Union $A \cup B$

The union $A \cup B$ of sets A and B consists of the elements in A or B , and is indicated by the shaded region in the following Venn diagram.



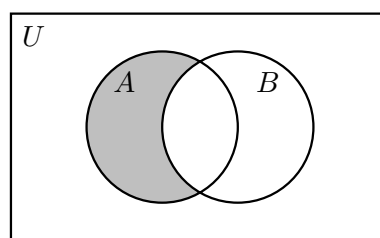
12.3 Intersection $A \cap B$

The intersection $A \cap B$ of sets A and B consists of the elements in A and B , indicated by the shaded region in the following Venn diagram.

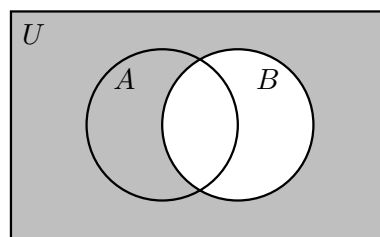


12.4 Difference $A - B$

The difference $A - B$ of sets A and B consists of the elements in A and *not* in B , indicated by the shaded region in the following Venn diagram.

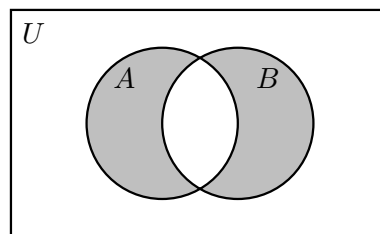


The difference $U - B$ relative to the universal set U is called the *complement* \bar{B} of B . Here is the Venn diagram of \bar{B} .



12.5 Symmetric difference $A \Delta B$

The union of $A - B$ and $B - A$ is called the *symmetric difference* $A \Delta B$ of A and B .



$A \Delta B$ consists of the elements of *one of* A, B but not the other.

It is clear from the diagram that we have not only

$$A \Delta B = (A - B) \cup (B - A),$$

but also

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

12.6 Ordered Pairs

Sometimes we do want order to be important. In computer science arrays are ubiquitous examples of ordered data structures. In maths, *ordered pairs* are frequently used. An ordered pair (a, b) consists simply of a first object a and a second object b . The objects a and b are sometimes called the *entries* or *coordinates* of the ordered pair.

Two ordered pairs (a, b) and (c, d) are equal if and only if $a = b$ and $c = d$.

Example. $\{0, 1\} = \{1, 0\}$ but $(0, 1) \neq (1, 0)$.

There's no reason we need to stop with pairs. We can similarly define ordered triples, quadruples, and so on. When there are k coordinates, we call the object an *ordered k -tuple*. Two ordered k -tuples are equal if and only if their i th coordinates are equal for $i = 1, 2, \dots, k$.

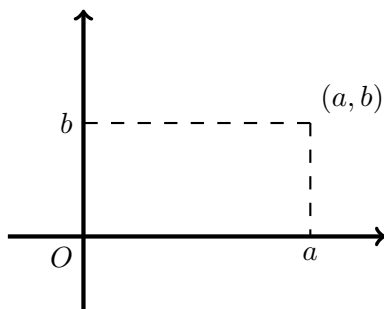
12.7 Cartesian product $A \times B$

The set of ordered pairs

$$A \times B = \{(a, b) : a \in A \text{ and } b \in B\}$$

is the *cartesian product* of sets A and B .

The commonest example is where $A = B = \mathbb{R}$ (the set of real numbers, or the number line). Then the pairs (a, b) are points in the plane, so $\mathbb{R} \times \mathbb{R}$ is the plane.



Because Descartes used this idea in geometry, the cartesian product is named after him.

12.8 $A \times B$ and multiplication

If A has $|A|$ elements and B has $|B|$ elements, then $A \times B$ has $|A| \times |B|$ elements.

Similarly, if L is a line of length l , and W is a line of length w , then $L \times W$ is a rectangle of

area $l \times w$. In fact, we call it an “ $l \times w$ rectangle.” This is probably the reason for using the \times sign, and for calling $A \times B$ a “product.”

Questions

12.1 Draw a Venn diagram for $A \cap \overline{B}$. What is another name for this set?

12.2 Check the de Morgan laws by drawing Venn diagrams for $\overline{A \cup B}$, $\overline{A} \cap \overline{B}$, $\overline{A \cap B}$ and $\overline{A} \cup \overline{B}$

12.3 Find which of the following is true by drawing suitable Venn diagrams.

$$A \cap (B \triangle C) = (A \cap B) \triangle (A \cap C)?$$

$$A \triangle (B \cap C) = (A \triangle B) \cap (A \triangle C)?$$

12.4 If plane = line \times line, what do you think line \times circle is? What about circle \times circle?

Lecture 13: Functions

A function can be thought of as a “black box” which accepts inputs and, for each input, produces a single output.

13.1 Defining functions via sets

Formally we represent a function f as a set X of possible inputs, a set Y so that every output of f is guaranteed to be in Y , and a set of (input,output) pairs from $X \times Y$. The vital property of a function is that each input gives exactly one output.

A function f consists of a domain X , a codomain Y , and a set of ordered pairs from $X \times Y$ which has exactly one ordered pair (x, y) for each $x \in X$.

When (a, b) is in this set we write $f(a) = b$.

The set of y values occurring in these pairs is the *range* of f .

Note that the range of a function is always a subset of its codomain but they may or may not be equal.

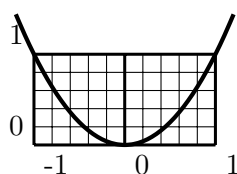
If the range of a function is equal to its codomain, we say the function is *onto*.

Examples.

1. The squaring function $\text{square}(x) = x^2$ with domain \mathbb{R} , codomain \mathbb{R} , and pairs

$$\{(x, x^2) : x \in \mathbb{R}\},$$

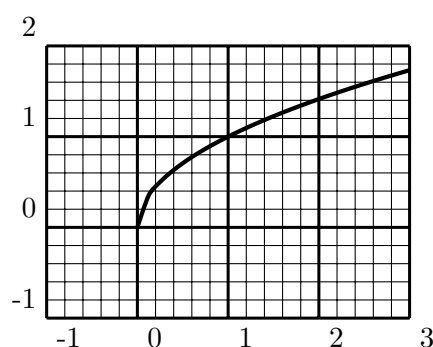
which form what we usually call the *plot* of the squaring function.



The range of this function (the set of y values) is the set $\mathbb{R}^{\geq 0}$ of real numbers ≥ 0 .

2. The square root function $\text{sqrt}(x) = \sqrt{x}$ with domain $\mathbb{R}^{\geq 0}$, codomain \mathbb{R} , and pairs

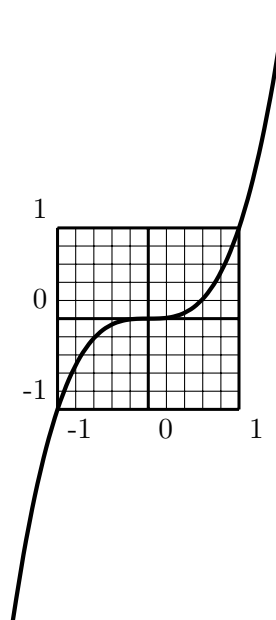
$$\{(x, \sqrt{x}) : x \in \mathbb{R} \text{ and } x \geq 0\}.$$



The range of this function (the set of y values) is the set $\mathbb{R}^{\geq 0}$.

3. The cubing function $\text{cube}(x) = x^3$ with domain \mathbb{R} , codomain \mathbb{R} , and pairs

$$\{(x, x^3) : x \in \mathbb{R}\},$$



The range of this function is the whole of the codomain \mathbb{R} , so it is onto.

13.2 Arrow notation

If f is a function with domain A and codomain B we write

$$f : A \rightarrow B,$$

and we say that f is from A to B .

For example, we could define

$$\text{square} : \mathbb{R} \rightarrow \mathbb{R}.$$

We could also define

$$\text{square} : \mathbb{R} \rightarrow \mathbb{R}^{\geq 0}.$$

Likewise, we could define

$$\text{cube} : \mathbb{R} \rightarrow \mathbb{R}.$$

However we could not define

$$\text{cube} : \mathbb{R} \rightarrow \mathbb{R}^{\geq 0},$$

because for some $x \in \mathbb{R}$, $\text{cube}(x)$ is negative. For example, $\text{cube}(-1) = -1$.

13.3 One-to-one functions

A function $f : X \rightarrow Y$ is *one-to-one* if for each y in the range of f there is only one $x \in X$ such that $f(x) = y$.

For example, the function $\text{cube}(x)$ is one-to-one because each real number y is the cube of exactly one real number x .

The function $\text{square} : \mathbb{R} \rightarrow \mathbb{R}$ is *not* one-to-one because the real number 1 is the square of two different real numbers, 1 and -1 . (In fact each real $y > 0$ is the square of two different real numbers, \sqrt{y} and $-\sqrt{y}$.)

On the other hand, $\text{square} : \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}$ is one-to-one because each real number y in $\mathbb{R}^{\geq 0}$ is the square of only one real number in $\mathbb{R}^{\geq 0}$, namely \sqrt{y} .

The last example shows that the domain of a function is an important part of its description, because changing the domain can change the properties of the function.

13.4 Proving a function is one-to-one

There is an equivalent way of phrasing the definition of one-to-one: a function $f : X \rightarrow Y$ is one-to-one when, for all $x_1, x_2 \in X$,

$$f(x_1) = f(x_2) \Rightarrow x_1 = x_2.$$

This can be useful for proving that some

functions are or are not one-to-one.

Example. The function $f : \mathbb{R} \rightarrow \mathbb{R}$ given by $f(x) = 6x + 2$ is one-to-one because

$$f(x_1) = f(x_2)$$

$$\Rightarrow 6x_1 + 2 = 6x_2 + 2$$

$$\Rightarrow 6x_1 = 6x_2$$

$$\Rightarrow x_1 = x_2.$$

Example. The function $f : \mathbb{R} \rightarrow \mathbb{R}$ given by $f(x) = x^2 + 1$ is not one-to-one because $f(-1) = 2$ and $f(1) = 2$ and so

$$f(-1) = f(1).$$

Questions

13.1 Some of the following “rules” do not define genuine functions. Which are they?

- For each set S of natural numbers, let $f(S)$ be the least member of S .
- For each set X of real numbers between 0 and 1, let $g(X)$ be the least member of X .
- For each circle C in the (x, y) plane, let $h(C)$ be the minimum distance from C to the x -axis.
- For each pair A, B of sets of real numbers, let $s(A, B)$ be the smallest set containing both A and B .
- For each pair A, B of sets of real numbers, let $t(A, B)$ be the largest set contained in both A and B .

13.2 For each of the following, say which can be defined with domain \mathbb{R} and codomain \mathbb{R} .

$$x^2, \quad 1/x, \quad \log x, \quad \sqrt{x}, \quad \sqrt[3]{x}$$

13.3 Find the range of the following functions, given that each has domain \mathbb{R} and codomain \mathbb{R} .

$$e^x, \quad \sin x, \quad \tan^{-1} x, \quad \frac{1}{1+x^2}$$

Lecture 14: Examples of functions

The functions discussed in the last lecture were familiar functions of real numbers. Many other examples occur elsewhere, however.

14.1 Functions of several variables

We might define a function

$$\text{sum} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \quad \text{by} \quad \text{sum}(x, y) = x + y.$$

Because the domain of this function is $\mathbb{R} \times \mathbb{R}$, the inputs to this function are ordered pairs (x, y) of real numbers. Because its codomain in \mathbb{R} , we are guaranteed that each output will be a real number. This function can be thought of as a function of two variables x and y .

Similarly we might define a function

$$\text{binomial} : \mathbb{R} \times \mathbb{R} \times \mathbb{N} \rightarrow \mathbb{R}$$

by

$$\text{binomial}(a, b, n) = (a + b)^n.$$

Here the inputs are ordered triples (x, y, n) such that x and y are real numbers and n is a natural number. We can think of this as a function of three variables.

14.2 Sequences

An infinite sequence of numbers, such as

$$1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \dots,$$

can be viewed as the function $f : \mathbb{N} \rightarrow \mathbb{R}$ defined by $f(n) = 2^{-n}$. In this case, the inputs to f are natural numbers, and its outputs are real numbers.

Any infinite sequence $a_0, a_1, a_2, a_3, \dots$ can be viewed as a function $g(n) = a_n$ from \mathbb{N} to some set containing the values a_n .

14.3 Characteristic functions

A subset of $\mathbb{N} = \{0, 1, 2, 3, \dots\}$ can be represented by its characteristic function. For example, the set of squares is represented by the function $\chi : \mathbb{N} \rightarrow \{0, 1\}$ defined by

$$\chi(n) = \begin{cases} 1 & \text{if } n \text{ is a square} \\ 0 & \text{if } n \text{ is not a square} \end{cases}$$

which has the following sequence of values

110010000100000010000000010000000000100...

(with 1s at the positions of the squares 0, 1, 4, 9, 16, 25, 36, ...).

Any property of natural numbers can likewise be represented by a characteristic function. For example, the function χ above represents the property of being a square.

Thus any set or property of natural numbers is represented by a function

$$\chi : \mathbb{N} \rightarrow \{0, 1\}.$$

Characteristic functions of two or more variables represent relations between two or more objects. For example, the relation $x \leq y$ between real numbers x and y has the characteristic function $\chi : \mathbb{R} \times \mathbb{R} \rightarrow \{0, 1\}$ defined by

$$\chi(x, y) = \begin{cases} 1 & \text{if } x \leq y \\ 0 & \text{otherwise.} \end{cases}$$

14.4 Boolean functions

The connectives \wedge, \vee and \neg are functions of variables whose values come from the set $\mathbb{B} = \{\mathbf{T}, \mathbf{F}\}$ of Boolean values (named after George Boole).

\neg is a function of one variable, so

$$\neg : \mathbb{B} \rightarrow \mathbb{B}$$

and it is completely defined by giving its values on \mathbf{T} and \mathbf{F} , namely

$$\neg \mathbf{T} = \mathbf{F} \quad \text{and} \quad \neg \mathbf{F} = \mathbf{T}.$$

This is what we previously did by giving the

truth table of \neg .

\wedge and \vee are functions of two variables, so

$$\wedge : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$$

and

$$\vee : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$$

They are completely defined by giving their values on the pairs $\{\mathbf{T}, \mathbf{T}\}, \{\mathbf{T}, \mathbf{F}\}, \{\mathbf{F}, \mathbf{T}\}, \{\mathbf{F}, \mathbf{F}\}$ in $\mathbb{B} \times \mathbb{B}$, which is what their truth tables do.

14.5* Characteristic functions and subsets of \mathbb{N}

Mathematicians say that two (possibly infinite) sets A and B have the same *cardinality* (size) if there is a one-to-one and onto function from A to B . This function associates each element of A with a unique element of B and vice-versa. With this definition, it is not too hard to show that, for example, \mathbb{N} and \mathbb{Z} have the same cardinality (they are both “countably infinite”).

It turns out, though, that $\mathcal{P}(\mathbb{N})$ has a strictly greater cardinality than \mathbb{N} . We can prove this by showing: *no sequence $f_0, f_1, f_2, f_3, \dots$ includes all characteristic functions for subsets of \mathbb{N} .* (This shows that there are more characteristic functions than natural numbers.)

In fact, for any infinite list $f_0, f_1, f_2, f_3, \dots$ of characteristic functions, we can define a characteristic function f which is *not* on the list. Imagine each function given as the infinite sequence of its values, so the list might look like this:

$$\begin{array}{ll} f_0 & \text{values } 0\underline{1}01010101\dots \\ f_1 & \text{values } 00\underline{0}0011101\dots \\ f_2 & \text{values } 11\underline{1}111111\dots \\ f_3 & \text{values } 000\underline{0}000000\dots \\ f_4 & \text{values } 1001\underline{0}01001\dots \\ & \vdots \end{array}$$

Now if we switch each of the underlined values to its opposite, we get a characteristic function

$$f(n) = \begin{cases} 1 & \text{if } f_n(n) = 0 \\ 0 & \text{if } f_n(n) = 1 \end{cases}$$

which is *different* from each function on the list. In fact, it has a different value from f_n on the number n .

