Notes on

Set Theory for Computer Science

by Prof Glynn Winskel

A brief history of sets

A set is an unordered collection of objects, and as such a set is determined by the objects it contains. Before the 19th century it was uncommon to think of sets as completed objects in their own right. Mathematicians were familiar with properties such as being a natural number, or being irrational, but it was rare to think of say the collection of rational numbers as itself an object. (There were exceptions. From Euclid mathematicians were used to thinking of geometric objects such as lines and planes and spheres which we might today identify with their sets of points.)

In the mid 19th century there was a renaissance in Logic. For thousands of years, since the time of Aristotle and before, learned individuals had been familiar with syllogisms as patterns of legitimate reasoning, for example:

> All men are mortal. Socrates is a man. Therefore Socrates is mortal.

But syllogisms involved descriptions of properties. The idea of pioneers such as Boole was to assign a meaning as a set to these descriptions. For example, the two descriptions "is a man" and "is a male homo sapiens" both describe the same set, *viz.* the set of all men. It was this objectification of meaning, understanding properties as sets, that led to a rebirth of Logic and Mathematics in the 19th century. Cantor took the idea of set to a revolutionary level, unveiling its true power. By inventing a notion of size of set he was able compare different forms of infinity and, almost incidentally, to shortcut several traditional mathematical arguments.

But the power of sets came at a price; it came with dangerous paradoxes. The work of Boole and others suggested a programme exposited by Frege, and Russell and Whitehead, to build a foundation for all of Mathematics on Logic. Though to be more accurate, they were really reinventing Logic in the process, and regarding it as intimately bound up with a theory of sets. The paradoxes of set theory were a real threat to the security of the foundations. But with a lot of worry and care the paradoxes were sidestepped, first by Russell and Whitehead's theory of stratified types and then more elegantly, in for example the influential work of Zermelo and Fraenkel. The notion of set is now a cornerstone of Mathematics.

The precise notion of proof present in the work of Russell and Whitehead laid the scene for Gödel's astounding result of 1931: any sound proof system able to deal with arithmetic will necessarily be incomplete, in the sense that it will be impossible to prove *all* the statements within the system which are true. Gödel's theorem relied on the mechanical nature of proof in order to be able to encode proofs back into the proof system itself. After a flurry of activity, through the work of Gödel himself, Church, Turing and others, it was realised by the mid 1930's that Gödel's incompleteness result rested on a fundamental notion, that of computability. Arguably this marks the birth of Computer Science.

Motivation

Why learn Set Theory? Set Theory is an important language and tool for reasoning. It's a basis for Mathematics—pretty much all Mathematics can be formalised in Set Theory. Why is Set Theory important for Computer Science? It's a useful tool for formalising and reasoning about computation and the objects of computation. Set Theory is indivisible from Logic where Computer Science has its roots. It has been and is likely to continue to be a a source of fundamental ideas in Computer Science from theory to practice; Computer Science, being a science of the artificial, has had many of its constructs and ideas inspired by Set Theory. The strong tradition, universality and neutrality of Set Theory make it firm common ground on which to provide unification between seemingly disparate areas and notations of Computer Science. Set Theory is likely to be around long after most present-day programming languages have faded from memory. A knowledge of Set Theory should facilitate your ability to think abstractly. It will provide you with a foundation on which to build a firm understanding and analysis of the new ideas in Computer Science that you will meet.

The art of proof

Proof is the activity of discovering and confirming truth. Proofs in mathematics are not so far removed from coherent logical arguments of an everyday kind, of the sort a straight-thinking lawyer or politician might apply—an Obama, not a Bush! A main aim of this course is to help you harness that everyday facility to write down proofs which communicate well to other people. Here there's an art in getting the balance right: too much detail and you can't see the wood for the trees; too little and it's hard to fill in the gaps. This course is *not* about teaching you how to do very formal proofs within a formal logical system, of the

kind acceptable to machine verification—that's an important topic in itself, and one which we will touch on peripherally.

In Italy it's said that it requires two people to make a good salad dressing; a generous person to add the oil and a mean person the vinegar. Constructing proofs in mathematics is similar. Often a tolerant openness and awareness is important in discovering or understanding a proof, while a strictness and discipline is needed in writing it down. There are many different styles of thinking, even amongst professional mathematicians, yet they can communicate well through the common medium of written proof. It's important not to confuse the rigour of a well-written-down proof with the human and very individual activity of going about discovering it or understanding it. Too much of a straightjacket on your thinking is likely to stymie anything but the simplest proofs. On the other hand too little discipline, and writing down too little on the way to a proof, can leave you uncertain and lost. When you cannot see a proof immediately (this may happen most of the time initially), it can help to write down the assumptions and the goal. Often starting to write down a proof helps you discover it. You may have already experienced this in carrying out proofs by induction. It can happen that the induction hypothesis one starts out with isn't strong enough to get the induction step. But starting to do the proof even with the 'wrong' induction hypothesis can help you spot how to strengthen it.

Of course, there's no better way to learn the art of proof than by doing proofs, no better way to read and understand a proof than to pause occasionally and try to continue the proof yourself. For this reason you are encouraged to do the exercises—most of them are placed strategically in the appropriate place in the text. I have provided weekly exercise sheets which you should do, and have supervisions in. Past Tripos questions are available from the Computer Laboratory web pages. Another tip: it helps to read the relevant part of the notes, even cursorily, before the lectures.

Additional reading: The notes are self-contained. The more set-theory oriented books below are those of Devlin, Nissanke and Stanat-McAllister. Online sources such as Wikipedia can also be helpful.

Devlin, K. (2003) *Sets, Functions, and Logic, An Introduction to Abstract Mathematics.* Chapman & Hall/CRC Mathematics (3rd ed.).
Biggs, N.L. (1989). *Discrete mathematics.* Oxford University Press.
Mattson, H.F. Jr (1993). *Discrete mathematics.* Wiley.
Nissanke, N. (1999). *Introductory logic and sets for computer scientists.* Addison-Wesley.
Pólya, G. (1980). *How to solve it.* Penguin.
Stanat, D.F., and McAllister, D.F. (1977), *Discrete Mathematics in Computer Science.*Prentice-Hall.

Acknowledgements: To Hasan Amjad, Katy Edgcombe, Marcelo Fiore, Thomas Forster, Ian Grant, Martin Hyland, Frank King, Ken Moody, Alan Mycroft, Andy Pitts, Peter Robinson, Sam Staton, Dave Turner for helpful suggestions.

Contents

Chapter 1

Mathematical argument

Basic mathematical notation and methods of argument are introduced, including a review of the important principle of mathematical induction.

1.1 Logical notation

We shall use some informal logical notation in order to stop our mathematical statements getting out of hand. For statements (or assertions) A and B, we shall commonly use abbreviations like:

- A & B for (A and B), the conjunction of A and B,
- $A \Rightarrow B$ for (A implies B), which means (if A then B), and so is automatically true when A is false,
- $A \iff B$ to mean (A iff B), which abbreviates (A if and only if B), and expresses that A implies B and B implies A.

We shall also make statements by forming disjunctions (A or B), with the self-evident meaning, and negations (not A), sometimes written $\neg A$, which is true iff A is false. There is a tradition to write for instance $7 \not< 5$ instead of $\neg(7 < 5)$, which reflects what we generally say: "7 is not less than 5" rather than "not 7 is less than 5."

The statements may contain variables (or unknowns, or place-holders), as in

$$(x \leq 3) \ \& \ (y \leq 7)$$

which is true when the variables x and y over integers stand for integers less than or equal to 3 and 7 respectively, and false otherwise. A statement like $P(x, y)$, which involves variables x, y, is called a predicate (or property, or relation, or condition) and it only becomes true or false when the pair x, y stand for particular things.

We use logical quantifiers $\exists$, read "there exists", and $\forall$, read " for all". Then you can read assertions like

$$\exists x.\ P(x)$$

as abbreviating "for some x, $P(x)$" or "there exists x such that $P(x)$", and

$$\forall x.\ P(x)$$

as abbreviating " for all x, $P(x)$" or "for any x, $P(x)$". The statement

$$\exists x, y, \cdots, z.\ P(x, y, \cdots, z)$$

abbreviates

$$\exists x \exists y \cdots \exists z.\ P(x, y, \cdots, z),$$

and

$$\forall x, y, \cdots, z.\ P(x, y, \cdots, z)$$

abbreviates

$$\forall x \forall y \cdots \forall z.\ P(x, y, \cdots, z).$$

Sometimes you'll see a range for the quantification given explicitly as in $\forall x\,(0 < x \leq k).\ P(x)$. Later, we often wish to specify a set X over which a quantified variable ranges. Then one writes $\forall x \in X.\ P(x)$—read "for all x in X, $P(x)$"— instead of $\forall x.\ x \in X \Rightarrow P(x)$, and $\exists x \in X.\ P(x)$ instead of $\exists x.\ x \in X\ \&\ P(x)$.

There is another useful notation associated with quantifiers. Occasionally one wants to say not just that there exists some x satisfying a property $P(x)$ but also that x is the *unique* object satisfying $P(x)$. It is traditional to write

$$\exists! x.\ P(x)$$

as an abbreviation for

$$(\exists x.\ P(x))\ \&\ (\forall y, z.\ P(y)\ \&\ P(z) \Rightarrow y = z)$$

which means that there is some x satisfying the property $P(x)$ and also that if any y, z both satisfy the property they are equal. This expresses that there exists a unique x satisfying $P(x)$.

Occasionally, and largely for abbreviation, we will write *e.g.*, $X =_{def} E$ to mean that X is defined to be E. Similarly, we will sometimes use *e.g.*, $P(x) \Leftrightarrow_{def} A$ in defining a property in terms of an expression A.

Exercise 1.1 What is the difference between $\forall x.(\exists y.P(x,y))$ and $\exists y.(\forall x.P(x,y))$? [You might like to consider $P(x,y)$ to mean "x loves y."] □

1.2 Patterns of proof

There is no magic formula for discovering proofs in anything but the simplest contexts. However often the initial understanding of a problem suggests a general pattern of proof. Patterns of proof like those below appear frequently, often locally as ingredients of a bigger proof, and are often amongst the first things to try. It is perhaps best to tackle this section fairly quickly at a first reading, and revisit it later when you have had more experience in doing proofs.

1.2.1 Chains of implications

To prove an $A \Rightarrow B$ it suffices to show that starting from the assumption A one can prove B. Often a proof of $A \Rightarrow B$ factors into a chain of implications, each one a manageable step:

$$\begin{aligned} A &\Rightarrow A_1 \\ &\Rightarrow \cdots \\ &\Rightarrow A_n \\ &\Rightarrow B\ . \end{aligned}$$

This really stands for

$$A \Rightarrow A_1,\ A_1 \Rightarrow A_2, \cdots, A_n \Rightarrow B\ .$$

One can just as well write "Therefore" (or "$\therefore$") between the different lines, and this is preferable if the assertions $A_1, \cdots, A_n$ are large.

A bi-implication $A \iff B$ stands for both $A \Rightarrow B$ and $B \Rightarrow A$. One often sees a proof of $A \iff B$ broken down into a chain

$$\begin{aligned} A &\iff A_1 \\ &\iff \cdots \\ &\iff A_n \\ &\iff B\ . \end{aligned}$$

A common mistake is not to check the equivalence in the backwards direction, so that while the implication A_{i-1} to A_i is obvious enough, the reverse implication from A_i to A_{i-1} is unclear, in which case an explanation is needed, or even untrue. Remember, while a proof of $A \iff B$ very often does factor into a proof of $A \Rightarrow B$ and $B \Rightarrow A$, the proof route taken in showing $B \Rightarrow A$ needn't be the reverse of that taken in showing $A \Rightarrow B$.

1.2.2 Proof by contradiction

The method of proof by contradiction was known to the ancients and carries the Latin name *reductio ad absurdum*. Sometimes the only known proof of an assertion A is by contradiction. In a proof by contradiction, to show A, one shows that assuming $\neg A$ leads to a conclusion which is false. We have thus shown $\neg A$ is not the case, so A.

That $\sqrt{2}$ is irrational was a dreadful secret known to the followers of Pythagoras. The proof is a proof by contradiction: Assume, with the aim of obtaining a contradiction, that $\sqrt{2}$ is rational, *i.e.* $\sqrt{2} = a/b$ where a and b are integers with no common prime factors. Then, $2b^2 = a^2$. Therefore 2 divides a, so $a = 2a_0$ for some integer a_0. But then $b^2 = 2a_0^2$. So 2 also divides b—a contradiction.

Beware: a "beginner's mistake" is an infatuation with proof by contradiction, leading to its use even when a direct proof is at least as easy.

Exercise 1.2 Show for any integer m that $\sqrt{m}$ is rational iff m is a square, *i.e.* $m = a^2$ for some integer a.[1] □

Sometimes one shows $A \Rightarrow B$ by proving its *contrapositive* $\neg B \Rightarrow \neg A$. Showing the soundness of such an argument invokes proof by contradiction. To see that $A \Rightarrow B$ follows from the contrapositive, assume we have $\neg B \Rightarrow \neg A$. We want to show $A \Rightarrow B$. So assume A. Now we use proof by contradiction to deduce B as follows. Assume $\neg B$. Then from $\neg B \Rightarrow \neg A$ we derive $\neg A$. But now we have both A and $\neg A$—a contradiction. Hence B.

1.2.3 Argument by cases

The truth of $(A_1 \text{ or } \cdots \text{ or } A_k) \Rightarrow C$ certainly requires the truth of $A_1 \Rightarrow C$, ..., and $A_k \Rightarrow C$. Accordingly, most often a proof of $(A_1 \text{ or } \cdots \text{ or } A_k) \Rightarrow C$ breaks down into k cases, showing $A_1 \Rightarrow C$, ..., and $A_k \Rightarrow C$. An example:

Proposition For all nonnegative integers $a > b$ the difference of squares $a^2 - b^2$ does not give a remainder of 2 when divided by 4.

Proof. We observe that

$$a^2 - b^2 = (a + b)(a - b) \ .$$

Either (i) a and b are both even, (ii) a and b are both odd, or (iii) one of a, b is even and the other odd.[2] We show that in all cases $a^2 - b^2$ does not give remainder 2 on division by 4.

Case (i): both a and b are even. In this case $a^2 - b^2$ is the product of two even numbers so divisible by 4, giving a remainder 0 and not 2.

Case (ii): both a and b are odd. Again $a^2 - b^2$ is the product of two even numbers so divisible by 4.

Case(iii): one of a and b is even and one odd. In this case both $a + b$ and $a - b$ are odd numbers. Their product which equals $a^2 - b^2$ is also odd. If $a^2 - b^2$ gave remainder 2 on division by 4 it would be even—a contradiction. □

1.2.4 Existential properties

To prove $\exists x.\ A(x)$ it suffices to exhibit an object a such that $A(a)$. Often proofs of existentially quantified statements do have this form. We'll see examples where this is not the case however (as in showing the existence of transcendental numbers). For example, sometimes one can show $\exists x.\ A(x)$ by obtaining a contradiction from its negation *viz.* $\forall x.\ \neg A(x)$ and this need not exhibit an explicit object a such that $A(a)$.

Exercise 1.3 Suppose 99 passengers are assigned to one of two flights, one to Almeria and one to Barcelona. Show one of the flights has at least 50 passengers assigned to it. (Which flight is it?) □

[1] Plato reported that the irrationality of $\sqrt{p}$ was known for primes p up to 17, which suggests that the ancient Greeks didn't have the general argument. But they didn't have the benefit of algebra to abbreviate their proofs.

[2] In checking the basic facts about even and odd numbers used in this proof it's helpful to remember that an even number is one of the form $2k$, for a nonnegative integer k, and that an odd number has the form $2k + 1$, for a nonnegative integer k.

1.2.5 Universal properties

The simplest conceivable way to prove $\forall x.\ A(x)$ is to let x be an arbitrary element and then show $A(x)$. But this only works in the easiest of cases. More often than not the proof requires a knowledge of how the elements x are built up, and this is captured by induction principles. The most well-known such principle is *mathematical induction*, which deserves a section to itself.

1.3 Mathematical induction

We review mathematical induction and some of its applications. Mathematical induction is an important proof technique for establishing a property holds of all nonnegative integers 0, 1, 2, ..., n, ...

The principle of mathematical induction
To prove a property $A(x)$ for all nonnegative integers x it suffices to show

- the *basis* $A(0)$, and
- the *induction step*, that $A(n) \Rightarrow A(n+1)$, for all nonnegative integers n.

(The property $A(x)$ is called the *induction hypothesis*.)

A simple example of mathematical induction:

Proposition 1.4 *For all nonnegative integers n*

$$0 + 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2} \ .$$

Proof. By mathematical induction, taking as induction hypothesis the property $P(n)$ of a nonnegative integer n that

$$0 + 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2} \ .$$

Basis: The sum of the series consisting of just 0 is clearly 0. Also $0(0+1)/2 = 0$. Hence we have established the basis of the induction $P(0)$.

Induction step: Assume $P(n)$ for a nonnegative integer n. Then adding $(n+1)$ to both sides of the corresponding equation yields

$$0 + 1 + 2 + 3 + \cdots + n + (n+1) = \frac{n(n+1)}{2} + (n+1) \ .$$

Now, by simple algebraic manipulation we derive

$$\frac{n(n+1)}{2} + (n+1) = \frac{(n+1)((n+1)+1)}{2} \ .$$

Thus

$$0 + 1 + 2 + 3 + \cdots + n + (n+1) = \frac{(n+1)((n+1)+1)}{2} \ ,$$

and we have established $P(n+1)$. Therefore $P(n) \Rightarrow P(n+1)$, and we have shown the induction step.

By mathematical induction we conclude that $P(n)$ is true for all nonnegative integers. □

The proof above is perhaps overly pedantic, but it does emphasise that, especially in the beginning, it is very important to state the induction hypothesis clearly, and spell out the basis and induction step of proofs by mathematical induction.

There's another well-known way to prove Proposition 1.4 spotted by Gauss in kindergarten. Asked to sum together all numbers from 1 to 100 he didn't go away—presumably the goal in setting the exercise, but replied 5,050, having observed that by aligning two copies of the sum, one in reverse order,

$$\begin{array}{ccccccccc} 1 & + & 2 & + & \cdots & + & 99 & + & 100 \\ 100 & + & 99 & + & \cdots & + & 2 & + & 1 \end{array}$$

each column—and there are 100 of them—summed to 101; so that twice the required sum is 100×101.

Exercise 1.5 Prove that 7 divides $2^{4n+2} + 3^{2n+1}$ for all nonnegative integers n.

We can also use mathematical induction to establish a definition over all the nonnegative integers.

Definition by mathematical induction
To define a function f on all nonnegative integers x it suffices to define

- $f(0)$, the function on 0, and
- $f(n+1)$ in terms of $f(n)$, for all nonnegative integers n.