For the given example, f has values

$$11011\dots$$

The construction of f is sometimes called a “diagonalisation argument”, because we get its values by switching values along the diagonal in the table of values of $f_0, f_1, f_2, f_3, \dots$

Questions

14.1 Suggest domains and codomains for the following functions, writing the domain as a cartesian product where applicable.

gcd, reciprocal, remainder \cap , \cup

14.2 If A and B are subsets of \mathbb{N} with characteristic functions χ_A and χ_B respectively, what set does the function $\chi_A(n)\chi_B(n)$ represent?

14.3 How many Boolean functions of n variables are there?

Lecture 15: Composition and inversion

Complicated functions are often built from simple parts. For example, the function $f : \mathbb{R} \rightarrow \mathbb{R}$ defined by $f(x) = (x^2 + 1)^3$ is computed by doing the following steps in succession:

- square,
- add 1,
- cube.

We say that $f(x) = (x^2 + 1)^3$ is the composite of the functions (from \mathbb{R} to \mathbb{R})

- $\text{square}(x) = x^2$,
- $\text{successor}(x) = x + 1$,
- $\text{cube}(x) = x^3$.

15.1 Notation for composite functions

In the present example we write

$$f(x) = \text{cube}(\text{successor}(\text{square}(x))),$$

or

$$f = \text{cube} \circ \text{successor} \circ \text{square}.$$

In general, if $f(x) = g(h(x))$ we write $f = g \circ h$ and say f is the *composite* of g and h .

Warning: Remember that $g \circ h$ means “do h first, then g .” $g \circ h$ is usually different from $h \circ g$.

Example.

$$\begin{aligned}\text{square}(\text{successor}(x)) &= (x + 1)^2 = x^2 + 2x + 1 \\ \text{successor}(\text{square}(x)) &= x^2 + 1\end{aligned}$$

15.2 Conditions for composition

Composite functions do not always exist.

Example. If $\text{reciprocal} : \mathbb{R} - \{0\} \rightarrow \mathbb{R}$ is defined by $\text{reciprocal}(x) = \frac{1}{x}$ and $\text{predecessor} :$

$\mathbb{R} \rightarrow \mathbb{R}$ is defined by $\text{predecessor}(x) = x - 1$, then $\text{reciprocal} \circ \text{predecessor}$ does not exist, because $\text{predecessor}(1) = 0$ is not a legal input for reciprocal .

To avoid this problem, we demand that the codomain of h be equal to the domain of g for $g \circ h$ to exist. This ensures that each output of h will be a legal input for g .

Let $h : A \rightarrow B$ and $g : C \rightarrow D$ be functions. Then $g \circ h : A \rightarrow D$ exists if and only if $B = C$.

15.3 The identity function

On each set A the function $i_A : A \rightarrow A$ defined by

$$i_A(x) = x,$$

is called the *identity function* (on A).

15.4 Inverse functions

Functions $f : A \rightarrow A$ and $g : A \rightarrow A$ are said to be inverses (of each other) if

$$f \circ g = g \circ f = i_A.$$

Example. square and sqrt are inverses of each other on the set $\mathbb{R}^{\geq 0}$ of reals ≥ 0 .

$$\text{sqrt}(\text{square}(x)) = x \text{ and } \text{square}(\text{sqrt}(x)) = x.$$

In fact, this is exactly what sqrt is supposed to do – reverse the process of squaring. However, this works only if we restrict the domain to $\mathbb{R}^{\geq 0}$. On \mathbb{R} we do not have $\text{sqrt}(\text{square}(x)) = x$ because, for example,

$$\text{sqrt}(\text{square}(-1)) = \text{sqrt}(1) = 1.$$

This problem arises whenever we seek an inverse for a function which is not one-to-one. The squaring function on \mathbb{R} sends both 1 and -1 to 1, but we want a single value 1 for $\text{sqrt}(1)$. Thus we have to restrict the squaring function to $\mathbb{R}^{\geq 0}$.

15.5 Conditions for inversion

A function f can have an inverse without its domain and codomain being equal.

The inverse of a function $f : A \rightarrow B$ is a function $f^{-1} : B \rightarrow A$ such that

$$f^{-1} \circ f = i_A \quad \text{and} \quad f \circ f^{-1} = i_B.$$

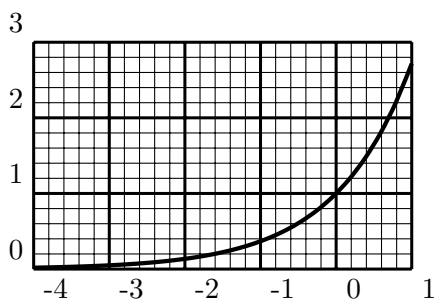
Note that $f^{-1} \circ f$ and $f \circ f^{-1}$ are both identity functions but they have different domains.

Not every function has an inverse, but we can neatly classify the ones that do.

Let $f : A \rightarrow B$ be a function. Then $f^{-1} : B \rightarrow A$ exists if and only if f is one-to-one and onto.

Example: e^x and \log

Consider $f : \mathbb{R} \rightarrow \mathbb{R}^{\geq 0} - \{0\}$ defined by $f(x) = e^x$. We know that e^x is one-to-one (e.g. because it is strictly increasing), and onto. So it has an inverse f^{-1} on $\mathbb{R}^{\geq 0} - \{0\}$.



Plot of $y = e^x$.

In fact, $f^{-1} = \log(y)$ where

$$\log : \mathbb{R}^{\geq 0} - \{0\} \rightarrow \mathbb{R}.$$

Now

$$e^{\log x} = x \quad \text{and} \quad \log(e^x) = x,$$

so $e^{\log x}$ and $\log(e^x)$ are both identity functions, but they have different domains.

The domain of $e^{\log x}$ is $\mathbb{R}^{\geq 0} - \{0\}$ (note \log is defined only for reals > 0). The domain of $\log(e^x)$ is \mathbb{R} .

15.6 Operations

An *operation* is a particular type of function, with domain $A \times A \times A \times \dots \times A$ and codomain A , for some set A .

For example, the addition function $f(a, b) = a + b$ is called an *operation on* \mathbb{R} , because $f :$

$\mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$. (That is, addition is a function of two real variables, which takes real values.)

An operation with one variable is called *unary*, an operation with two variables is called *binary*, an operation with three variables is called *ternary*, and so on.

Examples

1. Addition is a binary operation on \mathbb{R} .
2. Successor is a unary operation on \mathbb{N} .
3. Intersection is a binary operation on $\mathcal{P}(A)$ for any set A .
4. Complementation is a unary operation on $\mathcal{P}(A)$ for any set A .

Questions

- 15.1** Suppose f, m and s are the following functions on the set of people.

$m(x)$ = mother of x

$f(x)$ = father of x

$s(x)$ = spouse of x

What are the English terms for the following composite functions?

$$m \circ s, \quad f \circ s, \quad m \circ m, \quad f \circ m, \quad s \circ s$$

- 15.2** Write the following functions as composites of $\text{square}(x)$, $\text{sqrt}(x)$, $\text{successor}(x)$ and $\text{cube}(x)$.

$$\sqrt{1+x^3}, \quad x^{3/2}, \quad (1+x)^3, \quad (1+x^3)^2$$

- 15.3** What interesting feature do the following functions have in common? (Hint: consider their inverses.)

- \neg on \mathbb{B}
- The reciprocal, $f(x) = \frac{1}{x}$, on $\mathbb{R} - \{0\}$
- The function $g(x) = \frac{x}{x-1}$, on $\mathbb{R} - \{1\}$.

Lecture 16: Relations

Mathematical objects can be related in various ways, and any particular way of relating objects is called a *relation* on the set of objects in question.

(This also applies to relations in the everyday sense. For example, “parent of” is a relation on the set of people.)

A *binary relation* R on a set A consists of A and a set of ordered pairs from $A \times A$.
When (a, b) is in this set we write aRb .

Similarly, a *ternary* relation on A would be defined by a set of ordered triples from $A \times A \times A$, and so on. (A *unary* relation on A is just a subset of A .)

16.1 Relations and functions

Any function $f : X \rightarrow Y$ can be viewed as a relation R on $X \cup Y$. The relation is defined by

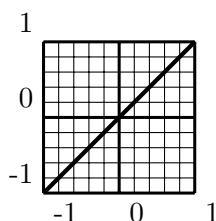
$$xRy \text{ if and only if } y = f(x).$$

However, not every relation is a function. Remember that a function must have exactly one output y for each input x in its domain. In a relation, on the other hand, an element x may be related to many elements y , or to none at all.

16.2 Examples

1. Equality on \mathbb{R} .

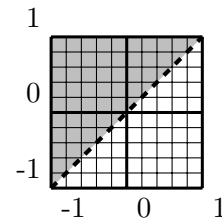
This is the relation consisting of the pairs (x, x) for $x \in \mathbb{R}$. Thus it is the following subset of the plane.



This relation is also a function (the identity function on \mathbb{R}), since there is exactly one pair for each $x \in \mathbb{R}$.

2. The $<$ relation on \mathbb{R} .

This relation consists of all the pairs (x, y) with $x < y$. It is the following shaded subset of the plane.

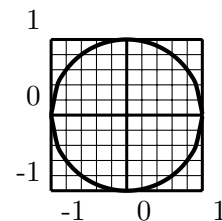


(The dashed line indicates that the points where $x = y$ are omitted.)

3. Algebraic curves.

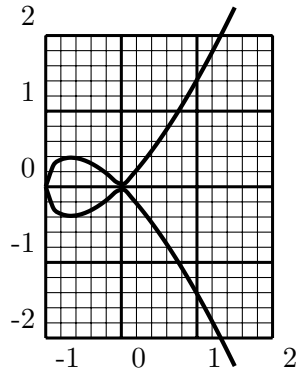
An algebraic curve consists of the points (x, y) satisfying an equation $p(x, y) = 0$, where p is a polynomial.

E.g. unit circle $x^2 + y^2 - 1 = 0$.



Notice that this relation is not a function, because there are two pairs with the same x , e.g. $(0, 1)$ and $(0, -1)$.

Likewise, the curve $y^2 = x^2(x + 1)$ is not a function.



4. The subset relation \subseteq .

This consists of the ordered pairs of sets (A, B) such that $A \subseteq B$. A and B must both be subsets of some universal set U .

5. Congruence modulo n .

For a fixed n , congruence modulo n is a binary relation. It consists of all the ordered pairs of integers (a, b) such that n divides $a - b$.

16.3 Properties of congruence

As the symbol \equiv suggests, congruence mod n is a lot like equality. Numbers a and b which are congruent mod n are not necessarily equal, but they are “equal up to multiples of n ,” because they have equal remainders when divided by n .

Because congruence is like equality, congruence $a \equiv b \pmod{n}$ behave a lot like equations. In particular, they have the following three properties.

1. Reflexive property.

$$a \equiv a \pmod{n}$$

for any number a .

2. Symmetric property.

$$a \equiv b \pmod{n} \Rightarrow b \equiv a \pmod{n}$$

for any numbers a and b .

3. Transitive property.

$$a \equiv b \pmod{n} \text{ and } b \equiv c \pmod{n} \Rightarrow$$

$$a \equiv c \pmod{n}$$

for any numbers a, b and c .

These properties are clear if one remembers that $a \equiv b \pmod{n}$ means a and b have the same remainder on division by n .

Questions

- 16.1 Which of the following relations $R(x, y)$ satisfy $\forall x \exists y R(x, y)$?

- $x \wedge y = \top$ (for propositions x, y)
- $x \subseteq y$ (for sets x, y of natural numbers)
- $x > y$ (for real numbers x, y)
- x divides y (for natural numbers x, y)

- 16.2 For each of the following relations between real numbers, find an equivalent equation.

- $x = 0 \vee y = 0$
- $x = 0 \wedge y = 0$

- 16.3 Use logic symbols and the \leq relation to write a relation between real numbers x, y which says that the point (x, y) lies in the square with corners $(0,0)$, $(1,0)$, $(0,1)$ and $(1,1)$.

Lecture 17: Equivalence relations

An *equivalence relation* R on a set A is a binary relation with the following three properties.

1. Reflexivity.

$$aRa \\ \text{for all } a \in A.$$

2. Symmetry.

$$aRb \Rightarrow bRa \\ \text{for all } a, b \in A.$$

3. Transitivity.

$$aRb \text{ and } bRc \Rightarrow aRc \\ \text{for all } a, b, c \in A.$$

Equality and congruence mod n (for fixed n) are examples of equivalence relations.

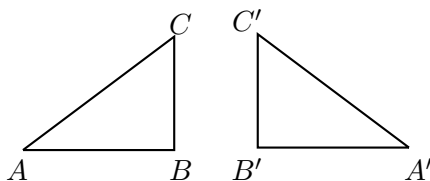
17.1 Other equivalence relations

1. Equivalence of fractions.

Two fractions are equivalent if they reduce to the same fraction when the numerator and denominator of each are divided by their gcd. E.g. $\frac{2}{4}$ and $\frac{3}{6}$ are equivalent because both reduce to $\frac{1}{2}$.

2. Congruence of triangles.

Triangles ABC and $A'B'C'$ are congruent if $AB = A'B'$, $BC = B'C'$ and $CA = C'A'$. E.g. the following triangles are congruent.

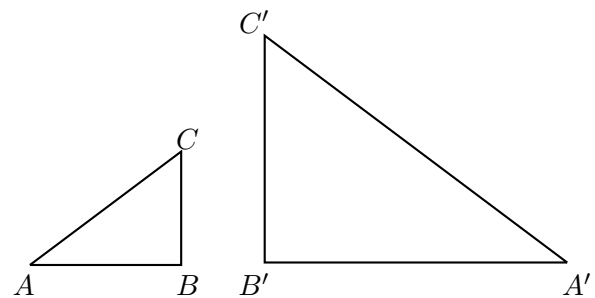


3. Similarity of triangles.

Triangles ABC and $A'B'C'$ are similar if

$$\frac{AB}{A'B'} = \frac{BC}{B'C'} = \frac{CA}{C'A'}.$$

E.g. the following triangles are similar



4. Parallelism of lines.

The relation $L \parallel M$ (L is parallel to M) is an equivalence relation.

Remark

In all these cases the relation is an equivalence because it says that objects are the *same* in some respect.

1. Equivalent fractions have the same reduced form.
2. Congruent triangles have the same side lengths.
3. Similar triangles have the same shape.
4. Parallel lines have the same direction.

Sameness is always reflexive (a is the same as a), symmetric (if a is the same as b , then b is the same as a) and transitive (if a is the same as b and b is the same as c , then a is the same as c).

17.2 Equivalence classes

Conversely, we can show that if R is a reflexive, symmetric and transitive relation then aRb says that a and b are the same in some respect: *they have the same R -equivalence class*.

If R is an equivalence relation we define the *R -equivalence class* of a to be

$$[a] = \{s : sRa\}.$$

Thus $[a]$ consists of all the elements related to a . It can also be defined as $\{s : aRs\}$, because sRa if and only if aRs , by symmetry of R .

Examples

- The parallel equivalence class of a line L consists of all lines parallel to L .
- The equivalence class of 1 for congruence mod 2 is the set of all odd numbers.

17.3 Equivalence class properties

Claim. *If two elements are related by an equivalence relation R on a set A , their equivalence classes are equal.*

Proof. Suppose $a, b \in A$ and aRb . Now

$$\begin{aligned} s \in [a] &\Rightarrow sRa \text{ by definition of } [a] \\ &\Rightarrow sRb \text{ by transitivity of } R \\ &\quad \text{since } sRa \text{ and } aRb \\ &\Rightarrow s \in [b] \text{ by definition of } [b]. \end{aligned}$$

Thus all elements of $[a]$ belong to $[b]$. Similarly, all elements of $[b]$ belong to $[a]$, hence $[a] = [b]$. \square

Claim. *If R is an equivalence relation on a set A , each element of A belongs to exactly one equivalence class.*

Proof. Suppose $a, b, c \in A$, and $c \in [a] \cap [b]$.

$$\begin{aligned} c \in [a] \text{ and } c \in [b] \\ \Rightarrow cRa \text{ and } cRb \\ \text{by definition of } [a] \text{ and } [b] \\ \Rightarrow aRc \text{ and } cRb \text{ by symmetry} \\ \Rightarrow aRb \text{ by transitivity} \\ \Rightarrow [a] = [b] \\ \text{by the previous claim.} \end{aligned}$$

17.4 Partitions and equivalence classes

A *partition* of a set S is a set of subsets of S such that each element of S is in exactly one of the subsets.