For example, the factorial function $n! = 1 \cdot 2 \cdots (n-1) \cdot n$ can be defined by the mathematical induction

$$\begin{aligned} 0! &= 1 \\ (n+1)! &= n! \cdot (n+1) \ . \end{aligned}$$

Given a series $x_0, x_1, \ldots, x_i, \ldots$ we can define the sum

$$\Sigma_{i=0}^{n} x_i = x_0 + x_1 + \cdots + x_n$$

by mathematical induction:[3]

$$\begin{aligned} \Sigma_{i=0}^{0} x_i &= x_0 \\ \Sigma_{i=0}^{n+1} x_i &= (\Sigma_{i=0}^{n} x_i) + x_{n+1} \ . \end{aligned}$$

Exercise 1.6 Let a and d be real numbers. Prove by mathematical induction that for all nonnegative integers n that

$$a + (a+d) + (a+2d) + \cdots + (a+(n-1)d) + (a+nd) = \frac{(n+1)(2a+nd)}{2} \ .$$

□

Exercise 1.7 Prove by mathematical induction that for all nonnegative integers n that

$$1 + 1/2 + 1/4 + 1/8 + \cdots + 1/2^n = 2 - \frac{1}{2^n} \ .$$

Let a and r be real numbers. Prove by mathematical induction that for all nonnegative integers n that

$$a + a \cdot r + a \cdot r^2 + \cdots + a \cdot r^n = \frac{a(1 - r^{n+1})}{1 - r} \ .$$

□

Exercise 1.8 The number of r combinations from $n \geq r$ elements

$${}^{n}C_r =_{def} \frac{n!}{(n-r)!r!}$$

expresses the number of ways of choosing r things from n elements.
(i) Show that

$$\begin{aligned} {}^{0}C_0 &= 1 \\ {}^{n+1}C_r &= \frac{(n+1)}{r} \cdot \ {}^{n}C_{r-1} \end{aligned}$$

for all nonnegative integers r, n with $r \leq n+1$.

[3] In the exercises it is recommended that you work with the more informal notation $x_0 + x_1 + \cdots + x_n$, and assume obvious properties such as that the sum remains the same under rearrangement of its summands. Such an 'obvious' property can be harder to spot, justify and handle with the more formal notation $\Sigma_{i=0}^{n} x_i$.

(ii) Show that

$$^{n+1}C_r = {}^n C_{r-1} + {}^n C_r$$

for all nonnegative integers r, n with $0 < r \leq n$.
(iii) Prove by mathematical induction that

$$^nC_0 + {}^n C_1 + \cdots + {}^n C_r + \cdots + {}^n C_n = 2^n$$

for all nonnegative integers n.[4] □

Sometimes it is inconvenient to start a mathematical induction at basis 0. It might be easier to start at some other integer b (possibly negative).

Mathematical induction from basis b
To prove a property $P(x)$ for all integers $x \geq b$ it suffices to show

- the *basis* $P(b)$, and
- the *induction step*, that $P(n) \Rightarrow P(n+1)$, for all integers $n \geq b$.

In fact this follows from ordinary mathematical induction (with basis 0) but with its induction hypothesis modified to be $P(b+x)$. Similarly it can sometimes be more convenient to give a definition by induction starting from basis b an integer different from 0.

Exercise 1.9 Write down the principle for definition by mathematical induction starting from basis an integer b. □

Exercise 1.10 Prove that 13 divides $3^{n+1} + 4^{2n-1}$ for all integers $n \geq 1$. □

Exercise 1.11 Prove $n^2 > 2n$ for all $n \geq 3$. □

Exercise 1.12 Prove by mathematical induction that

$$1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{1}{6}n(n+1)(2n+1)$$

for all integers $n \geq 1$.

□

Exercise 1.13 A *triomino* is an L-shaped pattern made from three square tiles. A $2^n \times 2^n$ chessboard, with squares the same size as the tiles, has an arbitrary square painted purple. Prove that the chessboard can be covered with triominoes so that only the purple square is exposed.
[Use mathematical induction: basis $n = 1$; the inductive step requires you to find four similar but smaller problems.] □

Tower of Hanoi

The tower of Hanoi is a puzzle invented by E. Lucas in 1883. The puzzle starts with a stack of discs arranged from largest on the bottom to smallest on top placed on a peg, together with two empty pegs. The puzzle asks for a method, comprising a sequence of moves, to transfer the stack from one peg to another, where moves are only allowed to place smaller discs, one at a time, on top of larger discs. We describe a method, in fact optimal, which requires $2^n - 1$ moves, starting from a stack of n discs.

Write $T(n)$ for the number of moves the method requires starting from a stack of n discs. When $n = 0$ it suffices to make no moves, and $T(0) = 0$. Consider starting from a stack of $n+1$ discs. Leaving the largest disc unmoved we can use the method for n discs to transfer the stack of n smaller discs to another peg—this requires $T(n)$ moves. Now we can move the largest disc to the remaining empty peg—this requires

[4] From school you probably know more intuitive ways to establish (i) and (ii) by considering the coefficients of powers of x in $(1+x)^n$; the *binomial theorem* asserts the equality of combinations ^nC_r and coefficients of x^r.

1 move. Using the method for n discs again we can put the stack of n smaller discs back on top of the largest disc—this requires a further $T(n)$ moves. We have succeeded in transferring the stack of $n+1$ discs to another peg. In total the method uses

$$T(n) + 1 + T(n) = 2 \cdot T(n) + 1$$

moves to transfer $n+1$ discs to a new peg. Hence,

$$\begin{aligned} T(0) &= 0 \\ T(n+1) &= 2 \cdot T(n) + 1 \end{aligned} .$$

Exercise 1.14 Prove by mathematical induction that $T(n) = 2^n - 1$ for all nonnegative integers n. □

Course-of-values induction (Strong induction)

A difficulty with induction proofs is finding an appropriate induction hypothesis. Often the property, say $B(x)$, one is originally interested in showing holds for all x isn't strong enough for the induction step to go through; it needs to be strengthened to $A(x)$, so $A(x) \Rightarrow B(x)$, to get a suitable induction hypothesis. Devising the induction hypothesis from the original goal and what's needed to carry through the induction step often requires insight.

One way to strengthen a hypothesis $A(x)$ is to assume it holds for all nonnegative numbers below or equal to x, *i.e.*

$$\forall k\ (0 \leq k \leq x).\ A(k)\ .$$

This strengthening occurs naturally when carrying out an induction step where the property of interest $A(x+1)$ may depend not just on $A(x)$ but also on $A(k)$ at several previous values k. With this strengthened induction hypothesis, mathematical induction becomes: To prove a property $A(x)$ for all nonnegative numbers x it suffices to show

- the *basis* $\forall k\ (0 \leq k \leq 0).\ A(k)$, and
- the *induction step*, that

$$[\forall k\ (0 \leq k \leq n).\ A(k)] \Rightarrow [\forall k\ (0 \leq k \leq n+1).\ A(k)]\ ,$$

  for all nonnegative integers n.

Tidying up, we obtain the principle of course-of-values induction (sometimes called 'strong' or 'complete' induction).

Course-of-values induction
To prove a property $A(x)$ for all nonnegative integers x it suffices to show that

- $[\forall k\ (0 \leq k < n).\ A(k)] \Rightarrow A(n)$,

for all nonnegative integers n.

In other words, according to course-of-values induction to prove the property $A(x)$ for all nonnegative integers x, it suffices to prove the property holds at n from the assumption that property holds over the 'course of values' 0, 1, ..., $n-1$ below a nonnegative integer n, *i.e.*that $A(n)$ follows from $A(0)$, $A(1)$, ..., and $A(n-1)$. Notice that when $n = 0$ the course of values below 0 is empty, and it is not unusual that the case $n = 0$ has to be considered separately.

There is an accompanying method to define a function on all the nonnegative integers by course-of-values induction: To define an operation f on all nonnegative integers n it suffices to define

- $f(n)$, the function on n, in terms of the results $f(k)$ of the function on k for $0 \leq k < n$.

Definition by course-of-values induction is used directly in recurrence relations such as that for defining the Fibonacci numbers. The Fibonacci numbers $0, 1, 1, 2, 3, 5, 8, 13, \ldots$ are given by the clauses

$$\mathit{fib}(0) = 0, \quad \mathit{fib}(1) = 1, \quad \mathit{fib}(n) = \mathit{fib}(n-1) + \mathit{fib}(n-2) \text{ for } n > 1 \ ,$$

in which the nth Fibonacci number is defined in terms of the two preceding numbers.

Just as with mathematical induction it is sometimes more convenient to start a course-of-values induction at an integer other than 0.

Course-of-values induction from integer b
To prove a property $A(x)$ for all nonnegative integers $x \geq b$ it suffices to show that

- $[\forall k \ (b \leq k < n).\ A(k)] \Rightarrow A(n)$,

for all integers $n \geq b$.

This principle follows from course-of-values induction (starting at 0) but with induction modified to $A(x + b)$. There's an analogous definition by course-of-values induction from integer b.

We can use course-of-values induction to prove an important theorem known to Euclid. Recall, a prime is an integer $p > 1$ such that if $p = ab$, the product of positive integers, then either $a = 1$ or $b = 1$. Equivalently, an integer $p > 1$ is prime if whenever p divides a product of positive integers ab, then p divides a or p divides b.

Theorem 1.15 *Every integer $n \geq 2$ can be written as a product of prime numbers.*

Proof. For integers $n \geq 2$, we take the induction hypothesis $A(n)$ to be: n can be written as a product of primes. We prove this by course-of-values induction (starting from 2). For this it suffices to show that

$$[\forall m \ (2 \leq m < n).\ A(m)] \Rightarrow A(n) \, ,$$

for all $n \geq 2$.

Assume $\forall m \ (2 \leq m < n).\ A(m)$. There are two cases in showing $A(n)$.

In the case where n is prime, n is product of one prime, so automatically $A(n)$.

Otherwise, in the case where n is not prime, then $n = ab$ for some integers a and b, where $2 \leq a < n$ and $2 \leq b < n$.[5] By the assumption, both $A(a)$ and $A(b)$, *i.e.* both a and b can be written as products of primes. It follows that their product ab can also be written in this form, *i.e.* $A(n)$.

By course-of-values induction we have established $A(n)$ for all integers $n \geq 2$. □

Remark Often it is said that every positive integer (including 1) can be written as a product of primes. This relies on the convention that the empty product of numbers, and in particular primes, is 1.

Exercise 1.16 There are five equally-spaced stepping stones in a straight line across a river. The distance d from the banks to the nearest stone is the same as that between the stones. You can hop distance d or jump $2d$. So for example you could go from one river bank to the other in 6 hops. Alternatively you might first jump, then hop, then jump, then hop. How many distinct ways could you cross the river (you always hop or jump forwards, and don't overshoot the bank)?

Describe how many distinct ways you could cross a river with n similarly spaced stepping stones. □

Exercise 1.17 Let

$$\varphi = \frac{1 + \sqrt{5}}{2}$$

[5] Here we are using the fact that if a positive integer a properly divides another n, then $a < n$.

—called the golden ratio. Show that both φ and $-1/\varphi$ satisfy the equation

$$x^2 = x + 1 \ .$$

Deduce that they both satisfy

$$x^n = x^{n-1} + x^{n-2} \ .$$

Using this fact, prove by course-of-values induction that the nth Fibonacci number,

$$fib(n) = \frac{\varphi^n - (-1/\varphi)^n}{\sqrt{5}} \ .$$

[Consider the cases $n = 0$, $n = 1$ and $n > 1$ separately.] □

Why does the principle of mathematical induction hold? On what key feature of the nonnegative integers does mathematical induction depend?

Suppose that not all nonnegative integers satisfy a property $A(x)$. By course-of-values induction, this can only be if

$$[\forall k \ (b \leq k < l).\ A(k)] \Rightarrow A(l) \ ,$$

is false for some nonnegative integer l, *i.e.* that

$$[\forall k \ (b \leq k < l).\ A(k)] \ \text{ and } \ \neg A(l) \ ,$$

for some nonnegative integer l. This says that l is the least nonnegative integer at which the property $A(x)$ fails.

In fact this is the key feature of the nonnegative integers on which mathematical induction depends:

> **The least-number principle:** Assuming that a property fails to hold for all nonnegative integers, there is a least nonnegative integer at which the property fails (the *'least counterexample,'* or *'minimal criminal'*).

We derived the least-number principle from course-of-values induction, and in turn course-of-values induction was derived from mathematical induction. Conversely we can derive mathematical induction from the least-number principle. Let $A(x)$ be a property of the nonnegative integers for which both the basis, $A(0)$, and the induction step, $A(n) \Rightarrow A(n+1)$ for all nonnegative integers n, hold. Then the least-number principle implies that $A(x)$ holds for all nonnegative integers. Otherwise, according to the least number principle there would be a least nonnegative integer l for which $\neg A(l)$. If $l = 0$, then $\neg A(0)$. If $l \neq 0$, then $l = n + 1$ for some nonnegative integer n for which $A(n)$ yet $\neg A(n+1)$. Either the basis or the induction step would be contradicted.

Sometimes it can be more convenient to use the least-number principle directly in a proof rather than proceed by induction. The following theorem is an important strengthening of Theorem 1.15—it provides a unique decomposition of a positive integer into its prime factors. The proof, which uses the least-number principle, is marginally easier than the corresponding proof by course-of-values induction.

Theorem 1.18 *(Fundamental theorem of arithmetic) Every integer $n \geq 1$ can be written* uniquely *as a product of prime numbers in ascending order, i.e. as a product*

$$p_1^{r_1} \cdots p_k^{r_k}$$

where $p_1, \cdots, p_k$ are primes such that $p_1 < \cdots < p_k$ and $r_1, \cdots, r_k$ are positive integers.

Proof. By Theorem 1.15, and adopting the convention that an empty product is 1, we know that every integer $n \geq 1$ can be written as a product of prime numbers. When ordered this gives a product of prime numbers in ascending order. But it remains to prove its uniqueness.

We shall use the fact about primes that if a prime p divides a product of positive integers $n_1 \cdots n_k$, then p divides a factor n_i. We shall also use that, if a prime p divides a prime q then $p = q$.

Let $A(n)$ express the property that n can be written uniquely as a product of primes in ascending order. We are interested in the showing that $A(n)$ holds for all positive integers n. Suppose, to obtain a contradiction, that $A(n)$ fails to hold for all positive integers n. By the least-number principle, there is a least positive integer n_0 at which $\neg A(n_0)$, *i.e.*

$$n_0 = p_1^{r_1} \cdots p_k^{r_k} = q_1^{s_1} \cdots q_l^{s_l}$$

are two distinct products of prime numbers in ascending order giving n_0.

As at least one of these products has to be nonempty, there is some prime dividing n_0. Amongst such there is a least prime d which divides n_0. We argue that this least prime d must equal both p_1 and q_1. To see this, notice that d must divide one of the prime factors p_j, but then $d = p_j$. If $j \neq 1$ we would obtain the contradiction that p_1 was a lesser prime than d dividing n_0. Hence $d = p_1$, and by a similar argument $d = q_1$ too.

Now, dividing both products by $d = p_1 = q_1$, yields

$$m = p_1^{r_1 - 1} \cdots p_k^{r_k} = q_1^{s_1 - 1} \cdots q_l^{s_l} .$$

where $m < n_0$. Whether or not $r_1 = 1$ or $s_1 = 1$, we can obtain two distinct products of primes in ascending order giving m. But this yields $\neg A(m)$, a contradiction as $m < n_0$ —by definition the least such positive integer. □

There are many structures other than the nonnegative integers for which if a property does not hold everywhere then it fails to hold at some 'least' element. These structures also possess induction principles analogous to mathematical induction. But describing the structures, and their induction principles (examples of well-founded induction to be studied later), would be very hard without the language and concepts of set theory. (Well-founded induction plays an essential part in establishing the termination of programs.)

Chapter 2

Sets and Logic

This chapter introduces sets. In it we study the structure on subsets of a set, operations on subsets, the relations of inclusion and equality on sets, and the close connection with propositional logic.

2.1 Sets

A *set* (or class) is an (unordered) collection of objects, called its *elements* or *members*. We write $a \in X$ when a is an element of the set X. We read $a \in X$ as "a is a member of X" or "a is an element of X" or "a belongs to X", or in some contexts as just "a in X". Sometimes we write *e.g.* $\{a, b, c, \cdots\}$ for the set of elements $a, b, c, \cdots$. Some important sets:

$\emptyset$ the empty set with no elements, sometimes written $\{\ \}$. (Contrast the empty set $\emptyset$ with the set $\{\emptyset\}$ which is a singleton set in the sense that it has a single element, *viz.* the empty set.)

$\mathbb{N}$ the set of natural numbers $\{1, 2, 3, \cdots\}$.

$\mathbb{N}_0$ the set of natural numbers with zero $\{0, 1, 2, 3, \cdots\}$. (This set is often called ω.)

$\mathbb{Z}$ the set of integers, both positive and negative, with zero.

$\mathbb{Q}$ the set of rational numbers.

$\mathbb{R}$ the set of real numbers.

In computer science we are often concerned with sets of strings of symbols from some alphabet, for example the set of strings accepted by a particular automaton.

A set X is said to be a *subset* of a set Y, written $X \subseteq Y$, iff every element of X is an element of Y, *i.e.*

$$X \subseteq Y \iff \forall z \in X.\ z \in Y.$$

Synonymously, then we also say that X is *included* in Y.

A set is determined solely by its elements in the sense that two sets are equal iff they have the same elements. So, sets X and Y are equal, written $X = Y$, iff every element of A is a element of B and *vice versa*. This furnishes a method for showing two sets X and Y are equal and, of course, is equivalent to showing $X \subseteq Y$ and $Y \subseteq X$.

Sets and properties

Sometimes a set is determined by a property, in the sense that the set has as elements precisely those which satisfy the property. Then we write

$$X = \{x \mid P(x)\},$$

meaning the set X has as elements precisely all those x for which the property $P(x)$ is true. If X is a set and $P(x)$ is a property, we can form the set

$$\{x \in X \mid P(x)\}$$

which is another way of writing

$$\{x \mid x \in X\ \&\ P(x)\}.$$

This is the subset of X consisting of all elements x of X which satisfy $P(x)$.

When we write $\{a_1, \cdots, a_n\}$ we can understand this as the set

$$\{x \mid x = a_1 \text{ or } \cdots \text{ or } x = a_n\} .$$

Exercise 2.1 This question is about strings built from the symbols a's and b's.
For example aab, $ababaaa$, etc. are strings, as is the empty string ε.

(i) Describe the set of strings x which satisfy

$$ax = xa .$$

Justify your answer.

(ii) Describe the set of strings x which satisfy

$$ax = xb .$$

Justify your answer. □

2.2 Set laws

2.2.1 The Boolean algebra of sets

Assume a set U. Subsets of U support operations closely related to those of logic. The key operations are

Union	$A \cup B = \{x \mid x \in A \text{ or } x \in B\}$
Intersection	$A \cap B = \{x \mid x \in A \ \& \ x \in B\}$
Complement	$A^c = \{x \in U \mid x \notin A\}$.

Notice that the complement operation makes sense only with respect to an understood 'universe' U. A well-known operation on sets is that of set difference $A \setminus B$ defined to be $\{a \in A \mid a \notin B\}$; in the case where A and B are subsets of U set difference $A \setminus B = A \cap B^c$. Two sets A and B are said to be *disjoint* when $A \cap B = \emptyset$, so they have no elements in common.

Exercise 2.2 Let $A = \{1, 3, 5\}$ and $B = \{2, 3\}$. Write down explicit sets for:

(i) $A \cup B$ and $A \cap B$.

(ii) $A \setminus B$ and $B \setminus A$.

(iii) $(A \cup B) \setminus B$ and $(A \setminus B) \cup B$. □

The operations $\cup$ and $\cap$ are reminiscent of sum and multiplication on numbers, though they don't satisfy quite the same laws, *e.g.* we have $A \cup A = A$ generally while $a + a = a$ only when a is zero. Just as the operations sum and multiplication on numbers form an algebra so do the above operations on subsets of U. The algebra on sets and its relation to logical reasoning were laid bare by George Boole (1815-1864) in his "Laws of thought," and are summarised below. The laws take the form of algebraic identities between set expressions. (An algebra with operations $\cup, \cap$, and $(-)^c$ satisfying these laws is called a *Boolean algebra*.) Notice the laws $A \cup \emptyset = A$ and $A \cap U = A$ saying that $\emptyset$ and U behave as units with respect to the operations of union and intersection respectively.

The Boolean identities for sets: Letting A, B, C range over subsets of U,

Associativity	$A \cup (B \cup C) = (A \cup B) \cup C$	$A \cap (B \cap C) = (A \cap B) \cap C$
Commutativity	$A \cup B = B \cup A$	$A \cap B = B \cap A$
Idempotence	$A \cup A = A$	$A \cap A = A$
Empty set	$A \cup \emptyset = A$	$A \cap \emptyset = \emptyset$
Universal set	$A \cup U = U$	$A \cap U = A$
Distributivity	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
Absorption	$A \cup (A \cap B) = A$	$A \cap (A \cup B) = A$
Complements	$A \cup A^c = U$	$A \cap A^c = \emptyset$
	$(A^c)^c = A$	
De Morgan's laws	$(A \cup B)^c = A^c \cap B^c$	$(A \cap B)^c = A^c \cup B^c$.

To show such algebraic identities between set expressions, one shows that an element of the set on the left is an element of the set on the right, and *vice versa*. For instance suppose the task is to prove

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

for all sets A, B, C. We derive

$$\begin{aligned} x \in A \cap (B \cup C) &\iff x \in A \text{ and } (x \in B \cup C) \\ &\iff x \in A \text{ and } (x \in B \text{ or } x \in C) \\ &\iff (x \in A \text{ and } x \in B) \text{ or } (x \in A \text{ and } x \in C) \\ &\iff x \in A \cap B \text{ or } x \in A \cap C \\ &\iff x \in (A \cap B) \cup (A \cap C) \ . \end{aligned}$$

The 'dual' of the identity is

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

To prove this we can 'dualize' the proof just given by interchanging the symbols $\cup, \cap$ and the words 'or' and 'and.' There is a duality principle for sets, according to which any identity involving the operations $\cup, \cap$ remains valid if the symbols $\cup, \cap$ are interchanged throughout. We can also prove the dual of identities directly, just from the laws of sets, making especial use of the De Morgan laws. For example, once we know

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

for all sets A, B, C we can derive its dual in the following way. First deduce that

$$A^c \cap (B^c \cup C^c) = (A^c \cap B^c) \cup (A^c \cap C^c) \ ,$$

for sets A, B, C. Complementing both sides we obtain

$$(A^c \cap (B^c \cup C^c))^c = ((A^c \cap B^c) \cup (A^c \cap C^c))^c \ .$$

Now argue by De Morgan's laws and laws for complements that the left-hand-side is

$$\begin{aligned} (A^c \cap (B^c \cup C^c))^c &= (A^c)^c \cup ((B^c)^c \cap (C^c)^c) \\ &= A \cup (B \cap C) \ , \end{aligned}$$

while the right-hand-side is

$$\begin{aligned}((A^c \cap B^c) \cup (A^c \cap C^c))^c &= (A^c \cap B^c)^c \cap (A^c \cap C^c)^c \\ &= ((A^c)^c \cup (B^c)^c) \cap ((A^c)^c \cup (C^c)^c) \\ &= (A \cup B) \cap (A \cup C) \ .\end{aligned}$$

We have deduced the dual identity

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

Exercise 2.3 Prove the remaining set identities above. □

The set identities allow deductions like those of school algebra. For example, we can derive

$$U^c = \emptyset \qquad\qquad \emptyset^c = U \ .$$

To derive the former, using the Universal-set and Complements laws:

$$U^c = U^c \cap U = \emptyset \ ,$$

Then by Complements on this identity we obtain $\emptyset^c = U$.

Using the Distributive laws and De Morgan laws with the Idempotence and Complements laws we can derive standard forms for set expressions. Any set expression built up from basic sets can be transformed to a union of intersections of basic sets and their complements, or alternatively as an intersection of unions of basic sets and their complements, *e.g.*:

$$\cdots \cup (A_1^c \cap A_2 \cap \cdots \cap A_k) \cup \cdots$$
$$\cdots \cap (A_1^c \cup A_2 \cup \cdots \cup A_k) \cap \cdots$$

The method is to first use the De Morgan laws to push all occurrences of the complement operation inwards so it acts just on basic sets; then use the Distributive laws to bring unions (or alternatively intersections) to the top level. With the help of the Idempotence and Complements laws we can remove redundant occurrences of basic sets. The standard forms for set expressions reappear in propositional logic as *disjunctive* and *conjunctive normal forms* for propositions.

Exercise 2.4 Using the set laws transform $(A \cap B)^c \cap (A \cup C)$ to a standard form as a union of intersections. □

The Boolean identities hold no matter how we interpret the basic symbols as sets. In fact, any identity, true for all interpretations of the basic symbols as sets, can be deduced from Boole's identities using the laws you would expect of equality; in this sense the Boolean identities listed above are *complete*.

Although the Boolean identities concern the equality of sets, they can also be used to establish the inclusion of sets because of the following facts.

Proposition 2.5 *Let A and B be sets. Then,*

$$A \subseteq B \iff A \cap B = A \ .$$

Proof. *"only if"*: Suppose $A \subseteq B$. We have $A \cap B \subseteq A$ directly from the definition of intersection. To show equality we need the converse inclusion. Let $x \in A$. Then $x \in B$ as well, by supposition. Therefore $x \in A \cap B$. Hence, $A \subseteq A \cap B$. *"if"*: Suppose $A \cap B = A$. Then $A = A \cap B \subseteq B$. □

Exercise 2.6 Let A and B be sets. Prove $A \subseteq B \iff A \cup B = B$. □

Proposition 2.7 *Let $A, B \subseteq U$. Then,*

$$A \subseteq B \iff A^c \cup B = U \ .$$

Proof. Let $A, B \subseteq U$. Then,

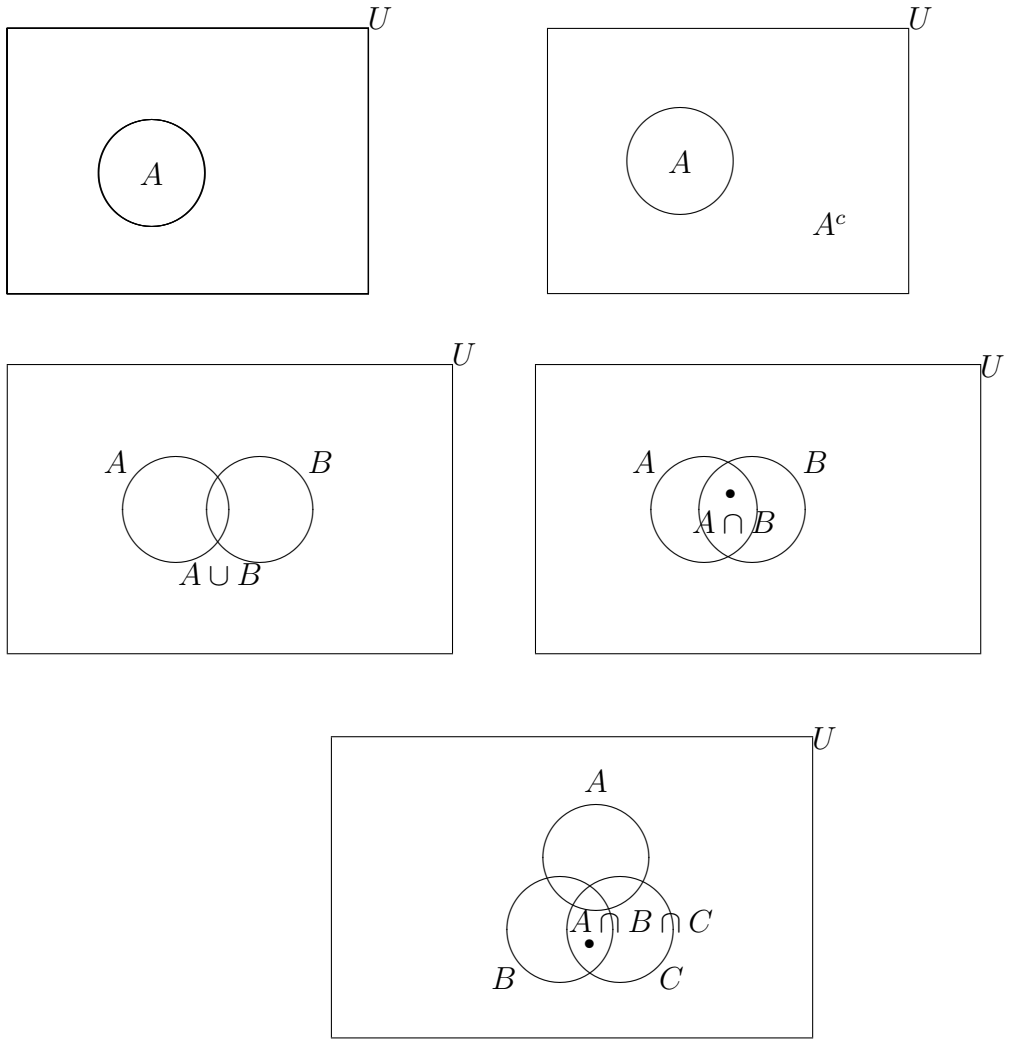
$$\begin{aligned}A \subseteq B &\iff \forall x \in U.\ x \in A \Rightarrow x \in B \\ &\iff \forall x \in U.\ x \notin A \text{ or } x \in B \\ &\iff \forall x \in U.\ x \in A^c \cup B \\ &\iff A^c \cup B = U \ .\end{aligned}$$

□

Exercise 2.8 Let $A, B \subseteq U$. Prove that $A \subseteq B \iff A \cap B^c = \emptyset$. □

2.2.2 Venn diagrams

When an expression describing a set is small it can be viewed pictorially as a *Venn diagram*[1] in which sets are represented as regions in the plane. In each diagram below the outer rectangle represents the universe U and the circles the sets A, B, C.



Exercise 2.9 Describe the set $A \cup B \cup C$ as a union of 7 disjoint sets (*i.e.*, so each pair of sets has empty intersection). □

Exercise 2.10 In a college of 100 students, 35 play football, 36 row and 24 play tiddlywinks. 13 play football and row, 2 play football and tiddlywinks but never row, 12 row and play tiddlywinks, while 4 practice all three activities. How many students participate in none of the activities of football, rowing and tiddlywinks? □

2.2.3 Boolean algebra and properties

A property $P(x)$ where $x \in U$ determines a subset of U, its *extension*, the set $\{x \in U \mid P(x)\}$. For instance U might be the set of integers $\mathbb{Z}$, when a suitable property could be "x is zero" or " x is a prime number"; the extension of the first property is the singleton set $\{0\}$, while the extension of the second is the set of primes. In many computer science applications U is a set of program states and then properties can specify the values stored in certain locations: for example "state x has value 3 in location Y and 5 in location Z." Alternatively U might consist of all the inhabitants of a country when properties of interest could be those of a census, specifying for example sex, age, household.

[1] After John Venn (1834-1923).

Logical operations on properties are paralleled by Boolean operations on their extensions as sets:

Property	Its extension as a set
$P(x)$	$\{x \in U \mid P(x)\}$
$Q(x)$ & $R(x)$	$\{x \in U \mid Q(x)\} \cap \{x \in U \mid R(x)\}$
$Q(x)$ or $R(x)$	$\{x \in U \mid Q(x)\} \cup \{x \in U \mid R(x)\}$
$\neg P(x)$	$\{x \in U \mid P(x)\}^c$
$Q(x) \Rightarrow R(x)$	$\{x \in U \mid Q(x)\}^c \cup \{x \in U \mid R(x)\}$

We can think of the meaning (or semantics) of a property as being the set which is its extension. Then logical operations on properties correspond to Boolean operations on sets. Two properties being equivalent corresponds to them having the same extension. The relation of entailment between properties corresponds to the relation of inclusion between sets. We can reason about properties by reasoning about sets.

2.3 Propositional logic

Much of the power of Boolean algebra of sets derives from its close connection with logic. In this section we start to make reasoning itself an object of study. We show how to view propositions as sets. This provides a a link between set theory and logic, and explains why the Boolean laws of sets that we have seen coincide with the laws of propositional logic. It will justify the transfer of laws and calculations on sets to laws and calculations on logic, and *vice versa*.

2.3.1 Boolean propositions

The first step is to present the syntax of Boolean propositions:

$$\mathtt{A}, \mathtt{B}, ... ::= \mathtt{a}, \mathtt{b}, \mathtt{c}, \cdots \mid \mathtt{T} \mid \mathtt{F} \mid \mathtt{A} \wedge \mathtt{B} \mid \mathtt{A} \vee \mathtt{B} \mid \neg \mathtt{A}$$

By which we mean a proposition, which we will typically call $\mathtt{A}, \mathtt{B}, \cdots$, is either a propositional variable from among $\mathtt{a}, \mathtt{b}, \mathtt{c} \cdots \in \mathtt{Var}$, a set of propositional variables, the proposition true $\mathtt{T}$ or the proposition false $\mathtt{F}$, or built up using the logical operations of conjunction $\wedge$, disjunction $\vee$ or negation $\neg$. We define implication $\mathtt{A} \Rightarrow \mathtt{B}$ to stand for $\neg\mathtt{A} \vee \mathtt{B}$, and $\mathtt{A} \Leftrightarrow \mathtt{B}$ as $(\mathtt{A} \Rightarrow \mathtt{B}) \wedge (\mathtt{B} \Rightarrow \mathtt{A})$. To avoid excessive brackets in writing Boolean propositions we adopt the usual convention that the operation $\neg$ binds more tightly than the two other operations $\wedge$ and $\vee$, so that $\neg\mathtt{A} \vee \mathtt{B}$ means $(\neg\mathtt{A}) \vee \mathtt{B}$.

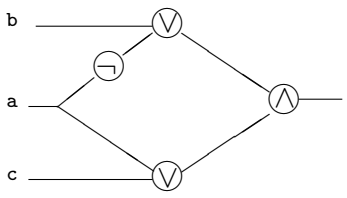
Boolean propositions are ubiquitous in science and everyday life. They are an unavoidable ingredient of almost all precise discourse, and of course of mathematics and computer science. They most often stand for simple assertions we might make about the world, once we have fixed the meaning of the basic propositional variables. For example, we might take the propositional variables to mean basic propositions such as

> "It's raining", "It's sunny", "Dave wears sunglasses",
> "Lucy carries an umbrella", ...

which would allow us to describe more complex situations with Boolean propositions, such as

> "It's sunny $\wedge$ Dave wears sunglasses $\wedge$ $\neg$(Lucy carries an umbrella)".

But, for example, Boolean propositions can also be used to stand for Boolean circuits built from and-, or- and not-gates. Then the propositional variables correspond to input wires which can be at high or low voltages, by convention understood as true $\mathtt{T}$ and false $\mathtt{F}$. For example,



is a Boolean circuit representing the Boolean proposition $(\neg\mathtt{a} \vee \mathtt{b}) \wedge (\mathtt{a} \vee \mathtt{c})$; giving particular high ($\mathtt{T}$) or low ($\mathtt{F}$) voltages to the input wires $\mathtt{a}, \mathtt{b}, \mathtt{c}$ on the left, determines, as we move from left to right, a particular value of high ($\mathtt{T}$) or low ($\mathtt{F}$) on the output wire, at the extreme right.

We can evaluate a Boolean proposition to a truth value once we are given an assignment of truth values to its propositional variables. A traditional way to do this is via the method of *truth tables*—see Section 2.3.4.

We often want to know when one Boolean proposition is equivalent to another. In particular we might want to know when one Boolean circuit can be replaced by another, presumably simpler one. Fortunately the laws for equivalence of Boolean propositions coincide with the set laws we have just seen once we read T as the universal set, F as the empty set, $\wedge$ as intersection, $\vee$ as union and $\neg$ as complementation. But why is this so? The key to the answer is to regard propositions as implicitly describing properties of situations, or states of the world, and as we've seen properties can be regarded as sets.

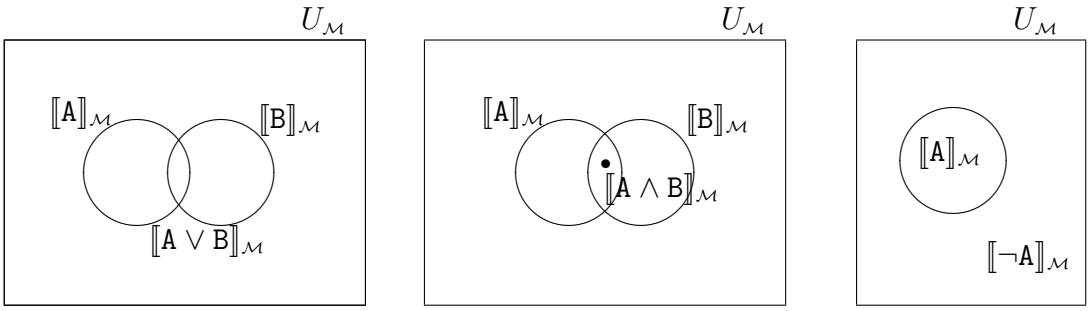
2.3.2 Models

To link the set laws with logic, we show how to interpret Boolean propositions as sets. The idea is to think of a proposition as denoting the *set* of states, or situations, or worlds, or individuals, of which the proposition is true. The states might literally be states in a computer, but the range of propositional logic is much more general and it applies to any collection of situations, individuals or things of which the properties of interest are either true or false. For this reason we allow interpretations to be very general, as formalised through the notion of model.

A *model* $\mathcal{M}$ for Boolean propositions consists of a set $U_{\mathcal{M}}$, of *states*, called the *universe* of $\mathcal{M}$, together with an interpretation $[\![\mathtt{A}]\!]_{\mathcal{M}}$ of propositions A as subsets of $U_{\mathcal{M}}$ which satisfies

$$\begin{aligned}
[\![\mathtt{T}]\!]_{\mathcal{M}} &= U_{\mathcal{M}} \\
[\![\mathtt{F}]\!]_{\mathcal{M}} &= \emptyset \\
[\![\mathtt{A} \wedge \mathtt{B}]\!]_{\mathcal{M}} &= [\![\mathtt{A}]\!]_{\mathcal{M}} \cap [\![\mathtt{B}]\!]_{\mathcal{M}} \\
[\![\mathtt{A} \vee \mathtt{B}]\!]_{\mathcal{M}} &= [\![\mathtt{A}]\!]_{\mathcal{M}} \cup [\![\mathtt{B}]\!]_{\mathcal{M}} \\
[\![\neg\mathtt{A}]\!]_{\mathcal{M}} &= [\![\mathtt{A}]\!]_{\mathcal{M}}{}^{c} \ .
\end{aligned}$$

The idea is that $[\![\mathtt{A}]\!]_{\mathcal{M}}$ is the set of states satisfying A, or put another way, the set of states which make A true. All states satisfy T so $[\![\mathtt{T}]\!]_{\mathcal{M}}$ is $U_{\mathcal{M}}$, the set of all states. No state should satisfy F so it is interpreted as the empty set of states. A clause above expresses that the set of states which satisfy $\mathtt{A} \wedge \mathtt{B}$ is the same as the set of states which satisfy A and which also satisfy B. So that clause constrains the interpretation of $\mathtt{A} \wedge \mathtt{B}$ to mean what it should mean. There are no clauses to constrain the interpretation $[\![\mathtt{b}]\!]_{\mathcal{M}}$ of propositional variables b; the interpretation $[\![\mathtt{b}]\!]_{\mathcal{M}}$ picks out the set of states which satisfy b, and in this sense it fixes the meaning of b. The following Venn diagrams illustrate $[\![\mathtt{A} \vee \mathtt{B}]\!]_{\mathcal{M}}$, $[\![\mathtt{A} \wedge \mathtt{B}]\!]_{\mathcal{M}}$ and $[\![\neg\mathtt{A}]\!]_{\mathcal{M}}$:



There are many examples of models.