Using what we showed in the last section, we have the following.

If R is an equivalence relation on a set A , then the equivalence classes of R form a partition of A . Two elements of A are related if and only if they are in the same equivalence class.

Example. Let R be the relation on \mathbb{Z} defined by aRb if and only if $a \equiv b \pmod{3}$. The three equivalence classes of R are

$$\begin{aligned} \{x : x \equiv 0 \pmod{3}\} &= \{3k : k \in \mathbb{Z}\} \\ \{x : x \equiv 1 \pmod{3}\} &= \{3k + 1 : k \in \mathbb{Z}\} \\ \{x : x \equiv 2 \pmod{3}\} &= \{3k + 2 : k \in \mathbb{Z}\}. \end{aligned}$$

These partition the set \mathbb{Z} .

Questions

17.1 Which of the following relations between integers x and y are equivalence relations?

- $|x| = |y|$
- $x^3 - y^3 = 0$
- $x^3 - y^3 = 1$
- x divides y
- 5 divides $x - y$
- the empty relation ($x = y \wedge x \neq y$)

17.2 For those relations in Question 17.1 that are *not* equivalence relations, say which properties of equivalence they fail to satisfy.

17.3 For those that *are* equivalence relations, say what is the “same” about the related objects.

17.4 Also, for those relations that are equivalence relations, describe their equivalence classes.

Lecture 18: Order relations

A *partial order relation* R on a set A is a binary relation with the following three properties.

1. Reflexivity.

$$aRa \\ \text{for all } a \in A.$$

2. Antisymmetry.

$$aRb \text{ and } bRa \Rightarrow a = b \\ \text{for all } a, b \in A.$$

3. Transitivity.

$$aRb \text{ and } bRc \Rightarrow aRc \\ \text{for all } a, b, c \in A.$$

The relation \subseteq on sets is a typical example of a partial order relation.

A *total order relation* R on a set A is a partial order relation that also has the property aRb or bRa for all $a, b \in A$.

by comparing their values at the least natural number where they differ. The function with the lower value is put lower in the ordering.

4. Ordering functions on \mathbb{R} .

There is no known total ordering of functions on \mathbb{R} . They can be partially ordered in various ways. For example, we can say that $f \leq g$ if $f(x) \leq g(x)$ for all x .

With this ordering there are many pairs of functions, e.g. square and cube, neither of which is \leq the other.

5. Divisibility on \mathbb{N} .

The relation “ a divides b ” is reflexive, antisymmetric and transitive, hence a partial order. It is not a total order because, e.g., neither of the numbers 2,3 divides the other.

18.1 Examples

1. The \leq relation on \mathbb{R} .

This is obviously reflexive ($a \leq a$), antisymmetric ($a \leq b$ and $b \leq a \Rightarrow a = b$) and transitive ($a \leq b$ and $b \leq c \Rightarrow a \leq c$).

It also has the property that for any a and b we have $a \leq b$ or $b \leq a$. This makes it a total order relation.

2. Alphabetical order of words.

Words on the English alphabet are ordered by comparing the leftmost letter at which they differ. The word with the earlier letter at this position is earlier in alphabetical order. This is another total order.

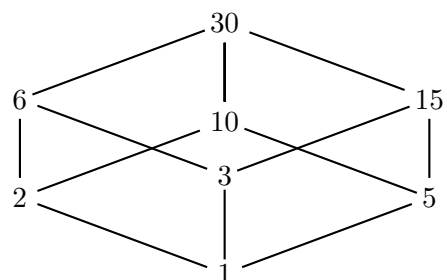
3. Alphabetical order of functions on \mathbb{N} . Functions on \mathbb{N} can also be totally ordered

18.2 Remarks

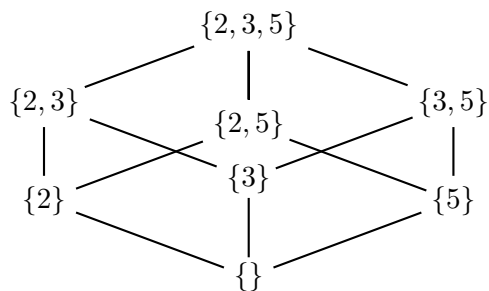
1. The set {divisors of 30}, ordered by the “divides” relation, has the same structure as the power set of {2, 3, 5}, ordered by \subseteq .

Here are the two sets, with lines connecting elements to those lower in the ordering.

Divisors of 30:



Subsets of $\{2, 3, 5\}$:



2. Such a correspondence between divisors and subsets of the prime factors occurs for any number with no repeated prime factor. It has a couple more interesting properties too:

- The gcd corresponds to \cap .
E.g. $\gcd(6, 10) = 2$, $\{2, 3\} \cap \{2, 5\} = \{2\}$.
- The *least common multiple* (lcm) corresponds to \cup .
E.g. $\text{lcm}(6, 10) = 30$, $\{2, 3\} \cup \{2, 5\} = \{2, 3, 5\}$.

3. In fact, every partial order relation has the structure of an ordering of sets by \subseteq . For each element a , consider the set $\{s : sRa\}$ of elements “less than or equal to a ”. Then

$$aRb \Leftrightarrow \{s : sRa\} \subseteq \{s : sRb\}.$$

4. The usual ordering of real numbers is almost an alphabetical ordering of their decimals. However, there are cases where two different decimals represent the same number, e.g.

$$0.4999 \dots = 0.50000 \dots$$

18.3 Well-ordering

A *well-order relation* R on a set A is a total order relation that also has the property that each nonempty subset of A has a least element.

The set $\mathbb{N} = \{0, 1, 2, 3, \dots\}$ is totally ordered by \leq and every nonempty subset of \mathbb{N} has a least element. So \leq on \mathbb{N} is a well-order relation.

The well-ordering of \mathbb{N} is the basis of proofs by induction. (We previously used well-ordering

when we said “there is no infinite descending sequence of positive integers.”)

It is also the basis of the “alphabetical” total ordering of functions on \mathbb{N} – to order two functions f and g we need to look at the least element of $\{n : f(n) \neq g(n)\}$.

The set \mathbb{R} of reals is also totally ordered by \leq , but not well-ordered. For example, the subset of reals > 0 has no least member.

Questions

18.1 Does “antisymmetric” mean “not symmetric”?

- If so, explain why.
- If not, give an example of a relation which is neither symmetric nor antisymmetric.

18.2 Draw a diagram of the positive divisors of 42 under the relation “divides.” Why does it resemble the diagram for the positive divisors of 30?

18.3 Define a total ordering of the pairs (m, n) of natural numbers. Is your ordering a well-ordering?

Lecture 19: Selections and Arrangements

19.1 Ordered selections without repetition

A reviewer is going to compare ten phones and list, in order, a top three. In how many ways can she do this? More generally, how many ways are there to arrange r objects chosen from a set of n objects?

In our example, the reviewer has 10 options for her favourite, but then only 9 for her second-favourite, and 8 for third-favourite. So there are $10 \times 9 \times 8$ ways she could make her list.

For an ordered selection without repetition of r elements from a set of n elements there are

n options for the 1st element
 $n - 1$ options for the 2nd element
 $n - 2$ options for the 3rd element
 \vdots
 $n - r + 1$ options for the r th element.

So we have the following formula.

The number of ordered selections without repetition of r elements from a set of n elements ($0 \leq r \leq n$) is

$$n(n-1) \cdots (n-r+1) = \frac{n!}{(n-r)!}.$$

When $r = n$ and all the elements of a set S are ordered, we just say that this is a *permutation* of S . Our formula tells us there are $n!$ such permutations. For example, there are $3! = 6$ permutations of the set $\{a, b, c\}$:

$(a, b, c), (a, c, b), (b, a, c), (b, c, a), (c, a, b), (c, b, a).$

19.2 Unordered selections without repetition

What if our reviewer instead chose an unordered top three? In how many ways could she do that? More generally, how many ways are there to choose (without order) r objects from a set of n objects?

A *combination* of r elements from a set S is a subset of S with r elements.

For every unordered list our reviewer could make there are $3! = 6$ corresponding possible ordered lists. And we've seen that she could make $10 \times 9 \times 8$ ordered lists. So the number of unordered lists she could make is $\frac{10 \times 9 \times 8}{6}$.

For every combination of r elements from a set of n elements there are $r!$ corresponding permutations. So, using our formula for the number of permutations we have the following.

The number of combinations of r elements from a set of n elements ($0 \leq r \leq n$) is

$$\frac{n(n-1) \cdots (n-r+1)}{r!} = \frac{n!}{r!(n-r)!} = \binom{n}{r}.$$

Notice that the notation $\binom{n}{r}$ is used for $\frac{n!}{r!(n-r)!}$. Expressions like this are called *binomial coefficients*. We'll see why they are called this in the next lecture.

19.3 Ordered selections with repetition

An ordered selection of r elements from a set X is really just a sequence of length r with each term in X . If X has n elements, then there are n possibilities for each term and so:

The number of sequences of r terms, each from some set of n elements, ($0 \leq r \leq n$) is

$$\underbrace{n \times n \times \cdots \times n}_r = n^r.$$

19.4 Unordered selections with repetition

A shop has a special deal on any four cans of soft drink. Cola, lemonade and sarsaparilla flavours are available. In how many ways can you select four cans?

We can write a selection in a table, for example,

C	L	S		and	C	L	S
•	••	•				•	•••

We can change a table like this into a string of 0s and 1s, by moving from left to right read-

ing a “•” as a 0 and a column separator as a 1. The tables above would be converted into

0 1 0 0 1 0 and 1 0 1 0 0 0

Notice that each string has four 0s (one for each can selected) and two 1s (one fewer than the number of flavours). We can choose a string like this by beginning with a string of six 0s and then choosing two 0s to change to 1s. There are $\binom{6}{2}$ ways to do this and so there are $\binom{6}{2}$ possible can selections.

An unordered selection of r elements, with repetition allowed, from a set X of n elements can be thought of as a multiset with r elements, each in X . As in the example, we can represent each such multiset with a string of r 0s and $n - 1$ 1s. We can choose a string like this by beginning with a string of $n + r - 1$ 0s and then choosing $n - 1$ 0s to change to 1s.

The number of multisets of r elements, each from a set of n elements, is

$$\binom{n+r-1}{n-1} = \frac{(n+r-1)!}{(n-1)!r!}.$$

19.5 The pigeonhole principle

The pigeonhole principle is a reasonably obvious statement, but can still be very useful.

If n items are placed in m containers with $n > m$, then at least one container has at least two items.

Example. If a drawer contains only blue, black and white socks and you take out four socks without looking at them, then you are guaranteed to have two of the same colour.

We can generalise the pigeonhole principle as follows.

If n items are placed in m containers, then at least one container has at least $\lceil \frac{n}{m} \rceil$ items.

In the above $\lceil \frac{n}{m} \rceil$ means the smallest integer greater than $\frac{n}{m}$ (or $\frac{n}{m}$ “rounded up”).

Example. If 21 tasks have been distributed between four processor cores, the busiest core must have been assigned at least 6 tasks.

Questions

- 19.1** A bank requires a PIN that is a string of four decimal digits. How many such PINs are there? How many are made of four different digits?
- 19.2** How many binary strings of length 5 are there? How many of these contain exactly two 1s?
- 19.3** In a game, each of ten players holds red, blue and green marbles, and places one marble in a bag. How many possibilities are there for the colours of marbles in the bag? If each player chooses their colour at random are all of these possibilities equally likely?
- 19.4** How many ways are there to partition (that is, split up) a set with 10 elements into one class of 5 elements, one class of 3 elements and one class of 2 elements? How many ways are there to partition a set with 10 elements into one class of 6 elements and two classes of 2 elements each?

Lecture 20: Pascal's triangle

20.1 Pascal's triangle

We can write the binomial coefficients in an (infinite) triangular array as follows:

$$\begin{array}{ccccccc}
 & & & & \binom{0}{0} & & \\
 & & & \binom{1}{0} & \binom{1}{1} & & \\
 & & \binom{2}{0} & \binom{2}{1} & \binom{2}{2} & & \\
 & \binom{3}{0} & \binom{3}{1} & \binom{3}{2} & \binom{3}{3} & & \\
 \binom{4}{0} & \binom{4}{1} & \binom{4}{2} & \binom{4}{3} & \binom{4}{4} & & \\
 \binom{5}{0} & \binom{5}{1} & \binom{5}{2} & \binom{5}{3} & \binom{5}{4} & \binom{5}{5} & \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
 \end{array}$$

Here are the first ten rows with the entries as integers:

$$\begin{array}{cccccccccc}
 & & & & & & & & 1 & & & & & & \\
 & & & & & & & 1 & & 1 & & & & & \\
 & & & & & 1 & & 2 & & 1 & & & & & \\
 & & & 1 & & 3 & & 3 & & 1 & & & & & \\
 & & 1 & & 4 & & 6 & & 4 & & 1 & & & & \\
 & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & & & \\
 1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 & & \\
 1 & & 7 & & 21 & & 35 & & 35 & & 21 & & 7 & & 1 \\
 1 & & 8 & & 28 & & 56 & & 70 & & 56 & & 28 & & 8 & & 1 \\
 1 & & 9 & & 36 & & 84 & & 126 & & 126 & & 84 & & 36 & & 9 & & 1 \\
 1 & & 10 & & 45 & & 120 & & 210 & & 252 & & 210 & & 120 & & 45 & & 10 & & 1
 \end{array}$$

This triangular array is often called *Pascal's triangle* (although Pascal was nowhere near the first to discover it).

20.2 Patterns

Writing the binomial coefficients this way reveals a lot of different patterns in them. Perhaps the most obvious is that every row reads the same left-to-right and right-to-left. Choosing r elements from a set of n elements to be in a combination is equivalent to choosing $n - r$ elements from the same set to not be in the combination. So:

$$\binom{n}{r} = \binom{n}{n-r} \text{ for } 0 \leq r \leq n.$$

This shows that every row reads the same left-to-right and right-to-left.

Another pattern is that every "internal" entry in the triangle is the sum of the two entries

above it. To see why this is, we'll begin with an example.

Example. Why is $\binom{6}{2} = \binom{5}{2} + \binom{5}{1}$?

There are $\binom{6}{2}$ combinations of 2 elements of $\{1, 2, 3, 4, 5, 6\}$. Every such combination either

- does not contain a 6, in which case it is one of the $\binom{5}{2}$ combinations of 2 elements of $\{1, 2, 3, 4, 5\}$; or
- does contain a 6, in which case the rest of the combination is one of the $\binom{5}{1}$ combinations of 1 element from $\{1, 2, 3, 4, 5\}$.

So $\binom{6}{2} = \binom{5}{2} + \binom{5}{1}$.

We can make a similar argument in general. Let X be a set of n elements and x is a fixed element of X . For any $r \in \{1, \dots, n\}$, there are $\binom{n-1}{r}$ combinations of r elements of X that do not contain x and there are $\binom{n-1}{r-1}$ combinations of r elements of X that do contain x . So:

$$\binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1} \text{ for } 1 \leq r \leq n.$$

This shows that every internal entry in Pascal's triangle is the sum of the two above it.

20.3 The binomial theorem

$$\begin{aligned}
 (x+y)^0 &= 1 \\
 (x+y)^1 &= x+y \\
 (x+y)^2 &= x^2 + 2xy + y^2 \\
 (x+y)^3 &= x^3 + 3x^2y + 3xy^2 + y^3 \\
 (x+y)^4 &= x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4 \\
 (x+y)^5 &= x^5 + 5x^4y + 10x^3y^2 + 10x^2y^3 + 5xy^4 + y^5
 \end{aligned}$$

Notice that the coefficients on the right are exactly the same as the entries in Pascal's triangle. Why does this happen? Think about expanding $(x+y)^3$ and finding the coefficient of xy^2 , for example.

$$\begin{aligned}
 (x+y)(x+y)(x+y) &= xxx + xxy + xyx + \underline{xyy} \\
 &\quad + yxx + \underline{yyx} + \underline{yyy} + yyy \\
 &= x^3 + 3x^2y + 3xy^2 + y^3
 \end{aligned}$$

The coefficient of xy^2 is 3 because we have three terms in the sum above that contain two y 's (those underlined). This is because there are $\binom{3}{2}$ ways to choose two of the three factors in a term to be y 's.