Example: One model $\mathcal{S}$ close to home is to take the universe $U_{\mathcal{S}}$ as the set of students in a class and to let propositional variables $\mathtt{a}, \mathtt{b}, \mathtt{c}, \cdots$ stand for specific properties such as "Being a student of Emmanuel College", "Having a driving licence", "Being nineteen", *etc.*. We achieve this in the model by having

$$\begin{aligned}
[\![\mathtt{a}]\!]_{\mathcal{S}} &= \{x \in U_{\mathcal{S}} \mid x \text{ is an Emmanuel student}\} \ , \\
[\![\mathtt{b}]\!]_{\mathcal{S}} &= \{x \in U_{\mathcal{S}} \mid x \text{ has a driving licence}\} \ , \\
[\![\mathtt{c}]\!]_{\mathcal{S}} &= \{x \in U_{\mathcal{S}} \mid x \text{ is nineteen}\} \ , \quad \textit{etc.}
\end{aligned}$$

Then, for instance, $[\![\mathtt{a} \wedge \neg\mathtt{b}]\!]_{\mathcal{S}}$ would be the set of students in the class from Emmanuel who don't have driving licences. □

Example: We briefly sketch another example of a model, $\mathcal{H}$, which is useful in the verification of hardware.[2] (The model will reappear in another guise as the model $\mathcal{TA}$, based on truth assignments, in Section 2.3.3.) Take the universe $U_{\mathcal{H}}$ to be the set of assignments of 'high' or 'low' voltages to connection points on a wiring board for simple circuits. Let the connection points be labelled $\mathtt{a}, \mathtt{b}, \mathtt{c}, \cdots$ so an assignment would specify a voltage $V_{\mathtt{c}} \in \{$'high', 'low'$\}$ for each connection point $\mathtt{c}$. Interpret propositional variables $\mathtt{a}, \mathtt{b}, \mathtt{c} \cdots$ as standing for assertions "$\mathtt{a\,is\,high}$", "$\mathtt{b\,is\,high}$", *etc.*. We achieve this in the model by taking the interpretation so that for propositional variable b, for instance, $[\![\mathtt{b}]\!]_{\mathcal{H}}$ is the set of all assignments V for which $V_{\mathtt{b}} =$ 'high'.

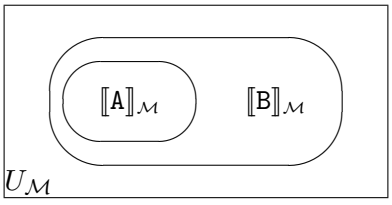
The relevance of this model for hardware verification is that basic hardware can be represented by propositions. For example, the positive pole of a battery connected to $\mathtt{b}$ would be represented as $\mathtt{b}$, while earthing $\mathtt{b}$ would be represented as $\neg\mathtt{b}$. A wire between $\mathtt{a}$ and $\mathtt{b}$ would ensure that the two connections were either both 'high' or both 'low', so be represented by $\mathtt{a} \Leftrightarrow \mathtt{b}$. A transistor with gate $\mathtt{g}$ and connections $\mathtt{s}$ and $\mathtt{d}$ would impose the condition $\mathtt{g} \Rightarrow (\mathtt{s} \Leftrightarrow \mathtt{d})$, because when the gate is 'high' the transistor behaves as a wire between $\mathtt{s}$ and $\mathtt{d}$. A more complicated circuit would be represented by the conjunction of the propositions representing its components. Another operation on circuits is that of 'hiding,' in which a specified connection is hidden in the sense that it can no longer be directly connected to. If a circuit is represented by the proposition $\mathtt{A}$ and the connection $\mathtt{b}$ is hidden, the behaviour resulting from hiding is usually represented by the proposition $\mathtt{A}[\mathtt{T}/\mathtt{b}] \vee \mathtt{A}[\mathtt{F}/\mathtt{b}]$, got by taking the disjunction of the results of instantiating $\mathtt{b}$ to $\mathtt{T}$ and to $\mathtt{F}$ in $\mathtt{A}$. But here the model begins to reveal its limitations; through hiding, a wire could become isolated from any source or earth, in which case it can be physically unrealistic to describe it as being either 'high' or 'low.' □

Example: A simple representation of a computer state (or store) is in terms of assignments of values to locations. Assume the locations have names $\mathtt{X}, \mathtt{Y}, \mathtt{Z}, \cdots$ and have integer contents. The set of states consists of all assignments of integers to locations. In this case it is sensible for the propositional variables of a model to stand for particular basic assertions about the contents of locations; for example, that "$\mathtt{X} = 3$", which would pick out the set of states for which the location $\mathtt{X}$ had contents 3, or that "$\mathtt{X} = \mathtt{Y} + \mathtt{Z}$", which would pick out the set of states at which the contents of location $\mathtt{X}$ equalled the contents of $\mathtt{Y}$ plus the contents of $\mathtt{Z}$. (The business of program verification often amounts to showing that if a program starts in state satisfying a particular proposition $\mathtt{A}$, then after successful execution it ends in a state satisfying another particular proposition $\mathtt{B}$.) □

Validity and entailment

Using models we can make precise when a proposition is valid (*i.e.*, always true) and when one proposition entails or is equivalent to another.

Definition: For a proposition $\mathtt{A}$, say $\mathtt{A}$ is *valid* in $\mathcal{M}$ iff $[\![\mathtt{A}]\!]_{\mathcal{M}} = U_{\mathcal{M}}$. That $\mathtt{A}$ is valid in $\mathcal{M}$ means $\mathtt{A}$ is true at all states of $\mathcal{M}$.
For propositions $\mathtt{A}$ and $\mathtt{B}$, say $\mathtt{A}$ *entails* $\mathtt{B}$ in $\mathcal{M}$ iff $[\![\mathtt{A}]\!]_{\mathcal{M}} \subseteq [\![\mathtt{B}]\!]_{\mathcal{M}}$. That $\mathtt{A}$ *entails* $\mathtt{B}$ in $\mathcal{M}$ means that any state satisfying $\mathtt{A}$ also satisfies $\mathtt{B}$. Whenever $\mathtt{A}$ is true so is $\mathtt{B}$. The following Venn diagram illustrates $\mathtt{A}$ entails $\mathtt{B}$ in $\mathcal{M}$:



We'll say a proposition is *valid* iff it is valid in all models. For a proposition $\mathtt{A}$, it's traditional to write

$$\models \mathtt{A}$$

to mean $\mathtt{A}$ is valid. (In propositional logic we are interested in when a proposition is always true in all models, and this is caught by validity.)

Let $\mathtt{A}$ and $\mathtt{B}$ be propositions. We'll say that $\mathtt{A}$ entails $\mathtt{B}$, written

$$\mathtt{A} \models \mathtt{B}\ ,$$

[2] See *e.g.* the Part II CS courses 'Specification and Verification I, II.'

iff $\mathtt{A}$ entails $\mathtt{B}$ in all models. We will say $\mathtt{A}$ and $\mathtt{B}$ are equivalent, and write

$$\mathtt{A} \equiv \mathtt{B} \ ,$$

when $\mathtt{A} \models \mathtt{B}$ and $\mathtt{B} \models \mathtt{A}$; in this case, $[\![\mathtt{A}]\!]_{\mathcal{M}} = [\![\mathtt{B}]\!]_{\mathcal{M}}$ in all models $\mathcal{M}$.

Recall a proposition $\mathtt{A} \Rightarrow \mathtt{B}$ is an abbreviation for $\neg\mathtt{A} \vee \mathtt{B}$. Whereas an implication is a proposition, entailment is a relation between propositions. There is a subtle connection between entailment and implication:

$$\mathtt{A} \text{ entails } \mathtt{B} \text{ in a model } \mathcal{M} \text{ iff } \mathtt{A} \Rightarrow \mathtt{B} \text{ is valid in } \mathcal{M}.$$

This is because

$$[\![\mathtt{A}]\!]_{\mathcal{M}} \subseteq [\![\mathtt{B}]\!]_{\mathcal{M}} \iff [\![\mathtt{A}]\!]_{\mathcal{M}}{}^{c} \cup [\![\mathtt{B}]\!]_{\mathcal{M}} = U_{\mathcal{M}} \ ,$$

by simple set theory (see Proposition 2.7, or draw a Venn diagram). The left-hand-side expresses that $\mathtt{A}$ entails $\mathtt{B}$ in the model. The right-hand-side expresses that $\mathtt{A} \Rightarrow \mathtt{B}$ is valid in $\mathcal{M}$, *i.e.* $[\![\mathtt{A} \Rightarrow \mathtt{B}]\!]_{\mathcal{M}} = U_{\mathcal{M}}$. It follows that:

Proposition 2.11 *For Boolean propositions* $\mathtt{A}$, $\mathtt{B}$,

$$\mathtt{A} \models \mathtt{B} \quad \textit{iff} \quad \models (\mathtt{A} \Rightarrow \mathtt{B}) \ .$$

An analogous result holds for equivalence:

Corollary 2.12 *For Boolean propositions* $\mathtt{A}$, $\mathtt{B}$,

$$\mathtt{A} \equiv \mathtt{B} \quad \textit{iff} \quad \models (\mathtt{A} \Leftrightarrow \mathtt{B}) \ .$$

Proof.

$$\begin{aligned} \mathtt{A} \equiv \mathtt{B} \text{ iff } & \mathtt{A} \models \mathtt{B} \text{ and } \mathtt{B} \models \mathtt{A} \\ \text{iff } & \models (\mathtt{A} \Rightarrow \mathtt{B}) \text{ and } \models (\mathtt{B} \Rightarrow \mathtt{A}), \text{by Proposition 2.11,} \\ \text{iff } & \models (\mathtt{A} \Rightarrow \mathtt{B}) \wedge (\mathtt{B} \Rightarrow \mathtt{A}) \quad \text{(Why?), } \textit{i.e.} \models (\mathtt{A} \Leftrightarrow \mathtt{B}) \ . \end{aligned}$$

To answer 'Why?' above, notice that generally, for propositions C and D,

$$\begin{aligned} (\models C \text{ and } \models D) \text{ iff } & [\![\mathtt{C}]\!]_{\mathcal{M}} = U_{\mathcal{M}} \text{ and } [\![\mathtt{D}]\!]_{\mathcal{M}} = U_{\mathcal{M}} \ , \text{ for all models } \mathcal{M}, \\ \text{iff } & [\![\mathtt{C} \wedge \mathtt{D}]\!]_{\mathcal{M}} = U_{\mathcal{M}} \ , \text{ for all models } \mathcal{M}, \\ \text{iff } & \models C \wedge D \ . \end{aligned}$$

□

2.3.3 Truth assignments

It's probably not hard to convince yourself that whether a state in a model satisfies a proposition is determined solely by whether the propositional variables are true or false there (and in fact we'll prove this soon). This suggests a model based on states consisting purely of truth assignments, and leads us to review the method of truth tables.

It is intended that a *truth assignment* should associate a unique truth value $\mathtt{T}$ (for true) and $\mathtt{F}$ (for false) to each propositional variable $\mathtt{a}, \mathtt{b}, \mathtt{c}, \cdots \in \mathtt{Var}$. So we define a truth assignment to be a set of propositional variables tagged by their associated truth values. An example of a truth assignment is the set

$$\{\mathtt{aT}, \mathtt{bF}, \mathtt{cT}, \cdots\} \ ,$$

where we tag $\mathtt{a}$ by $\mathtt{T}$ to show it is assigned true and $\mathtt{b}$ by $\mathtt{F}$ to show that it is assigned false—we cannot have *e.g.* both $\mathtt{aT}$ and $\mathtt{aF}$ in the truth assignment because the truth value assigned to a propositional variable has to be unique.[3]

[3]Later you'll see that in effect a truth assignment is a set-theoretic *function* from $\mathtt{Var}$ to the set $\{\mathtt{T}, \mathtt{F}\}$, but we don't have that technology to hand just yet.

Let $U_{\mathcal{TA}}$ be the set consisting of all truth assignments (an example of a set of sets!). We build a model $\mathcal{TA}$ with universe $U_{\mathcal{TA}}$ by defining:

$$\begin{aligned}
[\![\mathtt{a}]\!]_{\mathcal{TA}} &= \{t \in U_{\mathcal{TA}} \mid \mathtt{aT} \in t\} \\
[\![\mathtt{T}]\!]_{\mathcal{TA}} &= U_{\mathcal{TA}} \\
[\![\mathtt{F}]\!]_{\mathcal{TA}} &= \emptyset \\
[\![\mathtt{A} \wedge \mathtt{B}]\!]_{\mathcal{TA}} &= [\![\mathtt{A}]\!]_{\mathcal{TA}} \cap [\![\mathtt{B}]\!]_{\mathcal{TA}} \\
[\![\mathtt{A} \vee \mathtt{B}]\!]_{\mathcal{TA}} &= [\![\mathtt{A}]\!]_{\mathcal{TA}} \cup [\![\mathtt{B}]\!]_{\mathcal{TA}} \\
[\![\neg\mathtt{A}]\!]_{\mathcal{TA}} &= [\![\mathtt{A}]\!]_{\mathcal{TA}}{}^{c} \ .
\end{aligned}$$

The idea is that $[\![\mathtt{A}]\!]_{\mathcal{TA}}$ is the set of truth assignments which make A true, which is the reason why the clause for propositional variables takes the form it does. The definition above is an example of a *definition by structural induction*. We define an operation, that of $[\![\mathtt{A}]\!]_{\mathcal{TA}}$ on propositions A, by

- first defining the operation on the atomic expressions, specifying $[\![\mathtt{a}]\!]_{\mathcal{TA}}$, for propositional variables a, and $[\![\mathtt{T}]\!]_{\mathcal{TA}}$ and $[\![\mathtt{F}]\!]_{\mathcal{TA}}$, and
- then specifying the operation on a compound expression in terms of the operation on immediate subexpressions, *e.g.* $[\![\mathtt{A} \vee \mathtt{B}]\!]_{\mathcal{TA}}$ in terms of $[\![\mathtt{A}]\!]_{\mathcal{TA}}$ and $[\![\mathtt{B}]\!]_{\mathcal{TA}}$.

The model $\mathcal{TA}$ based on truth assignments has a privileged position amongst models. A proposition A is valid in all models iff A is valid in the particular model $\mathcal{TA}$; a proposition A entails a proposition B iff A entails B in the particular model $\mathcal{TA}$. This will follow from the next lemma. Its proof is an example of *proof by structural induction*. Notice the pattern. We prove a property (the induction hypothesis, IH) holds of all propositions by showing

- IH holds of all atomic expressions (propositional variables, T, F), and
- that IH holds of compound expressions (for instance $\mathtt{A} \wedge \mathtt{B}$) follows from IH holding of immediate subexpressions (in this instance A and B).

Lemma 2.13 *Let* A *be a proposition. Then,* $\models \mathtt{A}$*, i.e.* A *is valid in all models, iff* A *is valid in the model* $\mathcal{TA}$ *of truth assignments.*

Proof.
"only if": obvious.
"if": Let $\mathcal{M}$ be a model of propositional logic with set of states $U_{\mathcal{M}}$. For $u \in U_{\mathcal{M}}$, define the truth assignment $t(u)$ by

$$t(u) = \{\mathtt{aT} \mid \mathtt{a} \in \mathtt{Var} \ \&\ \mathtt{u} \in [\![\mathtt{a}]\!]_{\mathcal{M}}\} \cup \{\mathtt{aF} \mid \mathtt{a} \in \mathtt{Var} \ \&\ \mathtt{u} \notin [\![\mathtt{a}]\!]_{\mathcal{M}}\} \ ,$$

which assigns T to propositional variable a if $u \in [\![\mathtt{a}]\!]_{\mathcal{M}}$, and assigns F otherwise. We show, by structural induction on propositions A, that

$$\forall u.\ u \in [\![\mathtt{A}]\!]_{\mathcal{M}} \text{ iff } t(u) \in [\![\mathtt{A}]\!]_{\mathcal{TA}} \ , \tag{IH}$$

for all propositions A. (The statement IH is the induction hypothesis.) The proof splits into cases according to the syntactic form of A. (We use $=$ for the relation of syntactic identity.)

$\mathtt{A} = \mathtt{a}$, a propositional variable. By the definition of $t(u)$,

$$\begin{aligned}
u \in [\![\mathtt{a}]\!]_{\mathcal{M}} &\iff \mathtt{aT} \in t(u) \\
&\iff t(u) \in [\![\mathtt{a}]\!]_{\mathcal{TA}} \ .
\end{aligned}$$

Hence the induction hypothesis IH holds for propositional variables.

$\mathtt{A} = \mathtt{T}$: In this case IH holds because both $u \in [\![\mathtt{T}]\!]_{\mathcal{M}}$ and $t(u) \in [\![\mathtt{T}]\!]_{\mathcal{TA}}$ for all $u \in U_{\mathcal{M}}$.

$\mathtt{A} = \mathtt{F}$: In this case IH holds because both $u \notin [\![\mathtt{T}]\!]_{\mathcal{M}}$ and $t(u) \notin [\![\mathtt{T}]\!]_{\mathcal{TA}}$ for all $u \in U_{\mathcal{M}}$.

$\mathtt{A} = \mathtt{B} \wedge \mathtt{C}$. In this case the task is to show that IH holds for $\mathtt{B}$ and for $\mathtt{C}$ implies IH holds for $\mathtt{B} \wedge \mathtt{C}$. Assume that IH holds for $\mathtt{B}$ and $\mathtt{C}$. Then,

$$\begin{aligned} u \in [\![\mathtt{B} \wedge \mathtt{C}]\!]_{\mathcal{M}} &\iff u \in [\![\mathtt{B}]\!]_{\mathcal{M}} \text{ and } u \in [\![\mathtt{C}]\!]_{\mathcal{M}}, \text{ as } \mathcal{M} \text{ is a model,} \\ &\iff t(u) \in [\![\mathtt{B}]\!]_{\mathcal{TA}} \text{ and } t(u) \in [\![\mathtt{C}]\!]_{\mathcal{TA}}, \text{ by the induction hypothesis,} \\ &\iff t(u) \in [\![\mathtt{B} \wedge \mathtt{C}]\!]_{\mathcal{TA}}, \text{ as } \mathcal{TA} \text{ is a model.} \end{aligned}$$

$\mathtt{A} = \mathtt{B} \vee \mathtt{C}$. Similarly we argue

$$\begin{aligned} u \in [\![\mathtt{B} \vee \mathtt{C}]\!]_{\mathcal{M}} &\iff u \in [\![\mathtt{B}]\!]_{\mathcal{M}} \text{ or } u \in [\![\mathtt{C}]\!]_{\mathcal{M}}, \text{ as } \mathcal{M} \text{ is a model,} \\ &\iff t(u) \in [\![\mathtt{B}]\!]_{\mathcal{TA}} \text{ or } t(u) \in [\![\mathtt{C}]\!]_{\mathcal{TA}}, \text{ by the induction hypothesis,} \\ &\iff t(u) \in [\![\mathtt{B} \vee \mathtt{C}]\!]_{\mathcal{TA}}, \text{ as } \mathcal{TA} \text{ is a model.} \end{aligned}$$

$\mathtt{A} = \neg\mathtt{B}$. We argue

$$\begin{aligned} u \in [\![\neg\mathtt{B}]\!]_{\mathcal{M}} &\iff u \notin [\![\mathtt{B}]\!]_{\mathcal{M}}, \text{ as } \mathcal{M} \text{ is a model,} \\ &\iff t(u) \notin [\![\mathtt{B}]\!]_{\mathcal{TA}}, \text{ by the induction hypothesis.} \end{aligned}$$

By structural induction we conclude that the induction hypothesis holds for all propositions $\mathtt{A}$. We deduce that if $[\![\mathtt{A}]\!]_{\mathcal{TA}} = U_{\mathcal{TA}}$, the set of all truth assignments, then $[\![\mathtt{A}]\!]_{\mathcal{M}} = U_{\mathcal{M}}$ for any model $\mathcal{M}$, and hence $\mathtt{A}$ is valid. □

Corollary 2.14 *For all propositions* $\mathtt{A}$ *and* $\mathtt{B}$*,*

$$\mathtt{A} \models \mathtt{B} \textit{ iff } [\![\mathtt{A}]\!]_{\mathcal{TA}} \subseteq [\![\mathtt{B}]\!]_{\mathcal{TA}} \ .$$

Proof. *"only if"*: Suppose $\mathtt{A}$ entails $\mathtt{B}$, Then $\mathtt{A}$ entails $\mathtt{B}$ in any model, and so in particular in the model $\mathcal{TA}$, *i.e.* $[\![\mathtt{A}]\!]_{\mathcal{TA}} \subseteq [\![\mathtt{B}]\!]_{\mathcal{TA}}$. *"if"*: Suppose $[\![\mathtt{A}]\!]_{\mathcal{TA}} \subseteq [\![\mathtt{B}]\!]_{\mathcal{TA}}$. Then

$$[\![\mathtt{A} \Rightarrow \mathtt{B}]\!]_{\mathcal{TA}} \ = \ [\![\mathtt{A}]\!]_{\mathcal{TA}}{}^{c} \cup [\![\mathtt{B}]\!]_{\mathcal{TA}} \ = \ U_{\mathcal{TA}} \ . \qquad \text{(Why?)}$$

This means $\mathtt{A} \Rightarrow \mathtt{B}$ is valid in $\mathcal{TA}$. Therefore $\models \mathtt{A} \Rightarrow \mathtt{B}$ by Lemma 2.13. Hence, $\mathtt{A} \models \mathtt{B}$ by Proposition 2.11. □

2.3.4 Truth tables

Lemma 2.13 explains the widespread applicability of a calculational method that you already know. A way to check a proposition is valid, so true in any conceivable model in any conceivable state, is via the well-known method of truth tables.

A truth table explains the truth value assigned to a compound proposition (such as $\mathtt{A} \vee \mathtt{B}$) in terms of the truth values assigned to its constituent propositions ($\mathtt{A}$ and $\mathtt{B}$). This table explains the basic logical connectives:

A	B	¬A	A ∧ B	A ∨ B	A ⇒ B	A ⇔ B
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

Remark A slight digression. Sometimes students balk at the truth table for implication. An implication $\mathtt{A} \Rightarrow \mathtt{B}$ is true whenever $\mathtt{A}$ is false. This can be at variance with common usage, as often in everyday speech when we say that $\mathtt{A}$ implies $\mathtt{B}$ (or more usually, if $\mathtt{A}$ then $\mathtt{B}$) we mean that $\mathtt{A}$ has some relevance for, or causal influence on, $\mathtt{B}$. This everyday usage is clearly not that caught by the understanding of implication illustrated in the truth table. According to the truth table

`The moon is made of cheese implies I'm a professor`

is true regardless of whether or not I'm a professor. The implication we use, that of the truth table, is sometimes called *material* implication to distinguish it from the more sophisticated usages in natural language. □

One builds truth tables for a more complicated proposition out of truth tables for its subpropositions in a column-by-column manner. For example, the truth table for $(\mathtt{a} \wedge \mathtt{b}) \vee \neg\mathtt{a}$, built out of propositional variables $\mathtt{a}$ and $\mathtt{b}$, takes the form:[4]

$\mathtt{a}$	$\mathtt{b}$	$\neg\mathtt{a}$	$\mathtt{a} \wedge \mathtt{b}$	$(\mathtt{a} \wedge \mathtt{b}) \vee \neg\mathtt{a}$
F	F	T	F	T
F	T	T	F	T
T	F	F	F	F
T	T	F	T	T

If $\mathtt{c}$ were another propositional variable we could expand the truth table with truth assignments to $\mathtt{c}$ without affecting the resulting truth value for $(\mathtt{a}\wedge\mathtt{b})\vee\neg\mathtt{a}$. In this way one can expand a truth table for any proposition so that each row of the table corresponds to a truth assignment to all the propositional variables. The truth table built for a proposition $\mathtt{A}$ specifies those truth assignments (its rows) that result in the proposition $\mathtt{A}$ being true. This is just another way to describe $[\![\mathtt{A}]\!]_{\mathcal{TA}}$, the set of truth assignments that make $\mathtt{A}$ true. The proposition $\mathtt{A}$ is valid iff it is true for all truth assignments. We see this in its truth table through $\mathtt{A}$ being assigned $\mathtt{T}$ in every row. Such a proposition is traditionally called a *tautology*.

Proposition 2.15 *A proposition* $\mathtt{A}$ *is valid iff it is a tautology.*

Proof. We show by structural induction on $\mathtt{A}$ that for all propositions $\mathtt{A}$,

$$\forall t \in U_{\mathcal{TA}}.\ t \in [\![\mathtt{A}]\!]_{\mathcal{TA}} \iff \text{the truth table at row } t \text{ gives } \mathtt{T} \text{ for } \mathtt{A}\ , \tag{IH}$$

taking (IH) to be the induction hypothesis. Once this is shown, it follows that $\mathtt{A}$ is valid (*i.e.* $t \in [\![\mathtt{A}]\!]_{\mathcal{TA}}$ for all truth assignments t by Lemma 2.13) iff $\mathtt{A}$ is a tautology (*i.e.* the truth table gives $\mathtt{T}$ for $\mathtt{A}$ at all rows t).

To carry out the proof by structural induction we need to show: for any proposition $\mathtt{A}$, if IH holds for the immediate subpropositions of $\mathtt{A}$, then IH holds for $\mathtt{A}$. The proof falls into cases according to the form of $\mathtt{A}$.

$\mathtt{A} = \mathtt{a}$, a propositional variable. In this case,

$$\begin{aligned} t \in [\![\mathtt{a}]\!]_{\mathcal{TA}} &\iff \mathtt{aT} \in t \\ &\iff \text{the truth table at } t \text{ gives } \mathtt{T} \text{ for } \mathtt{a}. \end{aligned}$$

$\mathtt{A} = \mathtt{B} \wedge \mathtt{C}$.

$$\begin{aligned} t \in [\![\mathtt{B} \wedge \mathtt{C}]\!]_{\mathcal{TA}} &\iff t \in [\![\mathtt{B}]\!]_{\mathcal{TA}} \text{ and } t \in [\![\mathtt{C}]\!]_{\mathcal{TA}} \\ &\iff \text{the truth table at } t \text{ gives } \mathtt{T} \text{ for } \mathtt{B} \text{ and } \mathtt{T} \text{ for } \mathtt{C}, \text{ by IH}, \\ &\iff \text{the truth table at } t \text{ gives } \mathtt{T} \text{ for } \mathtt{B} \wedge \mathtt{C}. \end{aligned}$$

$\mathtt{A} = \mathtt{B} \vee \mathtt{C}$.

$$\begin{aligned} t \in [\![\mathtt{B} \vee \mathtt{C}]\!]_{\mathcal{TA}} &\iff t \in [\![\mathtt{B}]\!]_{\mathcal{TA}} \text{ or } t \in [\![\mathtt{C}]\!]_{\mathcal{TA}} \\ &\iff \text{the truth table at } t \text{ gives } \mathtt{T} \text{ for } \mathtt{B} \text{ or for } \mathtt{C}, \text{ by IH}, \\ &\iff \text{the truth table at } t \text{ gives } \mathtt{T} \text{ for } \mathtt{B} \vee \mathtt{C}. \end{aligned}$$

$\mathtt{A} = \neg\mathtt{B}$.

$$\begin{aligned} t \in [\![\neg\mathtt{B}]\!]_{\mathcal{TA}} &\iff t \notin [\![\mathtt{B}]\!]_{\mathcal{TA}} \\ &\iff \text{the truth table at } t \text{ does not give } \mathtt{T} \text{ for } \mathtt{B}, \text{ by IH}, \\ &\iff \text{the truth table at } t \text{ gives } \mathtt{F} \text{ for } \mathtt{B} \\ &\iff \text{the truth table at } t \text{ gives } \mathtt{T} \text{ for } \neg\mathtt{B}. \end{aligned}$$

□

Proposition 2.15 links facts about sets (validity in any model) to facts about the evaluation of propositions to truth values (truth tables). From "if", whenever we interpret a tautology in a model it will denote the universe of the model. From "only if", any proposition which always denotes the universe in any model has to be a tautology.

[4] Truth tables can get very big. The CS Part IB course 'Logic and Proof' presents more efficient ways to evaluate the truth value of propositions, methods which take more careful account of the possible sharing of subpropositions, and can exploit the order in which subpropositions are evaluated to truth values.

2.3.5 Methods

We can use truth tables to show an entailment $\mathtt{A} \models \mathtt{B}$, or an equivalence $\mathtt{A} \equiv \mathtt{B}$. Recall Proposition 2.11, that

$$\mathtt{A} \models \mathtt{B} \text{ iff } \models \mathtt{A} \Rightarrow \mathtt{B} \ .$$

So, by Proposition 2.15, one way to show $\mathtt{A} \models \mathtt{B}$ is to show that $(\mathtt{A} \Rightarrow \mathtt{B})$ is a tautology. But this amounts to showing that in any row (so truth assignment) where $\mathtt{A}$ gives $\mathtt{T}$ so does $\mathtt{B}$—$\mathtt{B}$ may give $\mathtt{T}$ on more rows than $\mathtt{A}$. Conversely, if $\mathtt{A} \models \mathtt{B}$, then any truth assignment making $\mathtt{A}$ true will make $\mathtt{B}$ true—a fact which transfers to their truth tables. For example, you can easily check that the truth tables for $\mathtt{A} \vee \mathtt{B}$ and $\neg(\neg\mathtt{A} \wedge \neg\mathtt{B})$ are the same; hence $\mathtt{A} \vee \mathtt{B} \equiv \neg(\neg\mathtt{A} \wedge \neg\mathtt{B})$. (In fact, we could have been even more parsimonious in the syntax of propositions, and taken $\mathtt{A} \vee \mathtt{B}$ to be an abbreviation for $\neg(\neg\mathtt{A} \wedge \neg\mathtt{B})$.)

Truth tables are one way to establish the equivalence $\mathtt{A} \equiv \mathtt{B}$ of propositions $\mathtt{A}$ and $\mathtt{B}$: check that the truth tables for $\mathtt{A}$ and $\mathtt{B}$ yield the same truth values on corresponding rows. But propositions stand for sets in any model so we can also use the identities of Boolean algebra to simplify propositions, treating conjunctions as intersections, disjunctions as unions and negations as complements. For example, from the De Morgan and Complement laws

$$\begin{aligned} \neg(\mathtt{a} \wedge \neg\mathtt{b}) &\equiv \neg\mathtt{a} \vee \neg\neg\mathtt{b} \\ &\equiv \neg\mathtt{a} \vee \mathtt{b} \ . \end{aligned}$$

As here we can make use of the fact that equivalence is substitutive in the following sense. Once we know two propositions $\mathtt{B}$ and $\mathtt{B}'$ are equivalent, if we have another proposition $\mathtt{C}$ in which $\mathtt{B}$ occurs we can replace some or all of its occurrences by $\mathtt{B}'$ and obtain an equivalent proposition $\mathtt{C}'$. One way to see this is by considering the truth table of $\mathtt{C}$—the eventual truth value obtained will be unaffected if $\mathtt{B}'$ stands in place of $\mathtt{B}$, provided $\mathtt{B} \equiv \mathtt{B}'$. This means that we can handle equivalence just as the equality of school algebra. (Exercise 2.25 guides you through a proof of the property of substitutivity of equivalence.)

Generally, using the set identities any proposition can be transformed to disjunctive form as a disjunction of conjunctions of propositional variables and their negations, or alternatively to conjunctive form as a conjunction of disjunctions of propositional variables and their negations, *e.g.*:

$$\begin{aligned} &\cdots \vee (\neg\mathtt{a}_1 \wedge \mathtt{a}_2 \wedge \cdots \wedge \mathtt{a}_k) \vee \cdots \\ &\cdots \wedge (\neg\mathtt{a}_1 \vee \mathtt{a}_2 \vee \cdots \vee \mathtt{a}_k) \wedge \cdots \end{aligned}$$

With the help of the Idempotence and Complement laws we can remove redundant occurrences of propositional variables to obtain normal forms (unique up to reordering), respectively *disjunctive* and *conjunctive normal forms* for propositions. The equivalence of two propositions can be checked by comparing their normal forms. The normal forms play a central role in theorem proving.

Exercise 2.16 Using the set laws express $\neg(\mathtt{a} \vee \mathtt{b}) \vee (\mathtt{a} \wedge \mathtt{c})$ in conjunctive form. □

Exercise 2.17 Do this exercise without using Proposition 2.15.
(i) Using the method of truth tables show $(\neg\mathtt{B} \Rightarrow \neg\mathtt{A}) \equiv (\mathtt{A} \Rightarrow \mathtt{B})$. Deduce

$$(\neg\mathtt{B} \Rightarrow \neg\mathtt{A}) \Leftrightarrow (\mathtt{A} \Rightarrow \mathtt{B})$$

is a tautology.
(ii) Show in any model $\mathcal{M}$ that

$$[\![\neg\mathtt{B} \Rightarrow \neg\mathtt{A}]\!]_{\mathcal{M}} = [\![\mathtt{A} \Rightarrow \mathtt{B}]\!]_{\mathcal{M}} \ ;$$

deduce

$$\models [(\neg\mathtt{B} \Rightarrow \neg\mathtt{A}) \Leftrightarrow (\mathtt{A} \Rightarrow \mathtt{B})] \ .$$

Parts (i) and (ii) give two methods for demonstrating entailments and tautologies linked by Proposition 2.15 . Method (ii) might look long-winded. However in practice one can drop the $[\![-]\!]_{\mathcal{M}}$ brackets, think of propositions as sets and use Venn diagrams or the set identities to simplify set expressions (just as we did above in simplifying $\neg(\mathtt{a} \wedge \neg\mathtt{b})$). □

Exercise 2.18

(i) Show that $\mathtt{A} \Leftrightarrow \mathtt{B} \equiv (\mathtt{A} \wedge \mathtt{B}) \vee (\neg\mathtt{A} \wedge \neg\mathtt{B})$.

(ii) Show that $A \Leftrightarrow (B \Leftrightarrow C) \equiv (A \Leftrightarrow B) \Leftrightarrow C$.
[The analogous result does not hold when $\Leftrightarrow$ is replaced by $\Rightarrow$—why not?]

(iii) Show that $\neg(B \Leftrightarrow C) \equiv ((\neg B) \Leftrightarrow C)$.

□

Exercise 2.19 Sheffer's stroke is not an affliction but a logical operation $\mathtt{A|B}$ out of which all the usual logical operations can be derived. It is defined by the following truth table:

A	B	A|B
F	F	T
F	T	T
T	F	T
T	T	F

Check that $\mathtt{A|B} \equiv \neg(\mathtt{A} \wedge \mathtt{B})$ by showing that they have the same truth table. Describe how to define the operations of negation, conjunction and disjunction out of Sheffer's stroke. □

Exercise 2.20 Verify that *Peirce's law*, $((\mathtt{A} \Rightarrow \mathtt{B}) \Rightarrow \mathtt{A}) \Rightarrow \mathtt{A}$, is a tautology. □

Exercise 2.21 Simplify the Boolean proposition

$$\neg(\neg(\mathtt{a} \wedge \neg(\mathtt{a} \wedge \mathtt{b})) \wedge \neg(\neg(\mathtt{a} \wedge \mathtt{b}) \wedge \mathtt{b})) \ .$$

□

Exercise 2.22 Simplify $[(\mathtt{a} \Rightarrow \mathtt{b}) \vee (\mathtt{a} \Rightarrow \mathtt{d})] \Rightarrow (\mathtt{b} \vee \mathtt{d})$ to the proposition $\mathtt{a} \vee \mathtt{b} \vee \mathtt{d}$. □

Exercise 2.23 Consider the argument: "If Anna can cancan or Kant can't cant, then Greville will cavil vilely. If Greville will cavil vilely, Will will want. But Will won't want. Therefore Kant can cant." By writing the statement in quotes as a proposition in terms of four propositional variables and simplifying, show that it is a tautology and hence that the argument holds. □

Exercise 2.24 Define the length of a Boolean proposition by structural induction as follows:

$$\begin{aligned}
&|\mathtt{a}| = 1, \quad |\mathtt{T}| = 1, \quad |\mathtt{F}| = 1, \\
&|\mathtt{A} \wedge \mathtt{B}| = |\mathtt{A}| + |\mathtt{B}| + 1, \\
&|\mathtt{A} \vee \mathtt{B}| = |\mathtt{A}| + |\mathtt{B}| + 1, \quad |\neg\mathtt{A}| = |\mathtt{A}| + 1 \ .
\end{aligned}$$

Define a translation which eliminates disjunction from Boolean expressions by the following structural induction:

$$\begin{aligned}
&tr(\mathtt{a}) = \mathtt{a}, \quad tr(\mathtt{T}) = \mathtt{T}, \quad tr(\mathtt{F}) = \mathtt{F}, \\
&tr(\mathtt{A} \wedge \mathtt{B}) = tr(\mathtt{A}) \wedge tr(\mathtt{B}), \\
&tr(\mathtt{A} \vee \mathtt{B}) = \neg(\neg tr(\mathtt{A}) \wedge \neg tr(\mathtt{B})), \quad tr(\neg\mathtt{A}) = \neg tr(\mathtt{A}) \ .
\end{aligned}$$

Prove by structural induction on Boolean propositions that

$$|tr(\mathtt{A})| \leq 3|\mathtt{A}| - 1 \ ,$$

for all Boolean propositions $\mathtt{A}$. □

Exercise 2.25 Define a Boolean propositional context to be given by

$$\mathtt{C}, \mathtt{C}', \ldots ::= \mathtt{a}, \mathtt{b}, \mathtt{c}, \cdots \ | \ \mathtt{T} \ | \ \mathtt{F} \ | \ [\_] \ | \ \mathtt{C} \wedge \mathtt{C}' \ | \ \mathtt{C} \vee \mathtt{C}' \ | \ \neg\mathtt{C}$$

where $\mathtt{a}, \mathtt{b}, \mathtt{c} \cdots \in \mathtt{Var}$. So a context is like a Boolean proposition but possibly with several occurrences of a 'hole' $[\_]$, into which a Boolean proposition can be substituted. Define the substitution $\mathtt{C}[\mathtt{B}]$ of a proposition $\mathtt{B}$ into a context $\mathtt{C}$ by structural induction on contexts as follows:

$$\begin{aligned}
&\mathtt{a}[\mathtt{B}] = \mathtt{a}, \quad \mathtt{T}[\mathtt{B}] = \mathtt{T}, \quad \mathtt{F}[\mathtt{B}] = \mathtt{F}, \quad [_][\mathtt{B}] = \mathtt{B}, \\
&(\mathtt{C} \wedge \mathtt{C}')[\mathtt{B}] = \mathtt{C}[\mathtt{B}] \wedge \mathtt{C}'[\mathtt{B}], \quad (\mathtt{C} \vee \mathtt{C}')[\mathtt{B}] = \mathtt{C}[\mathtt{B}] \vee \mathtt{C}'[\mathtt{B}], \\
&\neg\mathtt{C}[\mathtt{B}] = \neg(\mathtt{C}[\mathtt{B}]) \ .
\end{aligned}$$

Prove by structural induction on contexts that, for all contexts $\mathtt{C}$,

if $\mathtt{B}$ and $\mathtt{B}'$ are propositions for which $\mathtt{B} \equiv \mathtt{B}'$, then $\mathtt{C}[\mathtt{B}] \equiv \mathtt{C}[\mathtt{B}']$.