The same logic holds in general. The coefficient of $x^{n-r}y^r$ in $(x+y)^n$ will be $\binom{n}{r}$ because there will be $\binom{n}{r}$ ways to choose r of the n factors in a term to be y 's. This fact is called the *binomial theorem*.

Binomial theorem For any $n \in \mathbb{N}$,

$$(x+y)^n = \binom{n}{0}x^n y^0 + \binom{n}{1}x^{n-1}y^1 + \binom{n}{2}x^{n-2}y^2 + \cdots + \binom{n}{n-1}x^1 y^{n-1} + \binom{n}{n}x^0 y^n.$$

20.4 Inclusion-exclusion

A school gives out prizes to its best ten students in music and its best eight students in art. If three students receive prizes in both, how many students get a prize? If we try to calculate this as $10 + 8$ then we have counted the three over-achievers twice. To compensate we need to subtract three and calculate $10 + 8 - 3 = 15$.

In general, if A and B are finite sets then we have

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

With a bit more care we can see that if A , B and C are sets then we have

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

This is part of a more general law called the *inclusion-exclusion principle*.

Let X_1, X_2, \dots, X_t be finite sets. To calculate $|X_1 \cup X_2 \cup \cdots \cup X_t|$:

- add the sizes of the sets;
- subtract the sizes of the 2-way intersections;
- add the sizes of the 3-way intersections;
- subtract the sizes of the 4-way intersections;
- \vdots
- add/subtract the size of the t -way intersection.

To see why this works, think of an element x that is in n of the sets X_1, X_2, \dots, X_t . It is counted

$$\binom{n}{1} - \binom{n}{2} + \binom{n}{3} - \binom{n}{4} + \cdots \pm \binom{n}{n}$$

times. By the Binomial theorem with $x = 1$ and $y = -1$ (see Question 20.2), this is equal to 1. So each element is counted exactly once.

Questions

20.1 Substitute $x = 1$ and $y = -1$ into the statement of the binomial theorem. What

does this tell you about the rows of Pascal's triangle?

20.2 Find a pattern in the sums of the rows in Pascal's triangle. Prove your pattern holds using the binomial theorem. Also prove it holds by considering the powerset of a set.

20.3 Use inclusion-exclusion to work out how many numbers in the set $\{1, \dots, 100\}$ are divisible by 2 or 3 or 5.

Lecture 21: Probability

Probability gives us a way to model random processes mathematically. These processes could be anything from the rolling of dice, to radioactive decay of atoms, to the performance of a stock market index. The mathematical environment we work in when dealing with probabilities is called a probability space.

21.1 Probability spaces

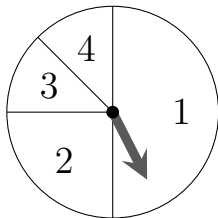
We'll start with a formal definition and then look at some examples of how the definition is used.

A probability space consists of

- a set S called a *sample space* which contains all the possible *outcomes* of the random process; and
- a *probability function* $\Pr : S \rightarrow [0, 1]$ such that the sum of the probabilities of the outcomes in S is 1.

Each time the process occurs it should produce exactly one outcome (never zero or more than one). The probability of an outcome is a measure of the likeliness that it will occur. It is given as a real number between 0 and 1 inclusive, where 0 indicates that the outcome cannot occur and 1 indicates that the outcome must occur.

Example.



The spinner above might be modeled by a probability space with sample space $S = \{1, 2, 3, 4\}$ and probability function given as follows.

$$\Pr(s) = \begin{cases} \frac{1}{2} & \text{for } s = 1 \\ \frac{1}{4} & \text{for } s = 2 \\ \frac{1}{8} & \text{for } s = 3 \\ \frac{1}{8} & \text{for } s = 4. \end{cases}$$

It can be convenient to give this as a table:

s	1	2	3	4
$\Pr(s)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$

Example. Rolling a fair six-sided die could be modeled by a probability space with sample space $S = \{1, 2, 3, 4, 5, 6\}$ and probability function \Pr given as follows.

s	1	2	3	4	5	6
$\Pr(s)$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

A sample space like this one where every outcome has an equal probability is sometimes called a *uniform sample space*. Outcomes from a uniform sample space are said to have been taken *uniformly at random*.

21.2 Events

An *event* is a subset of the sample space.

An event is just a collection of outcomes we are interested in for some reason.

Example. In the die rolling example with $S = \{1, 2, 3, 4, 5, 6\}$, we could define the event of rolling at least a 3. Formally, this would be the set $\{3, 4, 5, 6\}$. We could also define the event of rolling an odd number as the set $\{1, 3, 5\}$.

The probability of an event A is the sum of the probabilities of the outcomes in A .

Example. In the spinner example, for the event $A = \{1, 2, 4\}$, we have

$$\begin{aligned} \Pr(A) &= \Pr(1) + \Pr(2) + \Pr(4) \\ &= \frac{1}{2} + \frac{1}{4} + \frac{1}{8} \\ &= \frac{7}{8}. \end{aligned}$$

In a uniform sample space (where all outcomes are equally likely) the probability of an event A can be calculated as:

$$\Pr(A) = \frac{\text{number of outcomes in } A}{\text{number of outcomes}} = \frac{|A|}{|S|}.$$

21.3 Operations on events

Because events are defined as sets we can perform set operations on them. If A and B are

events for a sample space S , then

- $A \cup B$ is the event “ A or B ,”
- $A \cap B$ is the event “ A and B ,”
- \bar{A} is the event “not A .”

We always take the sample space as our universal set, so \bar{A} means $S - A$.

21.4 Probabilities of unions

We saw in the section on the inclusion-exclusion principal that $|A \cup B| = |A| + |B| - |A \cap B|$ for finite sets A and B . We have a similar law in probability.

For any two events A and B ,

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B).$$

Example. In our die rolling example, let $A = \{1, 2\}$ and $B = \{2, 3, 4\}$ be events. Then

$$\Pr(A \cup B) = \frac{|A|}{|S|} + \frac{|B|}{|S|} - \frac{|A \cap B|}{|S|} = \frac{2}{6} + \frac{3}{6} - \frac{1}{6} = \frac{2}{3}.$$

Two events A and B are *mutually exclusive* if $\Pr(A \cap B) = 0$, that is, if A and B cannot occur together. For mutually exclusive events, we have

$$\Pr(A \cup B) = \Pr(A) + \Pr(B).$$

21.5 Independent events

We say that two events are *independent* when the occurrence or non-occurrence of one event does not affect the likelihood of the other occurring.

Two events A and B are *independent* if

$$\Pr(A \cap B) = \Pr(A)\Pr(B).$$

Example. A binary string of length 3 is generated uniformly at random. The event A that the first bit is a 1 is independent of the event B that the second bit is a 1. But A is *not* independent of the event C that the string contains exactly two 1s.

Formally, the sample space is $S = \{000, 001, 010, 011, 100, 101, 110, 111\}$ and

$\Pr(s) = \frac{1}{8}$ for any $s \in S$. So,

$$\begin{aligned} A &= \{100, 101, 110, 111\} & \Pr(A) &= \frac{1}{2} \\ B &= \{010, 011, 110, 111\} & \Pr(B) &= \frac{1}{2} \\ C &= \{011, 101, 110\} & \Pr(C) &= \frac{3}{8} \\ A \cap B &= \{110, 111\} & \Pr(A \cap B) &= \frac{1}{4} \\ A \cap C &= \{101, 110\} & \Pr(A \cap C) &= \frac{1}{4} \end{aligned}$$

So $\Pr(A \cap B) = \Pr(A)\Pr(B)$ but $\Pr(A \cap C) \neq \Pr(A)\Pr(C)$.

21.6 Warning

Random processes can occur in both discrete and continuous settings, and probability theory can be applied in either setting. In this lecture, and in the next four lectures, we are discussing only the discrete case. Many of the definitions and results we state apply only in this case. Our definition of a probability space, for example, is actually the definition of a discrete probability space, and so on.

The discrete setting provides a good environment to learn most of the vital concepts and intuitions of probability theory. What you learn here is very useful in itself, and will act as a good base if you go on to study continuous probability.

Questions

An integer is chosen uniformly at random from the set $\{1, 2, \dots, 30\}$. Let A be the event that the integer is at most 20. Let B be the event that the integer is divisible by 6. Let C be the event that the integer’s last digit is a 5.

21.1 Write A, B and C as sets, and find their probabilities.

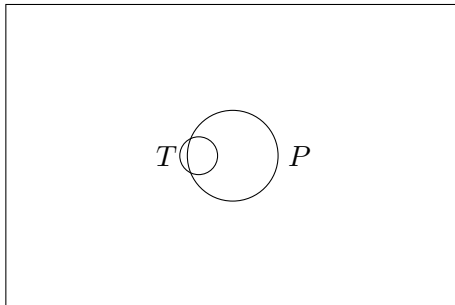
21.2 Find the probabilities of $A \cup B$, $A \cup C$ and $B \cup C$. Which pairs of A, B, C are mutually exclusive?

21.3 Find the probabilities of $A \cap B$, $A \cap C$ and $B \cap C$. Which pairs of A, B, C are independent?

Lecture 22: Conditional probability and Bayes' theorem

Your friend believes that Python coding has become more popular than AFL in Melbourne. She bets you \$10 that the next person to pass you on the street will be a Python programmer. You feel confident about this bet. However, when you see a man in a “Hello, world!” t-shirt approaching, you don’t feel so confident any more. Why is this?

We can think about this with a diagram. The rectangle represents the set of people in Melbourne, the circle P is the set of Python coders, and the circle T is the set of “Hello, world!” t-shirt owners.



Initially, you feel confident because the circle P takes up a small proportion of the rectangle. But when you learn that your randomly selected person is in the circle T , you feel bad because the circle P covers almost all of T . In mathematical language, the probability that a random Melbournian is a Python coder is low, but the probability that a random Melbournian is a Python coder given that they own a “Hello, world!” t-shirt is high.

22.1 Conditional probability

Conditional probabilities measure the likelihood of an event, given that some other event occurs.

For events A and B , the *conditional probability of A given B* is

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}.$$

This definition also implies that

$$\Pr(A \cap B) = \Pr(A|B)\Pr(B).$$

Example. The spinner from the last lecture is spun. Let A be the event that the result was at least 3 and B be the event that the result was

even. What is $\Pr(A|B)$?

$$\Pr(A \cap B) = \Pr(4) = \frac{1}{8}$$

$$\Pr(B) = \Pr(2) + \Pr(4) = \frac{1}{4} + \frac{1}{8} = \frac{3}{8}$$

Thus,

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \left(\frac{1}{8}\right) / \left(\frac{3}{8}\right) = \frac{1}{3}.$$

Example. A binary string of length 6 is generated uniformly at random. Let A be the event that the first bit is a 1 and B be the event that the string contains two 1s. What is $\Pr(A|B)$?

There are 2^6 strings in our sample space. Now $A \cap B$ occurs when the first bit is 1 and the rest of the string contains 1 one. There are $\binom{5}{1}$ such strings and so $\Pr(A \cap B) = \binom{5}{1}/2^6$. Also, there are $\binom{6}{2}$ strings containing two 1s and so $\Pr(B) = \binom{6}{2}/2^6$. Thus,

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} = \binom{5}{1} / \binom{6}{2} = \frac{1}{3}.$$

22.2 Independence again

Our definition of conditional probability gives us another way of defining independence. We can say that events A and B are independent if

$$\Pr(A) = \Pr(A|B).$$

This makes sense intuitively: it is a formal way of saying that the likelihood of A does not depend on whether or not B occurs.

22.3 Independent repeated trials

Generally if we perform exactly the same action multiple times, the results for each trial will be independent of the others. For example, if we roll a die twice, then the result of the first roll will be independent of the result of the second.

For two independent repeated trials, each from a sample space S , our overall sample space is $S \times S$ and our probability function will be given by $\Pr((s_1, s_2)) = \Pr(s_1)\Pr(s_2)$. For three independent repeated trials the sample space is $S \times S \times S$ and the probability function $\Pr((s_1, s_2, s_3)) = \Pr(s_1)\Pr(s_2)\Pr(s_3)$, and so on.

Example. The spinner from the previous example is spun twice. What is the probability that the results add to 5?

A total of 5 can be obtained as (1,4), (4,1), (2,3)

or (3,2). Because the spins are independent:

$$\begin{aligned}\Pr((1,4)) &= \Pr((4,1)) = \frac{1}{2} \times \frac{1}{8} = \frac{1}{16} \\ \Pr((2,3)) &= \Pr((3,2)) = \frac{1}{4} \times \frac{1}{8} = \frac{1}{32}\end{aligned}$$

So, because (1,4), (4,1), (2,3) and (3,2) are mutually exclusive, the probability of the total being 5 is $\frac{1}{16} + \frac{1}{16} + \frac{1}{32} + \frac{1}{32} = \frac{3}{16}$.

22.4 Bayes' theorem

Bayes' theorem gives a way of calculating the conditional probability of an event A given an event B when we already know the probabilities of A , of B given A , and of B given \bar{A} .

Bayes' theorem. For the events A and B ,

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B|A)\Pr(A) + \Pr(B|\bar{A})\Pr(\bar{A})}.$$

Note that the denominator above is simply an expression for $\Pr(B)$. The fact that

$$\Pr(B) = \Pr(B|A)\Pr(A) + \Pr(B|\bar{A})\Pr(\bar{A})$$

is due to the *law of total probability*.

22.5 Bayes' theorem examples

Example. The Australian government decides to crack down on vampires. They develop a test for vampirism that is right 99% of the time, and decide to screen everyone in the country. Suppose there are 100 vampires in the population of 24 million. We would guess that the test would come up positive for 99 of the 100 vampires, but also for 239 999 humans.

We are assuming that the conditional probability of testing positive given you're a vampire is 0.99. But what is the conditional probability of being a vampire given that you tested positive? From our guesses, we would expect this to be $\frac{99}{99+239999} \approx 0.0004$. Bayes' theorem gives us a way to formalise this:

$$\begin{aligned}\Pr(V|B) &= \frac{\Pr(P|V)\Pr(V)}{\Pr(P|V)\Pr(V) + \Pr(P|\bar{V})\Pr(\bar{V})} \\ &= \frac{\frac{100}{24000000} \times 0.99}{\frac{100}{24000000} \times 0.99 + (1 - \frac{100}{24000000}) \times 0.01} \\ &= \frac{99}{99+239999} \\ &\approx 0.0004.\end{aligned}$$

Example. A binary string is created so that the first bit is a 0 with probability $\frac{1}{3}$ and then each subsequent bit is the same as the preceding one

with probability $\frac{3}{4}$. What is the probability that the first bit is 0, given that the second bit is 0?

Let F be the event that the first bit is 0 and let S be the event that the second bit is 0. So $\Pr(F) = \frac{1}{3}$. If F occurs then the second bit will be 0 with probability $\frac{3}{4}$ and so $\Pr(S|F) = \frac{3}{4}$. If F does not occur then the second bit will be 0 with probability $\frac{1}{4}$ and so $\Pr(S|\bar{F}) = \frac{1}{4}$. So, by Bayes theorem,

$$\begin{aligned}\Pr(F|S) &= \frac{\Pr(F)\Pr(S|F)}{\Pr(F)\Pr(S|F) + \Pr(\bar{F})\Pr(S|\bar{F})} \\ &= \frac{\frac{1}{3} \times \frac{3}{4}}{\frac{1}{3} \times \frac{3}{4} + \frac{2}{3} \times \frac{1}{4}} \\ &= (\frac{1}{4}) / (\frac{5}{12}) \\ &= \frac{3}{5}.\end{aligned}$$

Questions

- 22.1** An integer is selected uniformly at random from the set $\{1, 2, \dots, 15\}$. What is the probability that it is divisible by 5, given that it is odd?
- 22.2** A standard die is rolled twice. What is the probability that the first roll is a 1, given that the sum of the rolls is 6?
- 22.3** A bag contains three black marbles and two white marbles and they are randomly selected and removed, one at a time until the bag is empty. Use Bayes' theorem to calculate the probability that the first marble selected is black, given that the second marble selected is black.

Lecture 23: Random variables

In a game, three standard dice will be rolled and the number of sixes will be recorded. We could let X stand for the number of sixes rolled. Then X is a special kind of variable whose value is based on a random process. These are called *random variables*.