□

Chapter 3

Relations and functions

In this chapter we study how to relate, possibly different, sets through the set-theoretic definitions of relation and function. We will rely on the product of sets as the central construction for connecting sets. We are led to consider sets with extra structure, and the cardinality of sets, in particular the important notion of countability.

3.1 Ordered pairs and products

Given two elements a, b we can form their ordered pair (a, b). Two ordered pairs are equal iff their first components are equal and their second components are equal too, *i.e.*

$$(a, b) = (a', b') \iff a = a' \ \& \ b = b' \ .$$

There is also the concept of *unordered* pair of elements a, b—this is just the set $\{a, b\}$. We'll only rarely use unordered pairs so "pair" alone will mean ordered pair.

Often you'll see the ordered pair (a, b) defined to be the set $\{\{a\}, \{a, b\}\}$—this is one particular way of coding the idea of ordered pair as a *set*. (See Exercise 3.2 below. Apart from this exercise we'll never again consider how ordered pairs are implemented.)

For sets X and Y, their *product* is the set

$$X \times Y = \{(a, b) \mid a \in X \ \& \ b \in Y\},$$

the set of all ordered pairs of elements with the first from X and the second from Y.

We can use the product construction on sets several times. A ternary product of sets $X \times Y \times Z$, consisting of triples (x, y, z), can be understood as $X \times (Y \times Z)$, and so on. In the case where all the sets in a product are the same, as in $X \times X$ we often write the product as X^2, with $X \times X \times X$ written as X^3, and generally a product $X \times \cdots \times X$, the product of n copies of X, as X^n. Such products are familiar from coordinate geometry: a point on a line can be identified with a real number in $\mathbb{R}$, the set of real numbers; the points on the plane can be identified with elements of the product $\mathbb{R} \times \mathbb{R}$, which we can also write as $\mathbb{R}^2$; three-dimensional space with $\mathbb{R}^3$, and so on.

Exercise 3.1 Prove

(i) $A \times (B \cup C) = (A \times B) \cup (A \times C)$

(ii) $A \times (B \cap C) = (A \times B) \cap (A \times C)$

(iii) $(A \times B) \cap (C \times D) = (A \cap C) \times (B \cap D)$

(iv) $(A \times B) \cup (C \times D) \subseteq (A \cup C) \times (B \cup D)$ [Show the converse inclusion does not hold in general.]

□

Exercise 3.2 Show that a set $\{\{a\}, \{a, b\}\}$ behaves as an ordered pair should, *i.e.*

$$\{\{a\}, \{a, b\}\} = \{\{a'\}, \{a', b'\}\} \iff a = a' \ \& \ b = b' \ .$$

[This is trickier than you might at first think. Consider the two cases $a = b$ and $a \neq b$.] □

3.2 Relations and functions

A *binary relation* between X and Y is a subset $R \subseteq X \times Y$—so a subset of pairs in the relation. We shall often write xRy for $(x, y) \in R$.

Let $R \subseteq X \times Y$. Write R^{-1} for the *converse*, or *inverse*, relation $R^{-1} = \{(y, x) \mid (x, y) \in R\}$; so $R^{-1} \subseteq Y \times X$ with $yR^{-1}x$ iff xRy.

A *partial function* from X to Y is a relation $f \subseteq X \times Y$ for which

$$\forall x, y, y'.\ (x, y) \in f\ \&\ (x, y') \in f \Rightarrow y = y'.$$

We use the notation $f(x) = y$ when there is a y such that $(x, y) \in f$ and then say $f(x)$ is *defined*, and otherwise say $f(x)$ is *undefined*; the set

$$\{x \in X \mid f(x) \text{ is defined}\}$$

is called the *domain of definition* of the partial function f. Sometimes we write $f : x \mapsto y$, or just $x \mapsto y$ when f is understood, for $y = f(x)$. Occasionally one sees just fx, without the brackets, for $f(x)$.

A *(total) function* from X to Y is a partial function from X to Y such that for all $x \in X$ there is some $y \in Y$ such that $f(x) = y$. Although total functions are a special kind of partial function it is traditional to understand something described as simply a function to be a total function, so we always say explicitly when a function is partial.

To stress the fact that we are thinking of a function f from X to Y as taking an element of X and yielding an element of Y we generally write it as $f : X \to Y$. To indicate a partial function f from X to Y we write $f : X \rightharpoonup Y$. For both functions and partial functions from X to Y, the set X is called the *domain* of the function and Y the *codomain* of the function.

Note that individual relations and functions are also sets. This fact determines equality between relations, and equality between functions; they are equal iff they consist of the same set of pairs. We can reword this fact in the case of functions and partial functions.

Proposition 3.3

(i) Let $R, R' \subseteq X \times Y$. Then,

$$R = R' \text{ iff } \forall x \in X, y \in Y.\ xRy \iff xR'y\ .$$

(ii) Let $f, f' : X \to Y$. Then,

$$f = f' \text{ iff } \forall x \in X.\ f(x) = f'(x)\ .$$

(iii) Let $f, f' : X \rightharpoonup Y$. Then,

$$\begin{aligned} f = f' \text{ iff } \forall x \in X.\ &(f(x) \text{ is defined } \Rightarrow f'(x) \text{ is defined } \&\ f(x) = f'(x)) \ \&\\ &(f'(x) \text{ is defined } \Rightarrow f(x) \text{ is defined } \&\ f(x) = f'(x))\ . \end{aligned}$$

So, to investigate whether two functions with the same domain and codomain are equal it suffices to show that they give the same results when applied to an arbitrary common argument.

Exercise 3.4 If A has k elements and B has m elements, how many relations are there between A and B? □

Exercise 3.5 Let R and S be relations between A and B. Show that, if $R \subseteq S$, then $R^{-1} \subseteq S^{-1}$. Prove that $(R \cap S)^{-1} = R^{-1} \cap S^{-1}$ and $(R \cup S)^{-1} = R^{-1} \cup S^{-1}$. □

Exercise 3.6 If A and B are finite sets with m and n elements respectively, how many functions and how many partial functions are there from A to B? □

3.2.1 Composing relations and functions

We compose relations, and so partial and total functions, R between X and Y and S between Y and Z by defining their *composition*, a relation between X and Z, by

$$S \circ R =_{def} \{(x,z) \in X \times Z \mid \exists y \in Y.\ (x,y) \in R \ \&\ (y,z) \in S\}\ .$$

Let $R \subseteq X \times Y$, $S \subseteq Y \times Z$ and $T \subseteq Z \times W$. It should not be hard to convince yourself that

$$T \circ (S \circ R) = (T \circ S) \circ R$$

i.e. composition is associative.

Exercise 3.7 Let $A = \{1,2,3,4\}$, $B = \{a,b,c,d\}$ and $C = \{x,y,z\}$. Let $R = \{(1,a),(2,d),(3,a),(3,b),(3,d)\}$ and $S = \{(b,x),(b,z),(c,y),(d,z)\}$. What is their composition $S \circ R$? □

Exercise 3.8 Show that the composition of relations is associative. □

Each set X is associated with an identity relation id_X where $\mathsf{id}_X = \{(x,x) \mid x \in X\}$. It is easy to see that for any relation R between X and Y

$$R \circ \mathsf{id}_X = \mathsf{id}_Y \circ R = R$$

—so the identity relation does indeed behave like an identity with respect to composition. Note that the identity relation is a function.

For functions $f : X \to Y$ and $g : Y \to Z$ their composition is also a function $g \circ f : X \to Z$ (check!). Similarly, the composition of partial functions $f : X \rightharpoonup Y$ and $g : Y \rightharpoonup Z$ is a partial function $g \circ f : X \rightharpoonup Z$ (check!).

We say a function $f : X \to Y$ is *injective* (or 1-1) iff

$$\forall x, x' \in X.\ f(x) = f(x') \Rightarrow x = x'\ .$$

In other words, taking the contrapositive of this implication, distinct elements of X go to distinct elements of Y. An injective function is often called an *injection*.

We say a function $f : X \to Y$ is *surjective* (or onto) iff

$$\forall y \in Y \exists x \in X.\ y = f(x)\ .$$

A surjective function is often called an *surjection*.

A function $f : X \to Y$ is *bijective* iff it is both injective and surjective. Bijective functions $f : X \to Y$ are called *bijections*; the sets X and Y are said to be in *1-1 correspondence*, or *bijective correspondence*.

A function $f : X \to Y$ has an *inverse* function $g : Y \to X$ iff $g(f(x)) = x$ for all $x \in X$, and $f(g(y)) = y$ for all $y \in Y$. Notice the symmetry: f has inverse g means g has inverse f, and *vice versa*.

Lemma 3.9 *A function $f : X \to Y$ is bijective iff it has an inverse function.*

Proof.
"if": Suppose $f : X \to Y$ has an inverse $g : Y \to X$. Let $x, x' \in X$ and suppose $f(x) = f(x')$. Then

$$x = g(f(x)) = g(f(x')) = x'\ .$$

Hence f is injective. Let $y \in Y$. Then $f(g(y)) = y$. Hence f is surjective. It follows that f is bijective.
"only if": Assume $f : X \to Y$ is bijective. Define the relation $g \subseteq Y \times X$ by $g = f^{-1}$, the converse relation of f, so $(y,x) \in g \iff f(x) = y$.

Suppose $(y,x),(y,x') \in g$. Then, $f(x) = y$ and $f(x') = y$, so $x = x'$ as f is injective. Given $y \in Y$ there is $x \in X$ such that $f(x) = y$ as f is surjective, making $(y,x) \in g$. This shows that g is a function $g : Y \to X$ which moreover satisfies

$$g(y) = x \iff f(x) = y\ . \qquad (\dagger)$$

We now deduce that g is injective. Suppose $g(y) = g(y')$, where $y, y' \in Y$. Letting $x = g(y) = g(y')$ we see from (†) that both $f(x) = y$ and $f(x) = y'$, whence $y = y'$.

If $g(f(x)) = x'$ then $f(x') = f(x)$ by (†), so $x = x'$, as f is injective. If $f(g(y)) = y'$ then $g(y') = g(y)$ by (†), so $y = y'$, as g is injective. This shows that g is an inverse to f. □

Suppose $f : X \to Y$ has an inverse $g : Y \to X$. Then g has f as its inverse. So by Lemma 3.9, both f and g are bijective. It is traditional to write f^{-1} for the inverse of a function f.

Exercise 3.10 Show that the composition of injective/surjective/bijective functions is respectively injective/surjective/bijective. □

Exercise 3.11 Let D be the set $\{x \in \mathbb{R} \mid x > 1\}$. Define a binary relation $g \subseteq D \times D$ by taking

$$(u, v) \in g \text{ iff } \frac{1}{u} + \frac{1}{v} = 1 \ .$$

(i) Express v as a formula in u for $(u, v) \in g$. Deduce that g is a function $g : D \to D$.

(ii) Define an inverse function to g and prove that it has the desired properties. Deduce that $g : D \to D$ is a bijection.

[The formula $\frac{1}{u} + \frac{1}{v} = 1$ expresses the relation between the distance of an object u and the distance of its image v from a lens with focal length 1.] □

3.2.2 Direct and inverse image under a relation

We extend relations, and thus partial and total functions, $R \subseteq X \times Y$ to an operation acting on subsets by taking

$$R\ A = \{y \in Y \mid \exists x \in A.\ (x, y) \in R\}$$

for $A \subseteq X$. The set $R\ A$ is called the *direct image* of A under R. We define

$$R^{-1}B = \{x \in X \mid \exists y \in B.\ (x, y) \in R\}$$

for $B \subseteq Y$. The set $R^{-1}B$ is called the *inverse image* of B under R; note that it is the same set as the direct image of the set B under the converse, or inverse, relation R^{-1}. Of course, the same notions of direct and inverse image also apply in the special cases where the relation is a partial function or function.

Exercise 3.12 Suppose $f : X \to Y$ is a function. Show f^{-1} preserves the Boolean operations of union, intersection and complement, *i.e.* for all $B, C \subseteq Y$,

$$\begin{aligned} f^{-1}(B \cup C) &= (f^{-1}B) \cup (f^{-1}C) \ , \qquad & f^{-1}\emptyset &= \emptyset \ , \\ f^{-1}(B \cap C) &= (f^{-1}B) \cap (f^{-1}C) \ , \qquad & f^{-1}Y &= X \ , \\ f^{-1}(B^c) &= (f^{-1}B)^c \ . \end{aligned}$$

What analogous properties hold of the direct image under f? Suppose now $f : X \rightharpoonup Y$ is a partial function. Describe how to modify the above identities to make them hold in this case. Which identities will hold if f is assumed only to be a relation? □

3.3 Relations as structure

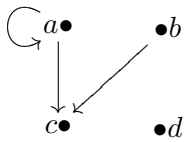
Often in mathematics and computer science we are not so much concerned with bare sets, but rather with sets that possess some extra structure. We consider three important examples, directed graphs, equivalence relations and partial orders. These all arise as a special kind of relation $R \subseteq X \times Y$, where in particular the sets X and Y are the same; then, we often describe R as being a relation *on* the set X.

3.3.1 Directed graphs

One of the simplest examples of sets with structure is that of directed graphs.

Definition: A *directed graph* (or digraph) is a set X on which there is a relation $R \subseteq X \times X$ and so is described by (X, R). The elements of X are called vertices (or nodes) and the elements of R directed edges (or arcs).

Finite directed graphs (*i.e.* those with a finite set of vertices) have a natural diagrammatic representation in which vertices are drawn as nodes and directed edges as arcs between them. Here, for example, is the diagram of the directed graph with vertices $\{a, b, c, d\}$ and directed edges $\{(a, c), (b, c), (a, a)\}$:



Directed graphs are ubiquitous in computer science. They appear both as representations of data-types, expressing relations between objects, and of processes, expressing the possible transitions between states.

3.3.2 Equivalence relations

One often encounters relations that behave like a form of equality or equivalence, captured in the definition of equivalence relation.

An *equivalence relation* is a relation $R \subseteq X \times X$ on a set X which is

- reflexive: $\forall x \in X.\ xRx$,
- symmetric: $\forall x, y \in X.\ xRy \Rightarrow yRx$ and
- transitive: $\forall x, y, z \in X.\ xRy\ \&\ yRz \Rightarrow xRz$.

If R is an equivalence relation on X then the *(R-)equivalence class* of an element $x \in X$ is the subset $\{x\}_R =_{def} \{y \in X \mid yRx\}$.

An equivalence relation on a set X determines a partition of X. A *partition* of a set X is a set P of non-empty subsets of X for which each element x of X belongs to one and only one member of P. In other words, a partition P of X is a collection of nonempty, disjoint subsets of X such that each element x of X belongs to a member of P.

Theorem 3.13 *Let R be an equivalence relation on a set X. The set $X/R =_{def} \{\{x\}_R \mid x \in X\}$ of equivalence classes with respect to R is a partition of the set X. Moreover, $\{x\}_R = \{y\}_R$ iff xRy for all $x, y \in X$.*

Proof. Let $x \in X$. Then as R is reflexive, $x \in \{x\}_R$. So every member of X/R is nonempty and each element of X belongs to a member of X/R. For X/R to be a partition we also require that its members are disjoint. However, we will show

$$\begin{array}{ll} (1) & \{x\}_R \cap \{y\}_R \neq \emptyset \Rightarrow xRy\ , \text{ and} \\ (2) & xRy \Rightarrow \{x\}_R = \{y\}_R\ , \end{array}$$

from which $\{x\}_R \cap \{y\}_R \neq \emptyset \Rightarrow \{x\}_R = \{y\}_R$ follows, for any elements $x, y \in X$.

(1) Suppose $\{x\}_R \cap \{y\}_R \neq \emptyset$. Then there is some $z \in \{x\}_R \cap \{y\}_R$. Hence zRx and zRy. Then xRz and zRy, as R is symmetric. As R is transitive we obtain xRy.

(2) Suppose xRy. Let $w \in \{x\}_R$. Then wRx and xRy, so wRy by transitivity of R. Thus $w \in \{y\}_R$. This shows $\{x\}_R \subseteq \{y\}_R$. Because R is symmetric we have yRx and so by a similar argument we also obtain $\{y\}_R \subseteq \{x\}_R$. Hence $\{x\}_R = \{y\}_R$.

If $\{x\}_R = \{y\}_R$, then certainly $x \in \{x\}_R \cap \{y\}_R$, so xRy by (1). Combining with (2) we see that $\{x\}_R = \{y\}_R$ iff xRy for all $x, y \in X$. □

There is a simple converse to Theorem 3.13:

Proposition 3.14 *Let P be a partition of a set X. The relation R on X, defined by*

$$xRy \iff \exists p \in P.\ x \in p\ \&\ y \in p\ ,$$

is an equivalence relation on the set X with $X/R = P$.

Exercise 3.15 Provide the proof to Proposition 3.14. □

Modular arithmetic

Modular arithmetic is central to number theory in mathematics and very important in computer science, for example in cryptography. It furnishes an example of an equivalence relation.

Let $k \in \mathbb{N}$. There are many situations in which it is useful to regard integers as essentially the same if they give the same remainder when divided by k. Integers a and b give the same remainder when divided by k iff their difference $a - b$ is divisible by k. For integers $a, b \in \mathbb{Z}$ define

$$\begin{aligned} a \equiv b \pmod{k} \text{ iff } \; & a - b \text{ is divisible by } k \\ & \textit{i.e. } (a-b) = n.k \;\; \text{for some } n \in \mathbb{Z}\,. \end{aligned}$$

Then we say "a is congruent to b modulo k."