Because the value of X is random, it doesn't make sense to ask whether $X = 0$, for example. But we can ask what the *probability is* that $X = 0$ or that $X \geq 2$. This is because " $X = 0$ " and " $X \geq 2$ " correspond to events from our sample space.

23.1 Formal definition

Formally, a random variable is defined as a function from the sample space to \mathbb{R} . In the example above, X is a function from the process's sample space that maps every outcome to the number of sixes in that outcome.

Example. Let X be the number of 1s in a binary string of length 2 chosen uniformly at random. Formally, X is a function from $\{00, 01, 10, 11\}$ to $\{0, 1, 2\}$ such that

$$X(00) = 0, \quad X(01) = 1, \quad X(10) = 1, \quad X(11) = 2.$$

For most purposes, however, we can think of X as simply a special kind of variable.

23.2 Probability distribution

We can completely describe a random variable X by listing, for each value x that X can take, the probability that $X = x$. This gives the *probability distribution* of the random variable. Again, formally this listing is a function from the values of X to their probabilities.

Example. Continuing with the last example, the probability distribution of X is given by

$$\Pr(X = x) = \begin{cases} \frac{1}{4} & \text{if } x = 0 \\ \frac{1}{2} & \text{if } x = 1 \\ \frac{1}{4} & \text{if } x = 2, \end{cases}$$

It can be convenient to give this as a table:

x	0	1	2
$\Pr(X = x)$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

Example. A standard die is rolled three times. Let X be the number of sixes rolled. What is the probability distribution of X ? Obviously X can only take values in $\{0, 1, 2, 3\}$. Each roll there is a six with probability $\frac{1}{6}$ and not a six with probability $\frac{5}{6}$. The rolls are independent.

$$\Pr(X = 0) = \frac{5}{6} \times \frac{5}{6} \times \frac{5}{6}$$

$$\Pr(X = 1) = \left(\frac{1}{6}\right)\left(\frac{5}{6}\right)\left(\frac{5}{6}\right) + \left(\frac{5}{6}\right)\left(\frac{1}{6}\right)\left(\frac{5}{6}\right) + \left(\frac{5}{6}\right)\left(\frac{5}{6}\right)\left(\frac{1}{6}\right)$$

$$\Pr(X = 2) = \left(\frac{1}{6}\right)\left(\frac{1}{6}\right)\left(\frac{5}{6}\right) + \left(\frac{1}{6}\right)\left(\frac{5}{6}\right)\left(\frac{1}{6}\right) + \left(\frac{5}{6}\right)\left(\frac{1}{6}\right)\left(\frac{1}{6}\right)$$

$$\Pr(X = 3) = \frac{1}{6} \times \frac{1}{6} \times \frac{1}{6}$$

So the probability distribution of X is

x	0	1	2	3
$\Pr(X = x)$	$\frac{125}{216}$	$\frac{75}{216}$	$\frac{15}{216}$	$\frac{1}{216}$

23.3 Independence

We have seen that two events are independent when the occurrence or non-occurrence of one event does not affect the likelihood of the other occurring. Similarly two random variables are *independent* if the value of one does not affect the likelihood that the other will take a certain value.

Random variables X and Y are *independent* if, for all x and y ,

$$\Pr(X = x \wedge Y = y) = \Pr(X = x)\Pr(Y = y).$$

Example. An integer is generated uniformly at random from the set $\{10, 11, \dots, 29\}$. Let X and Y be its first and second (decimal) digit. Then X and Y are independent random variables because, for $x \in \{1, 2\}$ and $\{0, 1, \dots, 9\}$,

$$\begin{aligned} \Pr(X = x \wedge Y = y) &= \frac{1}{20} \\ &= \frac{1}{2} \times \frac{1}{10} \\ &= \Pr(X = x)\Pr(Y = y). \end{aligned}$$

23.4 Operations

From a random variable X , we can create new random variables such as $X + 1$, $2X$ and X^2 . These variables work as you would expect them to.

Example. If X is the random variable with distribution

$$\frac{x}{\Pr(X=x)} \left\| \begin{array}{c|c|c} -1 & 0 & 1 \\ \hline \frac{1}{6} & \frac{1}{3} & \frac{1}{2} \end{array} \right\|,$$

then the distributions of $X+1$, $2X$ and X^2 are

$$\frac{y}{\Pr(X+1=y)} \left\| \begin{array}{c|c|c} 0 & 1 & 2 \\ \hline \frac{1}{6} & \frac{1}{3} & \frac{1}{2} \end{array} \right\|$$

$$\frac{y}{\Pr(2X=y)} \left\| \begin{array}{c|c|c} -2 & 0 & 2 \\ \hline \frac{1}{6} & \frac{1}{3} & \frac{1}{2} \end{array} \right\| \quad \frac{y}{\Pr(X^2=y)} \left\| \begin{array}{c|c} 0 & 1 \\ \hline \frac{1}{3} & \frac{2}{3} \end{array} \right\|.$$

23.5 Sums and products

From random variables X and Y we can define a new random variable $Z = X + Y$. Working out the distribution of Z can be complicated, however. We give an example below of doing this when X and Y are independent.

Example. Let X and Y be independent random variables with

$$\frac{x}{\Pr(X=x)} \left\| \begin{array}{c|c|c} 0 & 1 & 2 \\ \hline \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{array} \right\| \quad \frac{y}{\Pr(Y=y)} \left\| \begin{array}{c|c|c|c} 0 & 1 & 2 & 3 \\ \hline \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \right\|.$$

Let $Z = X + Y$. To find $\Pr(Z = z)$ for some value of z , we must consider all the ways that $X + Y$ could equal z . For example, $X + Y = 3$ could occur as $(X, Y) = (0, 3)$, $(X, Y) = (1, 2)$ or $(X, Y) = (2, 1)$. Because X and Y are independent, we can find the probability that each of these occur

$$\begin{aligned} \Pr(X = 0 \wedge Y = 3) &= \frac{1}{4} \times \frac{1}{6} = \frac{1}{24}, \\ \Pr(X = 1 \wedge Y = 2) &= \frac{1}{2} \times \frac{1}{3} = \frac{1}{6}, \\ \Pr(X = 2 \wedge Y = 1) &= \frac{1}{4} \times \frac{1}{3} = \frac{1}{12}. \end{aligned}$$

So, because the three are mutually exclusive,

$$\Pr(X = 3) = \frac{1}{24} + \frac{1}{6} + \frac{1}{12} = \frac{7}{24}.$$

Doing similar calculations for each possible value, we see that the distribution of Z is

$$\frac{z}{\Pr(Z=z)} \left\| \begin{array}{c|c|c|c|c|c} 0 & 1 & 2 & 3 & 4 & 5 \\ \hline \frac{1}{24} & \frac{1}{6} & \frac{7}{24} & \frac{7}{24} & \frac{1}{6} & \frac{1}{24} \end{array} \right\|.$$

The distribution of a product of two independent random variables can be found in a similar way.

Finding the distribution of sums or products of dependent random variables is even more complicated. In general, this requires knowing the probability of each possible combination of values the variables can take.

Questions

23.1 An elevator is malfunctioning. Every minute it is equally likely to ascend one floor, descend one floor, or stay where it is. When it begins malfunctioning it is on level 5. Let X be the level it is on three minutes later. Find the probability distribution for X .

23.2 An integer is generated uniformly at random from the set $\{11, 12, \dots, 30\}$. Let X and Y be its first and second (decimal) digit. Are the random variables X and Y independent?

23.3 Let X and Y be independent random variables with distributions

$$\frac{x}{\Pr(X=x)} \left\| \begin{array}{c|c} 0 & 2 \\ \hline \frac{1}{4} & \frac{3}{4} \end{array} \right\|$$

$$\frac{y}{\Pr(Y=y)} \left\| \begin{array}{c|c|c|c} 0 & 1 & 2 & 3 \\ \hline \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{array} \right\|.$$

Find the probability distribution of $Z = X + Y$.

Lecture 24: Expectation and variance

A standard die is rolled some number of times and the average of the rolls is calculated. If the die is rolled only once this average is just the value rolled and is equally likely to be 1, 2, 3, 4, 5 or 6. If the die is rolled ten times, then the average might be between 1 and 2 but this is pretty unlikely – it's much more likely to be between 3 and 4. If the die is rolled ten thousand times, then we can be almost certain that the average will be very close to 3.5. We will see that 3.5 is the expected value of a random variable representing the die roll.

24.1 Expected value

When we said “average” above, we really meant “mean”. Remember that the *mean* of a collection of numbers is the sum of the numbers divided by how many of them there are. So the mean of x_1, \dots, x_t is $\frac{x_1 + \dots + x_t}{t}$. The mean of 2, 2, 3 and 11 is $\frac{2+2+3+11}{4} = 4.5$, for example.

The expected value of a random variable is calculated as a weighted average of its possible values.

If X is a random variable with distribution

x	x_1	x_2	\dots	x_t
$\Pr(X = x)$	p_1	p_2	\dots	p_t

then the *expected value* of X is

$$E[X] = p_1x_1 + p_2x_2 + \dots + p_tx_t.$$

Example. If X is a random variable representing a die roll, then

$$E[X] = \frac{1}{6} \times 1 + \frac{1}{6} \times 2 + \dots + \frac{1}{6} \times 6 = 3.5.$$

Example. Someone estimates that each year the share price of Acme Corporation has a 10% chance of increasing by \$10, a 50% chance of increasing by \$4, and a 40% chance of falling by \$10. Assuming that this estimate is good, are Acme shares likely to increase in value over the long term?

We can represent the change in the Acme share price by a random variable X with distribution

x	-10	4	10
$\Pr(X = x)$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{1}{10}$

Then

$$E[X] = \frac{2}{5} \times -10 + \frac{1}{2} \times 4 + \frac{1}{10} \times 10 = -1$$

Because this value is negative, Acme shares will almost certainly decrease in value over the long term.

Notice that it was important that we weighted our average using the probabilities here. If we had just taken the average of -10, 4 and 10 we would have gotten the wrong answer by ignoring the fact that some values were more likely than others.

24.2 Law of large numbers

Our initial die-rolling example hinted that the average of a large number of independent trials will get very close to the expected value. This is mathematically guaranteed by a famous theorem called the *law of large numbers*.

Let X_1, X_2, \dots be independent random variables, all with the same distribution and expected value μ . Then

$$\lim_{n \rightarrow \infty} \frac{1}{n} (X_1 + \dots + X_n) = \mu.$$

24.3 Linearity of expectation

We saw in the last lecture that adding random variables can be difficult. Finding the expected value of a sum of random variables is easy if we know the expected values of the variables.

If X and Y are random variables, then

$$E[X + Y] = E[X] + E[Y].$$

This works even if X and Y are not independent.

Similarly, finding the expected value of a scalar multiple of a random variable is easy if we know the expected value of the variable.

If X is a random variable and $s \in \mathbb{R}$, then

$$E[sX] = sE[X].$$

Example. Two standard dice are rolled. What is the expected total?

Let X_1 and X_2 be random variables representing the first and second die rolls. From the earlier example $E[X_1] = E[X_2] = 3.5$ and so

$$E[X_1 + X_2] = E[X_1] + E[X_2] = 3.5 + 3.5 = 7.$$

Example. What is the expected number of ‘11’ substrings in a binary string of length 5 chosen uniformly at random?

For $i = 1, \dots, 4$, let X_i be a random variable that is equal to 1 if the i th and $(i + 1)$ th bits of the string are both 1. Then $X_1 + \dots + X_4$ is the number of ‘11’ substrings in the string. Because the bits are independent, $\Pr(X_i = 1) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$ and $E[X_i] = \frac{1}{4}$ for $i = 1, \dots, 4$. So,

$$E[X_1 + \dots + X_4] = E[X_1] + \dots + E[X_4] = \frac{4}{4} = 1.$$

Note that the variables X_1, \dots, X_4 in the above example were not independent, but we were still allowed to use linearity of expectation.

24.4 Variance

Think of the random variables X , Y and Z whose distributions are given below.

x	-1	99	y	-1	1
$\Pr(X = x)$	$\frac{99}{100}$	$\frac{1}{100}$	$\Pr(Y = y)$	$\frac{1}{2}$	$\frac{1}{2}$
z	-50	50			
$\Pr(Z = z)$	$\frac{1}{2}$	$\frac{1}{2}$			

These variables are very different. Perhaps X corresponds to buying a raffle ticket, Y to making a small bet on a coin flip, and Z to making a large bet on a coin flip. However, if you only consider expected value, all of these variables look the same – they each have expected value 0.

To give a bit more information about a random variable we can define its *variance*, which measures how “spread out” its distribution is.

If X is a random variables with $E[X] = \mu$,

$$\text{Var}[X] = E[(X - \mu)^2].$$

So the variance is a measure of how much we expect the variable to differ from its expected value.

Example. The variable X above will be 1 smaller than its expected value with probability $\frac{99}{100}$ and will be 99 larger than its expected value with probability $\frac{1}{100}$. So

$$\text{Var}[X] = \frac{99}{100} \times (-1)^2 + \frac{1}{100} \times 99^2 = 99.$$

Similarly,

$$\text{Var}[Y] = \frac{1}{2} \times (-1)^2 + \frac{1}{2} \times 1^2 = 1$$

$$\text{Var}[Z] = \frac{1}{2} \times (-50)^2 + \frac{1}{2} \times 50^2 = 2500.$$

Notice that the variance of X is much smaller than the variance of Z because X is very likely to be close to its expected value whereas Z will certainly be far from its expected value.

Example. Let X be a random variable with distribution given by

x	0	2	6
$\Pr(X = x)$	$\frac{1}{6}$	$\frac{1}{2}$	$\frac{1}{3}$

Then the expected value of X is

$$E[X] = \frac{1}{6} \times 0 + \frac{1}{2} \times 2 + \frac{1}{3} \times 6 = 3.$$

So, the variance of X is

$$\text{Var}[X] = \frac{1}{6} \times (0-3)^2 + \frac{1}{2} \times (2-3)^2 + \frac{1}{3} \times (6-3)^2 = 5.$$

Questions

- 24.1** Do you agree or disagree with the following statement? “The expected value of a random variable is the value it is most likely to take.”
- 24.2** Let X be the sum of 1000 spins of the spinner from Lecture 21, and let Y be 1000 times the result of a single spin. Find $E[X]$ and $E[Y]$. Which of X and Y do you think would have greater variance?
- 24.3** Let X be the number of heads occurring when three fair coins are flipped. Find $E[X]$ and $\text{Var}[X]$.

Lecture 25: Discrete distributions

In this lecture we'll introduce some of the most common and useful (discrete) probability distributions. These arise in various different real-world situations.

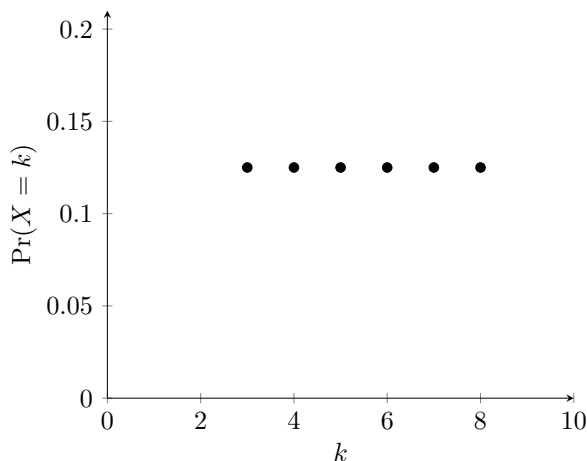
25.1 Discrete uniform distribution

This type of distribution arises when we choose one of a set of consecutive integers so that all choices are equally likely.

The *discrete uniform distribution* with parameters $a, b \in \mathbb{Z}$ ($a \leq b$) is given by $\Pr(X = k) = \frac{1}{b-a+1}$ for $k \in \{a, a+1, \dots, b\}$.

We have $E[X] = \frac{a+b}{2}$ and $\text{Var}[X] = \frac{(b-a+1)^2-1}{12}$.

Uniform distribution with $a = 3, b = 8$



25.2 Bernoulli distribution

This type of distribution arises when we have a single process that succeeds with probability p and fails otherwise. Such a process is called a *Bernoulli trial*.

The *Bernoulli distribution* with parameter $p \in [0, 1]$ is given by

$$\Pr(X = k) = \begin{cases} p & \text{for } k = 1 \\ 1 - p & \text{for } k = 0. \end{cases}$$

We have $E[X] = p$ and $\text{Var}[X] = p(1 - p)$.