It is easy to show that congruence modulo k is an equivalence relation on $\mathbb{Z}$, that it is:
Reflexive: $a \equiv a \pmod{k}$ as $(a-a) = 0 = 0.k$
Symmetric: If $a \equiv b \pmod{k}$ then k divides $(a-b)$. Hence k divides $-(a-b) = (b-a)$ giving $b \equiv a \pmod{k}$.
Transitive: Suppose $a \equiv b \pmod{k}$ and $b \equiv c \pmod{k}$. Then k divides both $a-b$ and $b-c$. Hence k divides their sum $(a-b)+(b-c) = (a-c)$. Thus $a \equiv c \pmod{k}$.

It is common to write $[a]$, called a congruence class, for the equivalence class of a w.r.t. congruence modulo k. The congruence classes correspond to the possible remainders on dividing by k, *viz.* 0, 1, ..., $(k-1)$.

Assume $a \equiv b \pmod{k}$ and $a' \equiv b' \pmod{k}$. Then

$$\begin{aligned} & a + a' \equiv b + b' \pmod{k} \qquad a - a' \equiv b - b' \pmod{k} \\ & a \times a' \equiv b \times b' \pmod{k}\,. \end{aligned}$$

Consequently we can define these operations on congruence classes by taking

$$\begin{aligned} & [a] + [a'] = [a + a'] \qquad [a] - [a'] = [a - a'] \\ & [a] \times [a'] = [a \times a']\,. \end{aligned}$$

The operations are well-defined because independently of how we choose representatives for the congruence classes we obtain the same result.

Exercise 3.16 Show that if $a \equiv b \pmod{k}$ and $a' \equiv b' \pmod{k}$, then $a + a' \equiv b + b' \pmod{k}$, $a - a' \equiv b - b' \pmod{k}$ and $a \times a' \equiv b \times b' \pmod{k}$. □

Exercise 3.17 Let $A = \{1, 2, 3\}$. List all the partitions of A. How many equivalence relations are there on A? How many relations are there on A? □

Exercise 3.18 Let $\cong$ be a relation on a set of sets S such that $A \cong B$ iff the sets A and B in S are in bijective correspondence. Show that $\cong$ is an equivalence relation. □

Exercise 3.19 Let R and S be equivalence relations on sets A and B respectively. Let $p : A \to A/R$ and $q : B \to B/S$ be the obvious functions from elements to their equivalence classes. Suppose $f : A \to B$ is a function. Show that the following two statements are equivalent:

(i) $\exists g : A/R \to B/S.\ g \circ p = q \circ f$;

(ii) $\forall a, a' \in A.\ aRa' \Rightarrow f(a) S f(a')$. □

Exercise 3.20 Suppose $(P, \longrightarrow)$ is a directed graph. A *bisimulation* on P is a relation $R \subseteq P \times P$ such that whenever $p\,R\,q$ then

- $\forall p' \in P.\ p \longrightarrow p' \Rightarrow \exists q' \in P.\ q \longrightarrow q'\ \&\ p'\,R\,q'$, and
- $\forall q' \in P.\ q \longrightarrow q' \Rightarrow \exists p' \in P.\ p \longrightarrow p'\ \&\ p'\,R\,q'$.

Define the *bisimilarity* relation $\sim$ on P by taking $p \sim q$ iff $p\,R\,q$, for some bisimulation R on P.

Show the following:

(i) the identity relation id_P is a bisimulation on P;

(ii) if R is a bisimulation on P then its converse relation R^{-1} is a bisimulation on P;

(iii) if relations R and S are bisimulations on P, then their composition $S \circ R$ is a bisimulation on P.

Deduce that the bisimilarity relation $\sim$ is an equivalence relation on P. Show that $\sim$ is itself a bisimulation on P. □

3.3.3 Partial orders

A very important example of sets with structure is that of sets equipped with a relation ordering elements (perhaps with respect to some characteristic like size).

Definition: A *partial order* (p.o.) is a set P on which there is a binary relation $\leq$, so described by $(P, \leq)$, which is:
(i) reflexive: $\forall p \in P.\ p \leq p$
(ii) transitive: $\forall p, q, r \in P.\ p \leq q\ \&\ q \leq r \Rightarrow p \leq r$
(iii) antisymmetric: $\forall p, q \in P.\ p \leq q\ \&\ q \leq p \Rightarrow p = q$.

A *total* order is a partial order $(P, \leq)$ in which every pair of elements are *comparable* in the sense that

$$p \leq q \text{ or } q \leq p$$

for all $p, q \in P$. If we relax the definition of partial order and do not insist on (iii) antisymmetry, and only retain (i) reflexivity and (ii) transitivity, we have defined a *preorder* on a set.

Example: Let P be a set consisting of subsets of a set S. Then P with the subset relation, $(P, \subseteq)$, is a partial order. □

Finite partial orders can be drawn as very special directed graphs. Taking the economy of not drawing unnecessary arcs, *viz.* the identity arcs and those that follow from transitivity, one obtains the *Hasse diagram* of a finite partial order.

Exercise 3.21 Draw the Hasse diagram of the partial order $(P, \subseteq)$ where P consists of all subsets of $\{a, b, c\}$. □

Often the partial order itself supports extra structure. For example, in a partial order $(P, \leq)$, we often consider the *least upper bound* (*lub*, or *supremum*, or *join*) of a subset of P. Let $X \subseteq P$. An element p such that $(\forall x \in X.\ x \leq p)$ is called an *upper bound* of the subset X. Accordingly, a *least upper bound* of X is an element $u \in P$ which is both an upper bound, *i.e.*

$$\forall x \in X.\ x \leq u\ ,$$

and the least such, *i.e.* for all $p \in P$,

$$(\forall x \in X.\ x \leq p) \Rightarrow u \leq p\ .$$

Note that if a least upper bound of a set exists it has to be unique; any two least upper bounds u_1 and u_2 have to satisfy both $u_1 \leq u_2$ and $u_2 \leq u_1$, and hence be equal. When it exists *the* least upper bound of a subset X is written as $\bigvee X$. Note that if $\bigvee \emptyset$ exists it is the least element in P, because then all p in P are upper bounds of $\emptyset$ so

$$\bigvee \emptyset \leq p\ .$$

In a dual way, an element p such that $(\forall x \in X.\ p \leq x)$ is called a *lower bound* of a subset $X \subseteq P$. The *greatest lower bound* (*glb*, *infimum* or *meet*) of a subset $X \subseteq P$. is an element $l \in P$ which is a lower bound of X and such that for all $p \in P$,

$$(\forall x \in X.\ p \leq x) \Rightarrow p \leq l\ .$$

A greatest lower bound of a subset X is unique if it exists, and is written as $\bigwedge X$. Note that if $\bigwedge \emptyset$ exists it is the greatest element in P, because then for all $p \in P$,

$$p \leq \bigwedge \emptyset \ .$$

A partial order need not have all lubs and glbs. When it has lubs and glbs of all subsets it is called a *complete lattice*.

Example: Let P be a set consisting of *all* subsets of a set S. (In other words P is the *powerset* of S—see the next chapter.) Then P with the subset relation, $(P, \subseteq)$, is a partial order with all lubs and glbs—lubs are given by unions and glbs by intersections. □

Exercise 3.22 Let $(\mathbb{N}, \leq)$ be the set of natural numbers with the relation $m \leq n$ meaning m *divides* n. Show $(\mathbb{N}, \leq)$ is a partial order with lubs and glbs of all pairs. What are these lubs and glbs in more traditional terms? If $\mathbb{N}$ is replaced by $\mathbb{Z}$, does the divides relation still yield a partial order? □

Exercise 3.23 Show that if a partial order has all lubs, then it necessarily also has all glbs and *vice versa*. □

Exercise 3.24 Let $(P, \lesssim)$ be a preorder. Define the relation $\simeq$ on P by

$$p \simeq q \text{ iff } p \lesssim q \ \&\ q \lesssim p \ .$$

Show $\simeq$ is an equivalence relation. Define $(P/\simeq, \leq)$ to comprise the set $P/\simeq$ of $\simeq$-equivalence classes on which

$$x \leq y \text{ iff } \exists p, q.\ x = \{p\}_\simeq \ \&\ y = \{q\}_\simeq \ \&\ p \lesssim q \ .$$

Show $(P/\simeq, \leq)$ is a partial order. [The partial order $(P/\simeq, \leq)$ is often written $(P/\simeq,\ \lesssim /\simeq)$ and called the *quotient* of the preorder $(P, \lesssim)$.] □

3.4 Size of sets

A useful way to compare sets is through an idea of their size. Write $A \cong B$ to mean there is bijective correspondence between sets A and B. The relation $A \cong B$ satisfies the properties of an equivalence relation on sets. Two sets in the relation $\cong$ are said to have the same *size* or *cardinality*.[1]

3.4.1 Countability

In computation we are particularly concerned with sets whose size does not exceed that of the set of natural numbers $\mathbb{N}$, sets which are said to be countable because they can be paired off, in the manner of counting, with initial segments of the natural numbers, or possibly even the whole of the natural numbers. Here's the definition.

A set A is *finite* iff there is a bijection from the set $\{m \in \mathbb{N} \mid m \leq n\}$ to A for some $n \in \mathbb{N}_0$; in other words, A is empty or in 1-1 correspondence with a set $\{1, 2, \cdots, n\}$. We say a set is *infinite* iff it is not finite. A set A is *countable* iff it is finite or there is a bijection

$$f : \mathbb{N} \to A \ .$$

For example, the sets of natural numbers $\mathbb{N}$, of integers $\mathbb{Z}$, of rational numbers $\mathbb{Q}$ and real numbers $\mathbb{R}$ are all infinite. The set $\mathbb{N}$ is countable, as are $\mathbb{Z}$ and $\mathbb{Q}$, while $\mathbb{R}$ is not countable—we'll see this shortly.

Lemma 3.25 *Any subset of natural numbers is countable.*

Proof. Let A be a subset of $\mathbb{N}$. Define a partial function $f : \mathbb{N} \rightharpoonup A$ by taking

- $f(1)$ to be the least number in A if A is nonempty and undefined otherwise, and

[1] In fact Russell and Whitehead's definition of the cardinal numbers, including the natural numbers, was as $\cong$-equivalence classes.

- $f(n+1)$, where $n \in \mathbb{N}$, to be the least number in A which is larger than $f(n)$ if $f(n)$ is defined and there is a member of A larger that $f(n)$; and to be undefined otherwise.

This definition of f is an example of *definition by mathematical induction*: we first define the basis of the definition by induction, $f(1)$, and then the induction step of the definition by induction, defining $f(n+1)$ in terms of $f(n)$.

From the definition of f it is clear that if $f(n+1)$ is defined, then so is $f(n)$ and $f(n) < f(n+1)$, for all $n \in \mathbb{N}$. It follows that if $n < n'$ and $f(n')$ is defined, then $f(n)$ is defined and $f(n) < f(n')$. Hence

$$D = \{n \in \mathbb{N} \mid f(n) \text{ is defined}\}$$

is either $\mathbb{N}$, or of the form $\{n \in \mathbb{N} \mid n \leq m\}$, a finite initial segment of the natural numbers. Furthermore $f : D \to A$ is injective as two distinct elements of D will be strictly ordered so have distinct images under f.

To show f is also surjective, suppose otherwise. Then there would be a *least* $a \in A$ not in the image fD. The element a cannot be the least element of A because this least element is $f(1)$, clearly in fD. So there must be a largest element a' of A such that $a' < a$. Because $a' < a$ there is $n \in D$ such that $f(n) = a'$. But then $f(n+1) = a$—contradiction. □

Corollary 3.26 *A set B is countable iff there is a bijection $g : A \to B$ from $A \subseteq \mathbb{N}$.*

Proof. *"only if":* follows directly from the definition of countability. *"if":* By Lemma 3.25 a subset $A \subseteq \mathbb{N}$ is countable so there is a bijection $f : D \to A$ where D is a finite initial segment or the whole of $\mathbb{N}$. The composition $g \circ f : D \to B$ is a bijection establishing the countability of B. □

In establishing countability of a set we do not need to be so demanding as Corollary 3.26; an injection from the set into the natural numbers suffices:

Lemma 3.27 *A set B is countable iff there is an injection $f : B \to \mathbb{N}$.*

Proof. *"only if":* Assuming B is countable there is a bijection $g : A \to B$ from $A \subseteq \mathbb{N}$ by Corollary 3.26. The function g has an inverse function $g^{-1} : B \to A$, by Lemma 3.9. Let $j; A \to \mathbb{N}$ be the inclusion function. Then $f = j \circ g^{-1} : B \to \mathbb{N}$ is an injection, being the composition of two injections. *"if":* An injection $f : B \to \mathbb{N}$ becomes a bijection $f : B \to f\,B$ because f regarded as a function from B to the direct image $f\,B$ is clearly both injective and surjective. Now its inverse $f^{-1} : f\,B \to B$ is a bijection from $f\,B \subseteq \mathbb{N}$. Hence by Corollary 3.26 the set B is countable. □

So, a set is countable iff there is an injection from it into the natural numbers. We could instead have taken this as our definition of countability. Though our current definition has the advantage of being more customary and directly related to our intuitions about counting. As we will see, the following fact, a slight relaxation of Lemma 3.27, is often useful in establishing that a set is countable.

Lemma 3.28 *A set B is countable iff there is an injection $f : B \to A$ into a set A which is countable.*

Proof. *"only if":* If B is countable, then there is an injection $f : B \to \mathbb{N}$ by Lemma 3.27, and $\mathbb{N}$ is countable. *"if":* Suppose $f : B \to A$ is an injection and the set A is countable. Then, by Lemma 3.27, there is an injection $h : A \to \mathbb{N}$. It follows that the composition $g = h \circ f : B \to \mathbb{N}$ is an injection. So, again by Lemma 3.27, the set B is countable. □

Notice that if $B \subseteq A$ then there is an *inclusion* function from B to A taking $b \in B$ to $b \in A$—the inclusion function is clearly injective. So Lemma 3.28 specialises to say a set B is countable iff it is included in a countable set A.

Lemma 3.29 *The set $\mathbb{N} \times \mathbb{N}$ is countable.*

Proof. The function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ defined by

$$f(m, n) = 2^m \times 3^n$$

is an injection into a countable set. By Lemma 3.28, $\mathbb{N} \times \mathbb{N}$ is countable. □

Corollary 3.30 *The set of positive rational numbers $\mathbb{Q}^+$ is countable.*

Proof. Any positive rational q can be written uniquely as a fraction m_q/n_q where m_q and n_q are natural numbers with no common factor. Define a function $f : \mathbb{Q}^+ \to \mathbb{N} \times \mathbb{N}$ by taking $f(q) = (m_q, n_q)$. Then f is an injection—two different rationals determine different fractions. So $\mathbb{Q}^+$ is countable by Lemma 3.28. □

Corollary 3.31 *If A and B are countable sets, then so is their product $A \times B$.*

Proof. Assume sets A and B are countable. Then there are injections $f_A : A \to \mathbb{N}$ and $f_B : B \to \mathbb{N}$, by Lemma 3.28. We can combine them into an lnjection $f : A \times B \to \mathbb{N} \times \mathbb{N}$ by defining $f(a, b) = (f_A(a), f_B(b))$. We spell out the easy check that f is an injection, which uses standard properties of pairs and that f_A and f_B are injections. Let (a, b) and (a', b') be pairs in $A \times B$ for which $f(a, b) = f(a', b')$. We obtain $(f_A(a), f_B(b)) = (f_A(a'), f_B(b'))$. Whence, $f_A(a) = f_A(a')$ and $f_B(b) = f_B(b')$. So $a = a'$ and $b = b'$ by the injectivity of f_A and f_B. Thus $(a, b) = (a', b')$. □

We have seen unions, like $A_1 \cup A_2 \cup \cdots \cup A_k$, of finitely sets $A_1, \cdots, A_k$. Imagine now that we are given an infinite sequence of sets $A_1, A_2, A_3, \cdots, A_n, \cdots$ indexed by the natural numbers. We can also form their union; we might write that union as

$$A_1 \cup A_2 \cup A_3 \cup \cdots A_n \cup \cdots$$

though this perhaps makes the union seem more mysterious than it is, because it suggests wrongly a form of limiting process like those in real or complex analysis. A better way to write the union of the sequence of sets is as

$$\bigcup_{n \in \mathbb{N}} A_n =_{def} \{x \mid \exists n \in \mathbb{N}.\ x \in A_n\} ,$$

which exposes how innocent this union of countably many sets really is. The next lemma shows that if each of the sets A_n is countable then so is their union. The lemma is often expressed as: a countable union of countable sets is countable.

Lemma 3.32 *Suppose $A_1, A_2, \cdots, A_n, \cdots$ are all countable sets. Their union $\bigcup_{n \in \mathbb{N}} A_n$ is countable.*

Proof. Write A for the set $\bigcup_{n \in \mathbb{N}} A_n$. By Lemma 3.27, for each n there is an injection $f_n : A_n \to \mathbb{N}$. Define an injection $h : A \to \mathbb{N} \times \mathbb{N}$ as follows. For $x \in A$, let n_x be the least number for which $x \in A_{n_x}$, and take

$$h(x) = (n_x, f_{n_x}(x)) .$$

We check that h is an injection: Suppose $h(x) = h(y)$ for $x, y \in A$. Then $n_x = n_y$ so $x, y \in A_{n_x}$ and $f_{n_x}(x) = f_{n_x}(y)$. But f_{n_x} is injective, so $x = y$. Hence A is countable by Lemmas 3.28 and 3.29. □

Notice thay the above lemma also applies to finite unions, because the A_n's could all be the empty set from some point on.

Exercise 3.33 Prove that the set $\mathbb{Z}$ of integers and the set $\mathbb{Q}$ of all rational numbers are countable. [Both proofs involve the same idea.] □

Exercise 3.34 Show that the set of all *finite* subsets of $\mathbb{N}$ is countable. □

Exercise 3.35 Show that $\mathbb{Q} \times \mathbb{Q}$ is countable. Deduce that any set of disjoint discs (*i.e.* circular areas which may or may not include their perimeter) in the plane $\mathbb{R} \times \mathbb{R}$ is countable. Is the same true if "discs" is replaced by "circles" (*i.e.* just the perimeters of the circles)? □

Exercise 3.36 Show that a nonempty set A is countable iff there is a surjection $f : \mathbb{N} \to A$. □

3.4.2 Uncountability

Not all sets are countable. One of the notable mathematical accomplishments of the 19th century was Georg Cantor's proof that the set of real numbers $\mathbb{R}$ is uncountable, *i.e.* not countable. This opened up new ways to prove the existence of certain kinds of real numbers. His second, simpler proof of the uncountability of $\mathbb{R}$ used a *diagonal argument*, a style of argument which reappears in showing the undecidability of the halting problem, is implicit in the proof of Gödel's incompleteness theorem, and can sometimes be used in establishing the hierarchies of complexity theory.