25.3 Geometric distribution

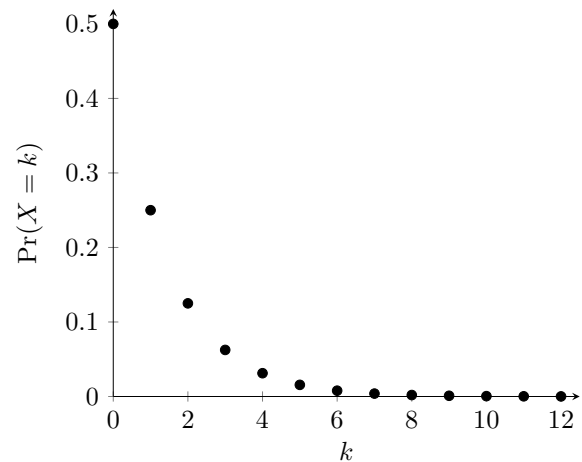
This distribution gives the probability that, in a sequence of independent Bernoulli trials, we see exactly k failures before the first success.

The *geometric distribution* with parameter $p \in [0, 1]$ is given by

$$\Pr(X = k) = p(1 - p)^k \text{ for } k \in \mathbb{N}.$$

We have $E[X] = \frac{1-p}{p}$ and $\text{Var}[X] = \frac{1-p}{p^2}$.

Geometric distribution with $p = 0.5$



Example. If every minute there is a 1% chance that your internet connection fails then the probability of staying online for exactly x consecutive minutes is approximated by a geometric distribution with $p = 0.01$. It follows that the expected value is $\frac{1-0.01}{0.01} = 99$ minutes and the variance is $\frac{1-0.01}{(0.01)^2} = 9900$.

25.4 Binomial distribution

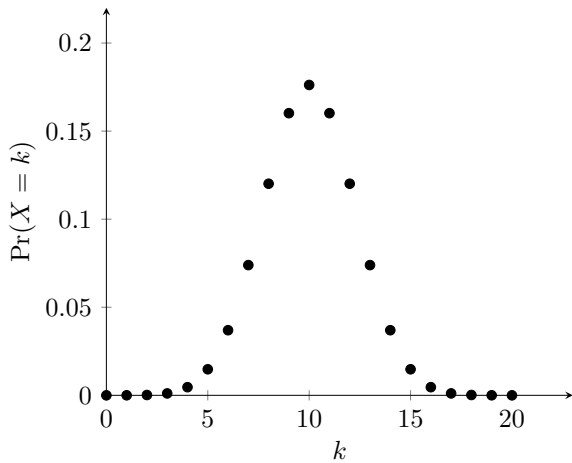
This distribution gives the probability that, in a sequence of n independent Bernoulli trials, we see exactly k successes.

The *binomial distribution* with parameters $n \in \mathbb{Z}^+$ and $p \in [0, 1]$ is given by

$$\Pr(X = k) = \binom{n}{k} p^k (1 - p)^{n-k} \text{ for } k \in \{0, \dots, n\}.$$

We have $E[X] = np$ and $\text{Var}[X] = np(1 - p)$.

Binomial distribution with $n = 20, p = 0.5$



Example. If 1000 people search a term on a certain day and each of them has a 10% chance of clicking a sponsored link, then the number of clicks on that link is approximated by a binomial distribution with $n = 1000$ and $p = 0.1$. It follows that the expected value is $1000 \times 0.1 = 100$ clicks and the variance is $1000 \times 0.1 \times 0.9 = 90$.

25.5 Poisson distribution

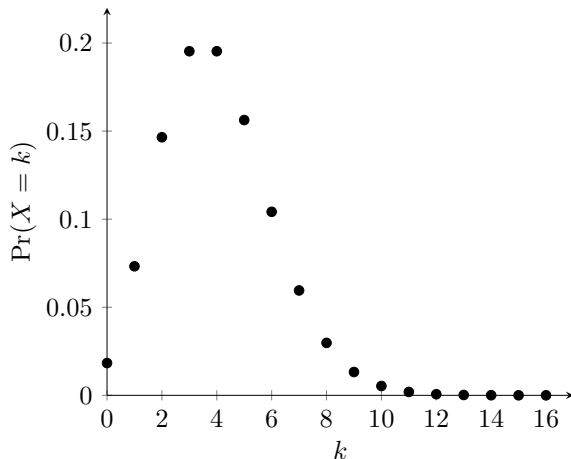
In many situations where we know that an average of λ events occur per time period, this distribution gives a good model of the probability that k events occur in a time period.

The *Poisson distribution* with parameter $\lambda \in \mathbb{R}$ (where $\lambda > 0$) is given by

$$\Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \text{ for } k \in \mathbb{N}.$$

We have $E[X] = \lambda$ and $\text{Var}[X] = \lambda$.

Poisson distribution with $\lambda = 4$



Example. If a call centre usually receives 6 calls per minute, then a Poisson distribution with

$\lambda = 6$ approximates the probability it receives k calls in a certain minute. It follows that the expected value is 6 calls and the variance is 6.

Questions

- 25.1** There is a 95% chance of a packet being received after being sent down a noisy line, and the packet is resent until it is received. What is the probability that the packet is received within the first three attempts?
- 25.2** A factory aims to have at most 2% of the components it makes be faulty. What is the probability of a quality control test of 20 random components finding that 2 or more are faulty, if the factory is exactly meeting its 2% target?
- 25.3** The number of times a machine needs adjusting during a day approximates a Poisson distribution, and on average the machine needs to be adjusted three times per day. What is the probability it does not need adjusting on a particular day?

Lecture 26: Recursion

Just as the structure of the natural numbers supports induction as a method of proof, it supports induction as a method of definition or of computation.

When used in this way, induction is usually called *recursion*, and one speaks of a *recursive definition* or a *recursive algorithm*.

26.1 Recursive Definitions

Many well known functions $f(n)$ are most easily defined in the “base step, induction step” format, because $f(n+1)$ depends in some simple way on $f(n)$.

The induction step in the definition is more commonly called the *recurrence relation* for f , and the base step the *initial value*.

Example. The factorial $f(n) = n!$

Initial value. $0! = 1$.

Recurrence relation. $(k+1)! = (k+1) \times k!$

Many programming languages allow this style of definition, and the value of the function is then computed by a descent to the initial value.

For example, to compute $4!$, the machine successively computes

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

which can finally be evaluated since $0! = 1$.

Remark. The numbers 4, 3, 2, 1 have to be stored on a “stack” before the program reaches the initial value $0! = 1$ which finally enables it to evaluate $4!$.

Thus a recursive program, though short, may run slowly and even cause “stack overflow.”

Example. The Fibonacci sequence

0, 1, 1, 2, 3, 5, 8, ...

The n^{th} number $F(n)$ in this sequence is defined by

Initial values. $F(0) = 0, F(1) = 1$.

Recurrence relation. $F(k+1) = F(k) + F(k-1)$.

Remark. Using a recursive program to compute Fibonacci numbers can easily lead to stack overflow, because each value depends on two previous values (each of which depends on another two, and so on).

A more efficient way to use the recursive definition is to use three variables to store $F(k+1)$, $F(k)$ and $F(k-1)$. The new values of these variables, as k increases by 1, depend only on the three stored values, not on all the previous values.

26.2 Properties of recursively defined functions

These are naturally proved by induction, using a base step and induction step which parallel those in the definition of the function.

Example. For $n \geq 5$, 10 divides $n!$

Proof *Base step.*

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 10 \times 4 \times 3,$$

hence 10 divides $5!$.

Induction step. We have to show

$$10 \text{ divides } k! \implies 10 \text{ divides } (k+1)!$$

Since $(k+1)! = (k+1) \times k!$ by the recurrence relation for factorial, the induction step is clear, and hence the induction is complete.

Example. $F(0) + F(1) + \cdots + F(n) = F(n+2) - 1$.

Proof *Base step.* $F(0) = 0 = F(2) - 1$, because $F(2) = 1$.

Induction step. We have to show

$$\begin{aligned} &F(0) + F(1) + \cdots + F(k) \\ &= F(k+2) - 1 \\ \implies &F(0) + F(1) + \cdots + F(k+1) \\ &= F(k+3) - 1. \end{aligned}$$

Well,

$$\begin{aligned} & F(0) + F(1) + \cdots + F(k) \\ &= F(k+2) - 1 \\ \Rightarrow & F(0) + F(1) + \cdots + F(k+1) \\ &= F(k+2) + F(k+1) - 1, \\ & \text{by adding } F(k+1) \text{ to both sides} \\ \Rightarrow & F(0) + F(1) + \cdots + F(k+1) \\ &= F(k+3) - 1 \\ & \text{since } F(k+2) + F(k+1) = F(k+3) \\ & \text{by the Fibonacci recurrence relation} \end{aligned}$$

This completes the induction.

26.3 Problems with recursive solutions

Sometimes a problem about n reduces to a problem about $n-1$ (or smaller numbers), in which case the solution may be a known recursively defined function.

Example. Find how many n -digit binary numerals contain no two consecutive 0s.

We can divide this problem into two cases.

1. Numerals which end in 1, e.g. 1101101.
In this case, the numeral before the 1 (110110 here) can be any $(n-1)$ digit numeral with no consecutive 0s.
2. Numerals which end in 0, e.g. 1011010.
In this case, the numeral must actually end in 10, to avoid consecutive 0s, but the numeral before the 10 (10110 here) can be any $(n-2)$ digit numeral with no consecutive 0s.

Thus among numerals with no consecutive 0s we find

1. Those with n digits ending in 1 correspond to those with $(n-1)$ digits.
2. Those with n digits ending in 0 correspond to those with $(n-2)$ digits.

Hence if we let $f(n)$ be the number of such numerals with n digits we have

$$f(n) = f(n-1) + f(n-2).$$

This is the Fibonacci recurrence relation.

It can also be checked that

$$f(1) = 2 = F(3) \text{ and } f(2) = 3 = F(4),$$

hence it follows (by induction) that

$$\begin{aligned} f(n) &= \text{number of } n \text{ digit numerals} \\ & \quad \text{with no consecutive 0s} \\ &= F(n+2). \end{aligned}$$

Questions

26.1 A function $s(n)$ is defined recursively by

$$\text{Initial value: } s(0) = 0$$

$$\text{Recurrence relation: } s(n+1) = s(n) + 2n + 1$$

Write down the first few values of $s(n)$, and guess what function s is.

26.2 Check that the function s you guessed in Question 26.1 satisfies

$$s(0) = 0 \text{ and } s(n+1) = s(n) + 2n + 1$$

(This proves by induction that your guess is correct.)

26.3 If a sequence satisfies the Fibonacci recurrence relation,

$$f(n) = f(n-1) + f(n-2),$$

must it agree with the Fibonacci sequence from some point onward?

Lecture 27: Recursive Algorithms

Recursion may be used to define functions whose definition normally involves "...", to give algorithms for computing these functions, and to prove some of their properties.

27.1 Sums

Example. $1 + 2 + 3 + \dots + n$

This is the function $f(n)$ defined by

Initial value. $f(1) = 1$

Recurrence relation. $f(k+1) = f(k) + (k+1)$

Example. $1 + a + a^2 + \dots + a^n$

This is the function $g(n)$ defined by

Initial value. $g(0) = 1$

Recurrence relation. $g(k+1) = g(k) + a^{k+1}$

We can use this relation to prove by induction that $g(n) = \frac{a^{n+1}-1}{a-1}$ (a formula for the sum of a geometric series), provided $a \neq 1$.

Proof

Base step. For $n = 0$, $1 = g(0) = \frac{a^{0+1}-1}{a-1}$, as required.

Induction step. We want to prove

$$g(k) = \frac{a^{k+1}-1}{a-1} \Rightarrow g(k+1) = \frac{a^{k+2}-1}{a-1}.$$

Well,

$$\begin{aligned} g(k) &= \frac{a^{k+1}-1}{a-1} \\ \Rightarrow g(k+1) &= \frac{a^{k+1}-1}{a-1} + a^{k+1} \\ \Rightarrow g(k+1) &= \frac{a^{k+1}-1 + (a-1)a^{k+1}}{a-1} \\ &= \frac{a^{k+2} + a^{k+1} - a^{k+1} - 1}{a-1} \\ &= \frac{a^{k+2}-1}{a-1} \text{ as required.} \end{aligned}$$

This completes the induction.

27.2 Products

Example. $1 \times 2 \times 3 \times \dots \times n$

This is the function $n!$ defined recursively by

Initial value. $0! = 1$

Recurrence relation. $(k+1)! = (k+1) \times k!$

27.3 Sum and product Notation

$1 + 2 + 3 + \dots + n$ is written $\sum_{k=1}^n k$,

$1 + a + a^2 + \dots + a^n$ is written $\sum_{k=0}^n a^k$.

Σ is capital sigma, standing for "sum."

$1 \times 2 \times 3 \times \dots \times n$ is written $\prod_{k=1}^n k$.

Π is capital pi, standing for "product."

27.4 Binary search algorithm

Given a list of n numbers in order

$$x_1 < x_2 < \dots < x_n,$$

we can find whether a given number a is on the list by repeatedly "halving" the list.

The algorithm **binary search** is specified recursively by a *base step* and a *recursive step*.

Base step. If the list is empty, report 'a is not in the list.'

Recursive step If the list is not empty, see whether its middle element is a . If so, report 'a found.'

Otherwise, if the middle element $m > a$, **binary search** the list of elements $< m$. And if the middle element $m < a$, **binary search** the list of elements $> m$.

27.5 Correctness

We prove that the algorithm works on a list of n items by strong induction on n .

Base step. The algorithm works correctly on a list of 0 numbers, by reporting that a is not on the list.

Induction step. Assuming the algorithm works correctly on any list of $< k + 1$ numbers, suppose we have a list of $k + 1$ numbers.

The recursive step either finds a as the middle number on the list, or else produces a list of $< k + 1$ numbers to search, which by assumption it will do correctly.

This completes the induction.

Remark. This example shows how easy it is to prove correctness of recursive algorithms, which may be why they are popular despite the practical difficulties in implementing them.

27.6 Running time

$\log_2 n$ is the number x such that

$$n = 2^x.$$

For example, $1024 = 2^{10}$, and therefore

$$\log_2 1024 = 10.$$

Similarly $\log_2 512 = 9$, and hence $\log_2 1000$ is between 9 and 10.

Repeatedly dividing 1000 by 2 (and discarding remainders of 1) runs for 9 steps:

$$500, 250, 125, 62, 31, 15, 7, 3, 1$$

The 10 halving steps for 1024 are

$$512, 256, 128, 64, 32, 16, 8, 4, 2, 1$$

This means that the binary search algorithm would do at most 9 “halvings” in searching a list of 1000 numbers and at most 10 “halvings” for 1024 numbers.

More generally, binary search needs at most $\lceil \log_2 n \rceil$ “halvings” to search a list of n numbers, where $\lceil \log_2 n \rceil$ is the *ceiling* of $\log_2 n$, the least integer $\geq \log_2 n$.

Remark. In a telephone book with 1,000,000 names, which is just under 2^{20} , it takes at most 20 halvings (using alphabetical order) to find whether a given name is present.

27.7 20 questions

A mathematically ideal way to play 20 questions would be to divide the number of possibilities in

half with each question.

E.g. if the answer is an integer, do binary search on the list of possible answers. If the answer is a word, do binary search on the list of possible answers (ordered alphabetically). If this is done, then 20 questions suffice to find the correct answer out of $2^{20} = 1,048,576$ possibilities.

Questions

27.1 Rewrite the following sums using \sum notation.

- $1 + 4 + 9 + 16 + \cdots + n^2$
- $1 - 2 + 3 - 4 + \cdots - 2n$

27.2 Which of the proofs in this lecture uses strong induction?

27.3 Imagine a game where the object is to identify a natural number between 1 and 2^{20} using 20 questions with YES-NO answers. The lecture explains why 20 questions are sufficient to identify any such number.

Explain why *less* than 20 YES-NO questions are *not* always sufficient.

Lecture 28: Recursion, lists and sequences

A *list* or *sequence* of objects from a set X is a function f from $\{1, 2, \dots, n\}$ to X , or (if infinite) from $\{1, 2, 3, \dots\}$ to X .

We usually write $f(k)$ as x_k and the list as $\langle x_1, x_2, \dots, x_n \rangle$, or $\langle x_1, x_2, x_3, \dots \rangle$. Thus

$$\begin{aligned} f(1) &= x_1 = \text{first item on list} \\ f(2) &= x_2 = \text{second item on list} \\ &\vdots \end{aligned}$$

The empty list is written $\langle \rangle$.

Example.

$\langle m, a, t, h, s \rangle$ is a function f from $\{1, 2, 3, 4, 5\}$ into the English alphabet, with $f(1) = m$, $f(2) = a$, etc.

28.1 Sequences

A sequence is also a list, but when we use the term sequence we are usually interested in the rule by which the successive terms t_1, t_2, \dots are defined.

Often, the rule is a recurrence relation.

Example. Arithmetic sequence

$$a, a + d, a + 2d, a + 3d, \dots$$

This is defined by

Initial value. $t_1 = a$

Recurrence relation. $t_{k+1} = t_k + d$

“Unfolding” this recurrence relation from t_n back to t_1 , we see that d gets added $n - 1$ times, hence

$$t_n = a + (n - 1)d.$$

Example. Geometric sequence

$$a, ar, ar^2, ar^3, \dots$$

Initial value. $t_1 = a$

Recurrence relation. $t_{k+1} = rt_k$

“Unfolding” t_n , we see that multiplication by r is done $n - 1$ times, hence

$$t_n = ar^{n-1}.$$

The above recurrence relations are called *first order* because t_{k+1} depends on only the previous value, t_k . (Or, because the values of all terms follow from one initial value.)

A *second order* recurrence relation requires *two* initial values, and is usually harder to unfold.

Example. A simple sequence in disguise

Initial values. $t_0 = 1, t_1 = 2$

Recurrence relation. $t_{k+1} = 2t_k - t_{k-1}$

Calculating the first values, we find

$$\begin{aligned} t_2 &= 2t_1 - t_0 = 2 \times 2 - 1 = 3, \\ t_3 &= 2t_2 - t_1 = 2 \times 3 - 2 = 4, \\ t_4 &= 2t_3 - t_2 = 2 \times 4 - 3 = 5. \end{aligned}$$

It looks like $t_n = n + 1$, and indeed we can prove this by induction. We already have the base step, by the initial value $t_0 = 1 = 0 + 1$. We do the induction step by strong induction: assuming $t_n = n + 1$ for all $n \leq k$, we deduce that $t_{k+1} = k + 2$.

In fact we have

$$\begin{aligned} t_{k+1} &= 2t_k - t_{k-1} \\ &\quad \text{by the recurrence relation} \\ &= 2(k + 1) - k \\ &\quad \text{by our assumption} \\ &= 2k + 2 - k = k + 2 \\ &\quad \text{as required.} \end{aligned}$$

This completes the induction.

Example. Fibonacci sequence

Initial values. $t_0 = 1, t_1 = 1$

Recurrence relation. $t_{k+1} = t_k + t_{k-1}$

It is possible to write t_n directly as a function of n . The function is not at all obvious, because it involves $\sqrt{5}$:

$$t_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1 + \sqrt{5}}{2} \right)^n - \left(\frac{1 - \sqrt{5}}{2} \right)^n \right).$$

We do not go into how to find such a formula in this unit. However, if someone gave you such a formula you could prove it is correct by induction.

28.2 Relations – homogeneous and inhomogeneous

Recurrence relations such as

$$t_{k+1} = 2t_k$$

and

$$t_{k+1} = t_k + t_{k-1}$$

in which each term is a multiple of some t_j , are called *homogeneous*.

The characteristic property of any homogeneous equation is that if $t_n = f(n)$ is a solution, then so is $t_n = cf(n)$, for any constant c .

E.g. $t_n = 2^n$ is a solution of $t_{k+1} = 2t_k$, and so is $t_n = c2^n$, for any constant c .

Relations like $t_{k+1} = t_k + 3$, in which there is a term other than the t_j terms, are called *inhomogeneous*.

Homogeneous recurrence relations are usually easier to solve, and in fact there is a general method for solving them (which we will not cover in this unit).

There is no general method for solving inhomogeneous recurrence relations, though they can often be solved if the term other than the t_j terms is simple.

Questions

28.1 Find the next four values of each of the following recurrence relations. What order is each recurrence relation? Which are homogeneous and which are inhomogeneous?

(a) $r_{k+1} = r_k + k^2$, $r_0 = 0$.

(b) $s_{k+1} = 3s_k - 2s_{k-1}$, $s_0 = 1$, $s_1 = 2$.

(c) $t_{k+1} = t_k + t_{k-2} + 1$, $t_0 = 1$, $t_1 = 1$, $t_2 = 1$.

28.2 Let T_n be the number of ways of tiling a $2 \times n$ strip with 2×1 tiles (which may be rotated so they are 1×2). Find T_n for $n = 1, 2, 3, 4$. Find a recurrence relation for T_n .

28.3 Let C_n be the number of ways of writing down n pairs of brackets so that they are correctly matched. Find C_n for $n = 1, 2, 3, 4$. Can you explain why $C_{k+1} = \sum_{i=0}^k C_i C_{k-i}$?

Lecture 29: Graphs

A *graph* consists of a set of objects called *vertices* and a list of pairs of vertices, called *edges*.

Graphs are normally represented by pictures, with vertex A represented by a dot labelled A and each edge AB represented by a curve joining A and B .

Such pictures are helpful for displaying data or relationships, and they make it easy to recognise properties which might otherwise not be noticed.

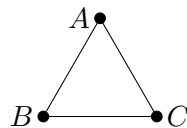
The description by lists of vertices and edges is useful when graphs have to be manipulated by computer. It is also a useful starting point for precise definitions of graph concepts.

29.1 Examples of graphs

Description

Vertices: A, B, C
Edges: AB, BC, CA

Picture

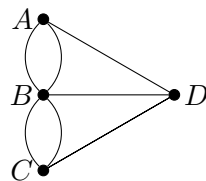


Such a graph, with at most one edge between each pair of vertices, and no vertex joined to itself, is called a *simple graph*.

Description

Vertices: A, B, C, D
Edges: $AB, AB, BC, BC, AD, BD, CD$

Picture



The two edges AB which join the same pair of vertices are called *parallel edges*.

Description

Vertices: A, B
Edges: AA, AB, AB

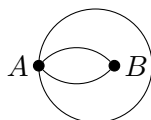
Picture



The edge joining A to A is called a *loop*.

The name *multigraph* is used when loops and/or parallel edges are allowed.

Warning: A graph can be represented by pictures that look very different. This last example could be redrawn as:



29.2 Problems given by graphs

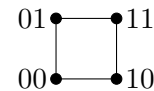
Many problems require vertices to be connected by a “path” of successive edges. We shall define paths (and related concepts) next lecture, but the following examples illustrate the idea and show how often it comes up.

They also show how helpful it is to have graph pictures when searching for paths.

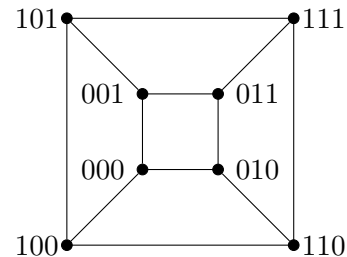
1. Gray Codes

The binary strings of length n are taken as the vertices, with an edge joining any two vertices that differ in only one digit. This graph is called the n -cube.

E.g. the 2-digit binary strings form a square (a “2-cube”).

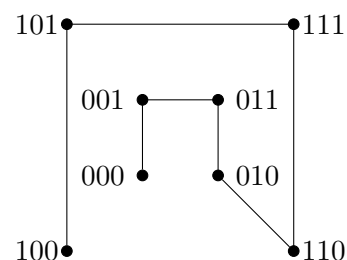


and the 3-digit binary strings form an ordinary cube (a “3-cube”).



A *Gray code* of length n is a path which includes each vertex of the n -cube exactly once. E.g. here is a path in the 3-cube which gives the Gray code

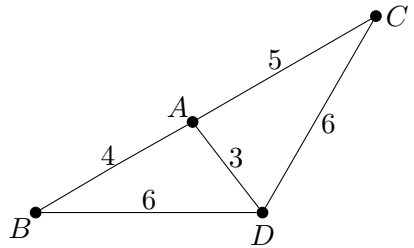
000, 001, 011, 010, 110, 111, 101, 100



Remark. The n -cube has been popular as a *computer architecture* in recent years. Processors are placed at the vertices of an n -cube (for $n = 15$ or so) and connected along its edges.

2. Travelling salesman problem

Vertices are towns. Two towns are joined by an edge labelled ℓ if there is a road of length ℓ between them. E.g.



The problem is to find a path of minimal length which includes all towns, in this case $BADC$.

3. Jug problem

Suppose we have three jugs, which hold exactly 3, 4 and 6 litres respectively. We fill the 6-litre jug, and then pour from one jug to another, always stopping when the jug being poured to becomes full or when the jug being poured from becomes empty.

Is it possible to reach a state where one jug contains 1 litre and another contains 5 litres?

We represent each possible state by a triple (a, b, c) , where

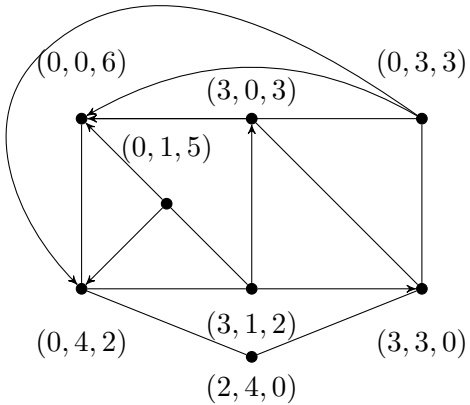
- a = number of litres in the 3-litre jug
- b = number of litres in the 4-litre jug
- c = number of litres in the 6-litre jug

Each state is a vertex, and if state (a', b', c') can be reached from (a, b, c) by pouring as described above, we put a directed edge in the graph:

$$(a, b, c) \bullet \longrightarrow \bullet (a', b', c')$$

If (a, b, c) can also be reached from (a', b', c') , we join them by an ordinary edge.

Then, listing the states that can be reached from $(0, 0, 6)$, we find the following graph.



A graph like this, with some directed edges, is called a *directed graph* or *digraph*.

This particular digraph shows that

$$(0, 0, 6) \rightarrow (0, 4, 2) \rightarrow (3, 1, 2) \rightarrow (0, 1, 5)$$

hence we can start with a full 6-litre jug and in three pourings get 1 litre in the 4-litre jug and 5 litres in the 6-litre jug.

Questions

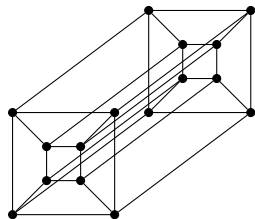
29.1 Write down descriptions of the following graphs

picture			
description			

29.2 Draw pictures of graphs with the following descriptions.

description	A, B, C AA, BB, CC	A, B, C, D AB, AC, AD
picture		

29.3 Use the following picture of the 4-cube to find a Gray code of length 4.



Lecture 30: Walks, paths and trails

There are several ways of “travelling” around the edges of a graph.

A *walk* is a sequence

$$V_1, e_1, V_2, e_2, V_3, e_3, \dots, e_{n-1}, V_n,$$

where each e_i is an edge joining vertex V_i to vertex V_{i+1} . (In a simple graph, where at most one edge joins V_i and V_{i+1} , it is sufficient to list the vertices alone.)

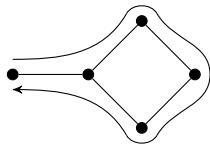
If $V_n = V_1$ the walk is said to be *closed*.

A *path* is a walk with no repeated vertices.

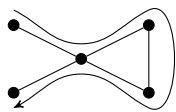
A *trail* is a walk with no repeated edges.

30.1 Examples

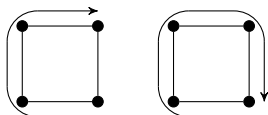
In these pictures, a walk is indicated by a directed curve running alongside the actual edges in the walk.



A walk which is not a trail or a path. (Repeated edge, repeated vertex.)



A trail which is not a path. (Repeated vertex.)



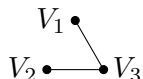
A nonclosed walk and a closed walk.

30.2 Adjacency matrix

If two vertices are joined by an edge we say that they are *adjacent*.

A simple graph G with vertices V_1, V_2, \dots, V_n is described by an *adjacency matrix* which has (i, j) entry (i^{th} row and j^{th} column) a_{ij} given by

$$a_{ij} = \begin{cases} 1 & \text{if } V_i \text{ is adjacent to } V_j \text{ in } G, \\ 0 & \text{otherwise.} \end{cases}$$

For example, the graph  has adjacency matrix $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$

30.3 Adjacency matrix powers

The *product* of matrices

$$\begin{pmatrix} a_{11} & a_{12} & \cdots \\ a_{21} & a_{22} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} \times \begin{pmatrix} b_{11} & b_{12} & \cdots \\ b_{21} & b_{22} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix}$$

is the matrix whose (i, j) entry is

$$a_{i1}b_{1j} + a_{i2}b_{2j} + a_{i3}b_{3j} + \cdots,$$

–the “dot product” of the i th row

$$a_{i1} \quad a_{i2} \quad a_{i3} \quad \cdots$$

of the matrix on the left with the j th column

$$b_{1j}$$

$$b_{2j}$$

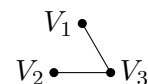
$$b_{3j}$$

$$\vdots$$

of the matrix on the right.

The (i, j) entry in the k^{th} power of the adjacency matrix gives the number of walks of length k between V_i and V_j .

For example, suppose we want the number of walks of length 2 from V_3 to V_3 in the graph



The adjacency matrix M tells us that the following edges exist.

$$\begin{array}{ccc}
\begin{pmatrix} \cdots & \cdots & 1 \\ \cdots & \cdots & 1 \\ 1 & 1 & 0 \end{pmatrix} & \leftarrow V_1 \text{ to } V_3 \\
& \leftarrow V_2 \text{ to } V_3 \\
\begin{array}{cc} \uparrow & \uparrow \\ V_3 & V_3 \\ \text{to} & \text{to} \\ V_1 & V_2 \end{array}
\end{array}$$

So when we square this matrix, the (3,3) entry in M^2

$$\begin{pmatrix} 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = 1 \times 1 + 1 \times 1 = 2$$

counts the walks from V_3 to V_3 , namely

$$V_3 \rightarrow V_1 \rightarrow V_3 \text{ and } V_3 \rightarrow V_2 \rightarrow V_3.$$

Similarly, the (i,j) entry in M^2 is the number of walks of length 2 from V_i to V_j . The (i,j) entry in M^3 is the number of walks of length 3 from V_i to V_j , and so on.

In fact,

$$M^2 \times M = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix} \times \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

has (3,2) entry

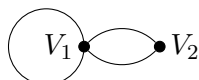
$$\begin{pmatrix} 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 2.$$

Hence the number of walks of length 3 from V_3 to V_2 is 2.

30.4 General adjacency matrix

The adjacency matrix can be generalised to multigraphs by making the (i,j) entry the *number* of edges from V_i to V_j (special case: count each loop twice).

For example, the graph



has adjacency matrix

$$N = \begin{pmatrix} 2 & 2 \\ 2 & 0 \end{pmatrix}$$

and so

$$N^2 = \begin{pmatrix} 8 & 2 \\ 2 & 4 \end{pmatrix}.$$

The (1,1) entry $2 \times 2 + 2 \times 2$ in N^2 , for example, indicates that there are 8 walks of length 2 from

V_1 to V_1 : 4 walks twice around the loop, and 4 walks from V_1 to V_2 (2 ways) then V_2 to V_1 (2 ways).

This count distinguishes between different directions around the loop. It may help to regard the loop as a pair of opposite *directed* loops.

We can generalise the adjacency matrix M to directed graphs by letting the (i,j) entry be the number of *directed* edges (which include ordinary edges) from V_i to V_j .

With this definition, the (i,j) entry of M^k gives the number of directed walks from V_i to V_j (i.e. walks that obey the directions of edges).

Questions

30.1 Draw the graph with adjacency matrix

$$M = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix},$$

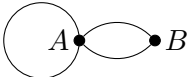
using V_1 , V_2 and V_3 as names for the vertices corresponding to columns 1, 2 and 3 respectively.

30.2 Calculate M^2 , and use it to find the number of walks of length 2 from V_1 to V_3 . Does this calculation give the number you would expect from the graph?

30.3 Without any calculation, show that the middle row of *any* power M^k is (0 1 0).

Lecture 31: Degree

The *degree* of a vertex A in a graph G is the number of times A occurs in the list of edges of G .

For example, if G is  then the list of edges is AA, AB, AB , and hence $\text{degree}(A) = 4$.

Intuitively speaking, $\text{degree}(A)$ is the number of “ends” of edges occurring at A . In particular, a loop at A contributes 2 to the degree of A .

31.1 The handshaking lemma

In any graph,
sum of degrees = $2 \times$ number of edges.

The reason for the name is that if each edge is viewed as a handshake,



then at each vertex V

$$\text{degree}(V) = \text{number of hands.}$$

Hence

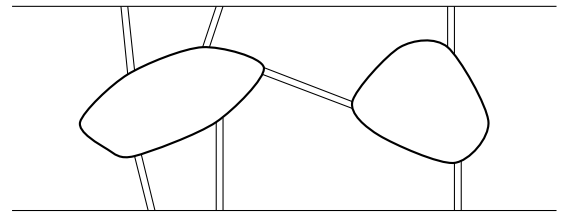
$$\begin{aligned} & \text{sum of degrees} \\ &= \text{total number of hands} \\ &= 2 \times \text{number of handshakes} \end{aligned}$$

An important consequence

The handshaking lemma implies that *in any graph the sum of degrees is even* (being $2 \times$ something). Thus it is impossible, e.g. for a graph to have degrees 1,2,3,4,5.

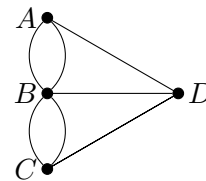
31.2 The seven bridges of Königsberg

In 18th century Königsberg there were seven bridges connecting islands in the river to the banks as follows.



The question came up: is it possible for a walk to cross all seven bridges without crossing the same bridge twice?

An equivalent question is whether there is a trail which includes all edges in the following graph.



31.3 Euler's solution

Euler (1737) observed that the answer is no, because

1. Each time a walk enters and leaves a vertex it “uses up” 2 from the degree.
2. Hence if all edges are used by the walk, all vertices except the first and last must have even degree.
3. The seven bridges graph in fact has four vertices of odd degree.

31.4 Euler's theorem

The same argument shows in general that

A graph with > 2 odd degree vertices has no trail using all its edges.

And a similar argument shows

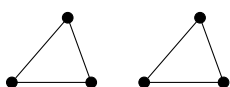
A graph with odd degree vertices has no *closed* trail using all its edges.

(Because in this case the first and last vertex are the same, and its degree is “used up” by a closed trail as follows: 1 at the start, 2 each time through, 1 at the end.)

31.5 The converse theorem

If, conversely, we have a graph G whose vertices all have even degree, is there a closed trail using all the edges of G ?

Not necessarily. For example, G might be *disconnected*:



We say a graph is *connected* if any two of its vertices are connected by a walk (or equivalently, by a trail or a path). We call a trail using all edges of a graph an *Euler trail*. Then we have

A connected graph with no odd degree vertices has a closed Euler trail.

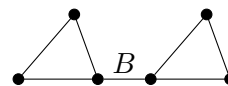
Such a closed trail can be constructed as follows:

1. Starting at any vertex V_1 , follow a trail t_1 as long as possible.
2. The trail t_1 eventually returns to V_1 , because it can leave any other vertex it enters. (Immediately after the start, V_1 has one “used” edge, and hence an odd number of “unused” edges. Any other vertex has an even number of “unused” edges.)
3. If t_1 does not use all edges, retrace it to the first vertex V_2 where t_1 meets an edge not in t_1 .
4. At V_2 add a “detour” to t_1 by following a trail out of V_2 as long as possible, not using edges in t_1 . As before, this trail eventually returns to its starting point V_2 , where we resume the trail t_1 . Let t_2 be the trail t_1 plus the detour from V_2 .
5. If t_2 does not use all the edges, retrace t_2 to the first vertex V_3 where t_2 meets an edge not in t_2 . Add a detour at V_3 , and so on.
6. Since a graph has only a finite number of edges, this process eventually halts. The result will be a closed trail which uses all the edges (this requires the graph to be

connected, since any unused edge would be connected to used ones, and thus would have eventually been used).

31.6 Bridges

A *bridge* in a connected graph G is an edge whose removal disconnects G . E.g. the edge B is a bridge in the following graph.



The construction of an Euler trail is improved by the doing the following (*Fleury’s algorithm*).

- Erase each edge as soon as it is used.
- Use a bridge in the remaining graph only if there is no alternative.

It turns out, when this algorithm is used, that it is not necessary to make any detours. The improvement, however, comes at the cost of needing an algorithm to recognise bridges.

Questions

31.1 For each of the following sequences, construct a graph whose vertices have those degrees, or explain why no such graph exists.

- 1, 2, 3, 4
- 1, 2, 1, 2, 1
- 1, 2, 2, 2, 1

31.2 A graph G has adjacency matrix

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Decide, without drawing the graph, whether G is connected or not.

31.3 A graph H has adjacency matrix

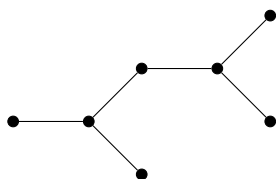
$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

What are the degrees of its vertices? Does H have a closed Euler trail?

Lecture 32: Trees

A *cycle* is a closed trail (with at least one edge) that doesn't repeat any vertex except that it ends where it started. A *tree* is a connected graph with no cycles.

For example,



is a tree.

32.1 The number of edges in a tree

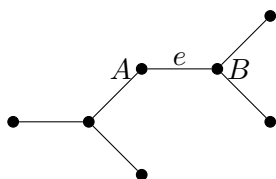
A tree with n vertices has $n - 1$ edges.

The proof is by strong induction on n .

Base step. A tree with 1 vertex has 0 edges (since any loop would itself be a cycle).

Induction step. Supposing any tree with $j \leq k$ vertices has $j - 1$ edges, we have to deduce that a tree with $k + 1$ vertices has k edges.

Well, given a tree T_{k+1} with $k + 1$ vertices, we consider any edge e in T_{k+1} , e.g.



Removing e disconnects the ends A and B of e . (If they were still connected, by some path p , then p and e together would form a cycle in T_{k+1} , contrary to its being a tree.)

Thus $T_{k+1} - \{e\}$ consists of two trees, say T_i and T_j with i and j vertices respectively. We have $i + j = k + 1$ but both $i, j \leq k$, so our induction assumption gives

T_i has $i - 1$ edges, T_j has $j - 1$ edges.

But then $T_{k+1} = T_i \cup T_j \cup \{e\}$ has $(i - 1) + (j - 1) + 1 = (i + j) - 1 = k$ edges, as required.

Remarks

1. This proof also shows that any edge in a tree is a bridge.
2. Since a tree has one more vertex than edge, it follows that m trees have m more vertices than edges.
3. The theorem also shows that adding any edge to a tree (without adding a vertex) creates a cycle. (Since the graph remains connected, but has too many edges to be a tree.)

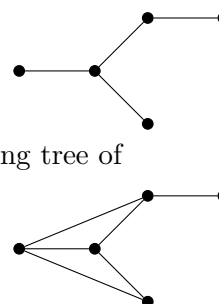
These remarks can be used to come up with several equivalent definitions of tree.

Next we see how any connected graph can be related to trees.

32.2 Spanning trees

A *spanning tree* of a graph G is a tree contained in G which includes all vertices of G .

For example,



is a spanning tree of

Any connected graph G contains a spanning tree.

This is proved by induction on the number of edges.

Base step. If G has no edge but is connected then it consists of a single vertex. Hence G itself is a spanning tree of G .

Induction step. Suppose any connected graph with $\leq k$ edges has a spanning tree, and

we have to find a spanning tree of a connected graph G_{k+1} with $k + 1$ edges.

If G_{k+1} has no cycle then G_{k+1} is itself a tree, hence a spanning tree of itself.

If G_{k+1} has a cycle p we can remove any edge e from p and $G_{k+1} - \{e\}$ is connected (because vertices previously connected via e are still connected via the rest of p). Since $G_{k+1} - \{e\}$ has one edge less, it contains a spanning tree T by induction, and T is also a spanning tree of G_{k+1} .

Remark It follows from these two theorems that a graph G with n vertices and $n - 2$ edges (or less) is *not* connected.

If it were, G would have a spanning tree T , with the same n vertices. But then T would have $n - 1$ edges, which is impossible, since it is more than the number of edges of G .

32.3 The greedy algorithm

Given a connected graph with weighted edges, a minimal weight spanning tree T of G may be constructed as follows.

1. Start with T empty.
2. While G has vertices not in T , add to T an edge e_{k+1} of minimal weight among those which do not close a cycle in T , together with the vertices of e_{k+1} .

This is also known as *Prim's algorithm*.

Remarks

1. T is not necessarily a tree at all steps of the algorithm, but it is at the end.
2. For a graph with n vertices, the algorithm runs for $n - 1$ steps, because this is the number of edges in a tree with n vertices.
3. The algorithm is called “greedy” because it always takes the cheapest step available, without considering how this affects future steps. For example, an edge of weight 4 may be chosen even though this prevents an edge of length 5 being chosen at the next step.
4. The algorithm always works, though this is *not* obvious, and the proof is not required for this course. (You can find it,

e.g. in Chartrand's *Introductory Graph Theory*.)

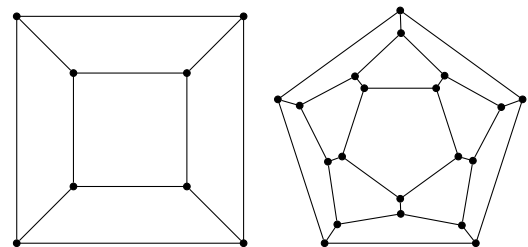
5. Another problem which can be solved by a “greedy” algorithm is splitting a natural number n into powers of 2. Begin by subtracting the largest such power $2^m \leq n$ from n , then repeat the process with $n - 2^m$, etc.

Questions

32.1 Which of the following graphs are trees? In each case we insist that $m \neq n$.

- vertices 1, 2, 3, 5, 7
an edge between m and n
if m divides n or n divides m
- vertices 1, 2, 3, 4, 5
an edge between m and n
if m divides n or n divides m
- vertices 2, 3, 4, 5, 6
an edge between m and n
if m divides n or n divides m

32.2 Find spanning trees of the following graphs (cube and dodecahedron).



32.3 Also find spanning trees of the cube and dodecahedron which are paths.

Lecture 33: Trees, queues and stacks

To search a graph G systematically, it helps to have a spanning tree T , together with an ordering of the vertices of T .

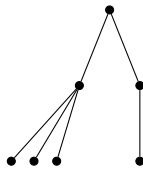
33.1 Breadth first ordering

The easiest ordering to understand is called *breadth first*, because it orders vertices “across” the tree in “levels.”

Level 0 is a given “root” vertex.

Level 1 is the vertices one edge away from the root.

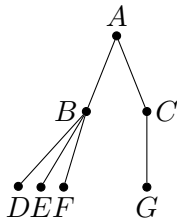
Level 2 are the vertices two edges away from the root,
... and so on.



Example.

A, B, C, D, E, F, G

is a breadth first ordering of



33.2 Queues

Breadth first ordering amounts to putting vertices in a *queue* - a list processed on a “first come, first served” or “first in, first out” basis.

- The root vertex is first in the queue (hence first out).
- Vertices adjacent to the head vertex v in the queue go to the tail of the queue (hence they come out after v), if they are not already in it.
- The head vertex v does not come out of the queue until all vertices adjacent to v have gone in.

33.3 Breadth first algorithm

For any connected graph G , this algorithm not only orders the vertices of G in a queue Q , it also builds a spanning tree T of G by attaching

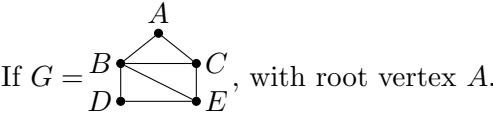
each vertex v to a “predecessor” among the adjacent vertices of v already in T . An arbitrary vertex is chosen as the root V_0 of T .

1. Initially, T = tree with just one vertex V_0 ,
 Q = the queue containing only V_0 .
2. While Q is nonempty
 - 2.1. Let V be the vertex at the head of Q
 - 2.2. If there is an edge $e = VW$ in G where W is not in T
 - 2.2.1. Add e and W to T
 - 2.2.2. Insert W in Q (at the tail).
 - 2.3. Else remove V from Q .

Remarks

1. If the graph G is not connected, the algorithm gives a spanning tree of the *connected component* containing the root vertex A , the part of G containing all vertices connected to A .
2. Thus we can recognise whether G is connected by seeing whether all its vertices are included when the algorithm terminates.
3. Being able to recognise connectedness enables us, e.g., to recognise bridges.

Example.



Then Q and T grow as follows:

Step	Q	T
1	A	$\bullet A$
2	AB	
3	ABC	
4	BC	
5	BCD	
6	$BCDE$	
7	CDE	
8	DE	
9	E	

33.4 Depth first algorithm

This is the same except it has a *stack* S instead of a queue Q . S is “last in, first out,” so we insert and remove vertices from the same end of S (called the *top* of the stack).

- Initially, T = tree with just one vertex V_0 ,
 S = the stack containing only V_0 .
- While S is nonempty
 - Let V be the vertex at the top of S
 - If there is an edge $e = VW$ in G where W is not in T
 - Add e and W to T
 - Insert W in S (at the top).
 - Else remove V from S .

Remark. The breadth first and depth first algorithms give two ways to construct a spanning tree of a connected graph.

Example.

We use the same G , and take the top of S to be its right hand end.

Step	S	T
1	A	$\bullet A$
2	AB	
3	ABC	
4	$ABCE$	
4	$ABCED$	
6	$ABCE$	
7	ABC	
8	AB	
9	A	

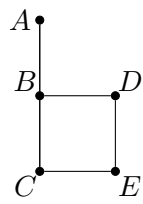
Questions

33.1 The following list gives the state, at successive stages, of either a queue or a stack.

- A
- AB
- ABC
- BC
- BCD
- CD
- D

Which is it: a queue or a stack?

33.2 Construct a breadth first spanning tree for the graph



33.3 Construct a depth first spanning tree for the graph in Question 33.2.

Useful notation

Sets of numbers

\mathbb{N}	the set of natural numbers	$\{0, 1, 2, 3, \dots\}$
\mathbb{Z}	the set of integers	$\{\dots, -2, -1, 0, 1, 2, \dots\}$
\mathbb{Q}	the set of rational numbers	$\{\frac{a}{b} : a, b \in \mathbb{Z}, b \neq 0\}$
\mathbb{R}	the set of real numbers	

Number Theory

$a \mid b$	a divides b	$b = qa$ for some $q \in \mathbb{Z}$
$\gcd(a, b)$	greatest common divisor of a and b	
$a \equiv b \pmod{n}$	a and b are congruent modulo n	$n \mid (a - b)$

Logic

$\neg p$	not p
$p \wedge q$	p and q
$p \vee q$	p or q
$p \rightarrow q$	p implies q
$\forall x$	for all x
$\exists x$	there exists x

Sets

$x \in A$	x is an element of A	
$\{x : P(x)\}$	the set of x such that $P(x)$	
$ A $	the number of elements in A	
$A \subseteq B$	A is a subset of B	
$A \cap B$	A intersect B	$\{x : x \in A \wedge x \in B\}$
$A \cup B$	A union B	$\{x : x \in A \vee x \in B\}$
$A - B$	set difference A minus B	$\{x : x \in A \wedge x \notin B\}$

Functions

$f : A \rightarrow B$	f is a function from A to B
-----------------------	-----------------------------------

Probability

$\Pr(E)$	probability of E
$\Pr(A B)$	conditional probability of A given B
$E[X]$	expected value of X
$\text{Var}[X]$	variance of X

Sums and products

$\sum_{i=a}^b f(i)$	sum of $f(i)$ from $i = a$ to $i = b$	$f(a) + f(a+1) + \dots + f(b)$
$\prod_{i=a}^b f(i)$	product of $f(i)$ from $i = a$ to $i = b$	$f(a) \times f(a+1) \times \dots \times f(b)$