Theorem 3.37 *The set of real numbers $\mathbb{R}$ is uncountable.*

Proof. The proof is by contradiction. Assume that $\mathbb{R}$ is countable. Then by Lemma 3.28, the interval $(0,1] = \{r \in \mathbb{R} \mid 0 < r \leq 1\} \subseteq \mathbb{R}$ will also be a countable set. Each number in $(0,1]$ is represented uniquely by a non-terminating decimal expansion; for example 0.13 is represented by the non-terminating decimal $0.12999\cdots$. The set $(0,1]$ is clearly infinite (any finite set of reals would contain a least element, which $(0,1]$ clearly does not). So $(0,1]$ being countable implies there is a bijection $f : \mathbb{N} \to (0,1]$. Write

$$f(n) = 0.d_1^n d_2^n d_3^n \cdots d_i^n \cdots$$

to describe the non-terminating decimal expansion of the nth real in the enumeration given by f. We'll produce a number r in $(0,1]$ which can't be in the enumeration, so yielding a contradiction. Define the number's decimal expansion to be

$$0.r_1 r_2 r_3 \cdots r_i \cdots$$

where

$$r_i = \begin{cases} 1 & \text{if } d_i^i \neq 1, \\ 2 & \text{if } d_i^i = 1\ . \end{cases}$$

Clearly $r \in (0,1]$. Thus there is some natural number k such that $f(k) = r$. Hence by the uniqueness of the decimal expansions $r_i = d_i^k$ for all i. In particular,

$$r_k = d_k^k\ .$$

But, from the definition of r_k, we have that $r_k = 1$ if $d_k^k \neq 1$, and $r_k = 2$ if $d_k^k = 1$. In either case, $r_k \neq d_k^k$—a contradiction.

We conclude that the original assumption, that $\mathbb{R}$ is countable, is false. □

To see why it is called a diagonal argument, imagine writing the enumerations as an array:

$$\begin{array}{lccccccc}
f(1) = & 0. & d_1^1 & d_2^1 & d_3^1 & \cdots & d_i^1 & \cdots \\
f(2) = & 0. & d_1^2 & d_2^2 & d_3^2 & \cdots & d_i^2 & \cdots \\
f(3) = & 0. & d_1^3 & d_2^3 & d_3^3 & \cdots & d_i^3 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
f(n) = & 0. & d_1^n & d_2^n & d_3^n & \cdots & d_i^n & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}$$

The decimal expansion of the real r which plays a key role in Cantor's argument is defined by running down the diagonal of the array changing 1's to 2's and non-1's to 1's. In this way the decimal expansion can never be in the enumeration; no matter which row one considers, the decimal expansion of r will differ on the diagonal.

Notice that Cantor's theorem establishes the existence of irrational numbers, in fact shows that the set of irrational numbers is uncountable, without exhibiting a single irrational number explicitly.

Exercise 3.38 Prove that the set of irrational numbers is uncountable. □

An analogous proof that there are uncountably many transcendental numbers is even more dramatic in that it is very hard to prove a number is transcendental. An *algebraic* number is a real number x that is the solution of a polynomial equation

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = 0$$

where $a_0, a_1, a_2, \cdots, a_n$ are integer coefficients. A real number which is not algebraic is called *transcendental*. There are only countably many such polynomial equations[2] and each has only finitely many solutions, so there are only countably many algebraic numbers. But there are uncountably many reals. It follows that there must be transcendental numbers, and indeed that the set of transcendental numbers is uncountable. (Do you now know a single transcendental number? Well, π and e are. But could you prove it? Probably not.)

Exercise 3.39 Prove that the set of transcendental numbers is uncountable. (This uses essentially the same idea as Exercise 3.38.) □

Investigating cardinality

By Lemma 3.28, if $f : B \to A$ is an injection and A is countable, then so is B. So, when investigating cardinality,

- to show a set B is countable it suffices to exhibit an injection from B set into a set A known to be countable; while
- to show a set A is uncountable it suffices to exhibit an injection from a set B known to be uncountable into A. Because then, if A were countable, so would B be countable—a contradiction.

Sometimes (though rarely in undergraduate courses) we need to investigate cardinality beyond countability or its failure. Then the key tool is the Schröder-Bernstein theorem.This says that two sets A and B have the same cardinality iff there are injections $f : A \to B$ and $g : B \to A$. The hard part of its proof shows how to construct a bijection between A and B out of the two injections—see Exercise 3.42.

Exercise 3.40 By using a variation on the diagonal argument above, show that the powerset, $\mathcal{P}(\mathbb{N}) =_{def} \{S \mid S \subseteq \mathbb{N}\}$, is uncountable. (See Section 4.3.2 for a proof.) □

Exercise 3.41 Which of the following sets are finite, which are infinite but countable, and which are uncountable?

- $\{f : \mathbb{N} \to \{0,1\} \mid \forall n \in \mathbb{N}.\ f(n) \leq f(n+1)\}$
- $\{f : \mathbb{N} \to \{0,1\} \mid \forall n \in \mathbb{N}.\ f(2n) \neq f(2n+1)\}$
- $\{f : \mathbb{N} \to \{0,1\} \mid \forall n \in \mathbb{N}.\ f(n) \neq f(n+1)\}$
- $\{f : \mathbb{N} \to \mathbb{N} \mid \forall n \in \mathbb{N}.\ f(n) \leq f(n+1)\}$
- $\{f : \mathbb{N} \to \mathbb{N} \mid \forall n \in \mathbb{N}.\ f(n) \geq f(n+1)\}$

□

Exercise 3.42 (Schröder-Bernstein theorem) Let $f : A \to B$ and $g : B \to A$ be injections between two sets A and B. Note that the function $g : B \to g\,B$ is a bijection, with inverse function $g^{-1} : g\,B \to B$. Define subsets $C_1, C_2, \cdots, C_n, \cdots$ of A by the following induction:

$$\begin{aligned} C_1 &= A \setminus (g\,B) \\ C_{n+1} &= (g \circ f)\,C_n \end{aligned} \quad .$$

[2] A polynomial is determined by its coefficients. So polynomials with integer coefficients are in 1-1 correspondence with tuples of integers in the set $\bigcup_{n \in N} \mathbb{Z}^n$, a countable union of countable sets, so countable.

Define $C = \bigcup_{n\in\mathbb{N}} C_n$. Now define a function $h : A \to B$ by

$$h(a) = f(a) \qquad \text{if } a \in C,$$
$$h(a) = g^{-1}(a) \quad \text{if } a \notin C$$

—if $a \notin C$ then $a \in g\,B$, so it makes sense to apply g^{-1} to $a \notin C$.

Prove h is a bijection from A to B by showing:

(i) h is injective: Show if $a \in C$ and $a' \notin C$, then $h(a) \neq h(a')$. Deduce that h is injective.

(ii) h is surjective: Show $(g \circ f)\,C = C \setminus C_1$. Deduce if $b \notin f\,C$, then $g(b) \notin C$, and hence that h is surjective.

□

Chapter 4

Constructions on sets

Forewarned by a problem first exposed by Bertrand Russell, we look to safe methods for constructing sets.

4.1 Russell's paradox

When set theory was being invented it was thought, first of all, that any property $P(x)$ determined a set

$$\{x \mid P(x)\} \ .$$

It came as a shock when Bertrand Russell realised that assuming the existence of certain sets described in this way gave rise to contradictions.[1]

Russell's paradox is really the demonstration that a contradiction arises from the liberal way of constructing sets above. His argument proceeds as follows. Consider the property

$$x \notin x$$

a way of writing "x is not an element of x." If we assume that properties determine sets, just as described, we can form the set

$$R = \{x \mid x \notin x\} \ .$$

Either $R \in R$ or not. If so, *i.e.* $R \in R$, then in order for R to qualify as an element of R, from the definition of R, we deduce $R \notin R$. So we end up asserting both something and its negation—a contradiction. If, on the other hand, $R \notin R$ then from the definition of R we see $R \in R$—a contradiction again. Either $R \in R$ or $R \notin R$ lands us in trouble.

We need to have some way which stops us from considering a collection like R as a set, and so as a legitimate element. In general terms, the solution is to discipline the way in which sets are constructed, so that starting from certain given sets, new sets can only be formed when they are constructed by using particular, safe ways from old sets. We shall state those sets we assume to exist right from the start and methods we allow for constructing new sets. Provided these are followed we avoid trouble like Russell's paradox and at the same time have a rich enough world of sets to support most mathematics.[2]

4.2 Constructing sets

4.2.1 Basic sets

We take the existence of the empty set $\emptyset$ for granted, along with certain sets of basic elements such as

$$\mathbb{N}_0 = \{0, 1, 2, \cdots\} \ .$$

We shall also take sets of symbols like

$$\{\text{“a”}, \text{“b”}, \text{“c”}, \text{“d”}, \text{“e”}, \ \cdots, \text{“z”}\}$$

[1] The shock was not just to Russell and his collaborator Alfred North Whitehead. Gottlob Frege received the news as his book on the foundations of mathematics via sets was being printed—the paradox was devastating for his work. Some were delighted however. The great mathematician Henri Poincaré is reported as gleefully saying "Logic is not barren, it's brought forth a paradox!"

[2] Occasionally we consider collections which are not sets. For example, it can be useful to consider the collection of all sets. But such a collection is not itself a set, so cannot be made a proper element of any collection. The word 'class' which originally was synonymous with 'set' is now generally reserved for a collection which need not necessarily be a set.

for granted, although we could, alternatively have represented them as particular numbers for example. The equality relation on a set of symbols is that given by syntactic identity written $=$. Two symbols are equal iff they are literally the same.

4.2.2 Constructions

We shall take for granted certain operations on sets which enable us to construct sets from given sets.

Comprehension

If X is a set and $P(x)$ is a property, we can form the set

$$\{x \in X \mid P(x)\} \ ,$$

the subset of X consisting of all elements x of X which satisfy $P(x)$.

Sometimes we'll use a further abbreviation. Suppose $e(x_1, \ldots, x_n)$ is some expression which for particular elements $x_1 \in X_1, \cdots x_n \in X_n$ yields a particular element and $P(x_1, \ldots, x_n)$ is a property of such $x_1, \ldots, x_n$. We use

$$\{e(x_1, \ldots, x_n) \mid x_1 \in X_1 \ \& \ \cdots \& \ x_n \in X_n \ \& \ P(x_1, \ldots, x_n)\}$$

to abbreviate

$$\{y \mid \exists x_1 \in X_1, \cdots, x_n \in X_n. \ y = e(x_1, \ldots, x_n) \& \ P(x_1, \ldots, x_n)\} \ .$$

For example,

$$\{2m + 1 \mid m \in \mathbb{N} \ \& \ m > 1\}$$

is the set of odd numbers greater than 3.

Remark Note a consequence of comprehension. The collection $\{x \mid x \text{ is a set }\}$ is not itself a set. If it were then by using comprehension Russell's collection R would be a set, which is contradictory in the manner of Russell's original argument. As the collection of all sets is not a set we fortunately side-step having to consider whether it is a member of itself. □

Powerset

We can form a set consisting of the set of all subsets of a set, the so-called *powerset*:

$$\mathcal{P}(X) = \{Y \mid Y \subseteq X\}.$$

This is *the* important construction for building bigger sets. We shall see shortly that a powerset $\mathcal{P}(X)$ always has larger size than X.

Exercise 4.1 Let B be a fixed subset of the set A. Define the relation R on $\mathcal{P}(A)$ by

$$(X, Y) \in R \iff X \cap B = Y \cap B \ .$$

Show that R is an equivalence relation and describe a bijection between the set of R-equivalence classes and $\mathcal{P}(B)$. □

Unordered pairs

A seemingly modest but important way to produce sets is through forming unordered pairs. Given two objects x and y—they might be sets—we can form the set $\{x, y\}$ whose sole elements are x and y.

Indexed sets

Suppose I is a set and that for any $i \in I$ there is a unique object x_i, maybe a set itself. Then

$$\{x_i \mid i \in I\}$$

is a set. The elements x_i are said to be *indexed* by the elements $i \in I$. Any collection of objects indexed by a set is itself a set.

Union

As we've seen, the set consisting of the *union* of two sets has as elements those elements which are either elements of one or the other set:

$$X \cup Y = \{a \mid a \in X \text{ or } a \in Y\}.$$

This union is an instance of a more general construction, "big union," that we can perform on any set of sets.

Big union

Let X be a set of sets. Their *union*

$$\bigcup X = \{a \mid \exists x \in X.\ a \in x\}$$

is a set. Note that given two sets X and Y we can first form the set $\{X, Y\}$; taking its big union $\bigcup \{X, Y\}$ we obtain precisely $X \cup Y$. When $X = \{Z_i \mid i \in I\}$ for some indexing set I we often write $\bigcup X$ as $\bigcup_{i \in I} Z_i$.

The above operations are in fact enough for us to be able to define the remaining fundamental operations on sets, *viz.* intersection, product, disjoint union and set difference, operations which are useful in their own right.

Intersection

As we've seen, elements are in the *intersection* $X \cap Y$, of two sets X and Y, iff they are in both sets, *i.e.*

$$X \cap Y = \{a \mid a \in X \ \&\ a \in Y\}\ .$$

By the way, notice that one way to write $X \cap Y$ is as $\{a \in X \mid a \in Y\}$ comprising the subset of the set X which satisfy the property of also being in Y; so $X \cap Y$ is a set by Comprehension.

Big intersection

Let X be a nonempty collection of sets. Then

$$\bigcap X = \{a \mid \forall x \in X.\ a \in x\}$$

is a set called its *intersection*. Again, that such an intersection is a set follows by Comprehension.

When $X = \{Z_i \mid i \in I\}$ for a nonempty indexing set I we often write $\bigcap X$ as $\bigcap_{i \in I} Z_i$.[3]

Product

As we've seen, for sets X and Y, their *product* is the set

$$X \times Y = \{(a, b) \mid a \in X \ \&\ b \in Y\}\ ,$$

the set of ordered pairs of elements with the first from X and the second from Y.

More generally $X_1 \times X_2 \times \cdots \times X_n$ consists of the set of n-tuples $(x_1, x_2, \ldots, x_n)$. When all the components of a product are the same set X the n-ary product is written as X^n. By convention X^0, a zero-ary product, is generally understood to be a singleton set consisting just of the empty tuple $()$.

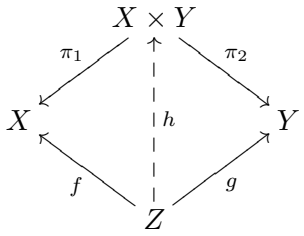
Exercise 4.2 Let X and Y be sets. Define the *projections*

$$\pi_1 : X \times Y \to X \text{ and } \pi_2 : X \times Y \to Y$$

by taking $\pi_1(a, b) = a$ and $\pi_2(a, b) = b$ for $(a, b) \in X \times Y$.

[3] In a context where all sets are understood to be subsets of a given universe U the empty intersection is taken to be U. In general though we can't assume there is a fixed set forming such a universe. We can't take the collection of all sets as the universe as this is not a set.

Let Z be a set and $f : Z \to X$ and $g : Z \to Y$. Show that there is a unique function $h : Z \to X \times Y$ such that $\pi_1 \circ h = f$ and $\pi_2 \circ h = g$.



□

Disjoint union

Frequently we want to join sets together but, in a way which, unlike union, does not identify the same element when it comes from different sets. We do this by making copies of the elements so that when they are copies from different sets they are forced to be distinct:

$$X_1 \uplus X_2 \uplus \cdots \uplus X_n = (\{1\} \times X_1) \cup (\{2\} \times X_2) \cup \cdots \cup (\{n\} \times X_n).$$

In particular, for $X \uplus Y$ the copies $(\{1\} \times X)$ and $(\{2\} \times Y)$ have to be disjoint, in the sense that

$$(\{1\} \times X) \cap (\{2\} \times Y) = \emptyset,$$

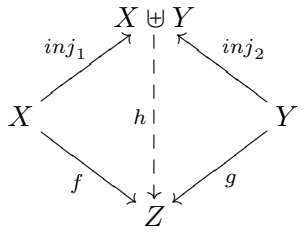
because any common element would be a pair with first element both equal to 1 and 2, clearly impossible.

Exercise 4.3 Let X and Y be sets. Define the *injections*

$$inj_1 : X \to X \uplus Y \text{ and } inj_2 : Y \to X \uplus Y$$

by taking $inj_1(a) = (1, a)$ for $a \in X$, and $inj_2(b) = (2, b)$ for $b \in Y$.

Let Z be a set and $f : X \to Z$ and $g : Y \to Z$. Show that there is a unique function $h : X \uplus Y \to Z$ such that $h \circ inj_1 = f$ and $h \circ inj_2 = g$.



□

Set difference

We can subtract one set Y from another X, an operation which removes all elements from X which are also in Y.

$$X \setminus Y = \{x \mid x \in X \ \& \ x \notin Y\} .$$

4.2.3 Axioms of set theory

The constructions we have described include most of the assumptions made in more axiomatic treatments of set theory based on the work of Zermelo and Frænkel. To spell out the connection, the existence of the set of natural numbers, powersets, unordered pairs, big unions and sets of indexed objects correspond to the axioms of *infinity*, *powerset*, *pairing*, *union* and *replacement* respectively; we have adopted the axiom of *comprehension* directly, and the axiom of *extensionality* amounts to saying a set is determined by its elements. For completeness we mention two remaining axioms, the axiom of *foundation* and the axiom of *choice*, which are generally assumed of sets. While sensible and safe axioms to assume of sets, they do not in fact follow from the constructions we have given so far.

The axiom of foundation

A set is built-up starting from basic sets by using the constructions described. We remark that a property of sets, called the *axiom of foundation*, follows from our informal understanding of sets and how we can construct them. Consider an element b_1 of a set b_0. It is either a basic element, like an integer or a symbol, or it is a set. If b_1 is a set then it must have been constructed from sets which have themselves been constructed earlier. Intuitively, we expect any chain of memberships

$$\cdots b_n \in \cdots \in b_1 \in b_0$$

to end in some b_n which is some basic element or the empty set. The statement that any such descending chain of memberships must be finite is called the axiom of foundation, and is an assumption generally made in set theory. Notice the axiom implies that no set X can be a member of itself as, if this were so, we'd get the infinite descending chain

$$\cdots X \in \cdots \in X \in X$$

—a contradiction.

General products and the axiom of choice

Occasionally it is important to have a general form of product in which instead of pairs with first and second coordinates, or finite tuples, we have tuples where the coordinates correspond to indices in a general set I. Let X_i be a set for each element i in a set I. By definition the *general product*

$$\prod_{i\in I} X_i = \{f : I \to \bigcup_{i\in I} X_i \mid \forall i \in I.\ f(i) \in X_i\}\ .$$

Given $f \in \prod_{i\in I} X_i$ we can get the ith coordinate as $f(i)$. Given $x_i \in X_i$ for each $i \in I$, we can form their tuple $f \in \prod_{i\in I} X_i$ by defining $f(i) = x_i$ for all $i \in I$. It's not too hard to construct the set $\prod_{i\in I} X_i$ out of the earlier constructions.

Is the set $\prod_{i\in I} X_i$ nonempty? It has to be empty if $X_i = \emptyset$ for any $i \in I$. But if X_i is nonempty for every $i \in I$ it would seem reasonable that one could make a function $f \in \prod_{i\in I} X_i$ by *choosing* some $f(i) \in X_i$ for all $i \in I$. When I is a finite set it is easy to make such a tuple, and so prove that $\prod_{i\in I} X_i$ is nonempty if each X_i is. For a while it was thought that this perfectly reasonable property was derivable from more basic axioms even when I is infinite. However, this turns out not to be so. Occasionally one must have recourse to the *axiom of choice* which says provided each X_i is nonempty for $i \in I$, then so is the product $\prod_{i\in I} X_i$.

4.3 Some consequences

4.3.1 Sets of functions

The set of all relations between sets X and Y is the set $\mathcal{P}(X \times Y)$. Using comprehension it is then easy to see that

$$(X \rightharpoonup Y) = \{f \in \mathcal{P}(X \times Y) \mid f \text{ is a partial function}\}$$

is a set; that of all partial functions from X to Y. Similarly,

$$(X \to Y) = \{f \in \mathcal{P}(X \times Y) \mid f \text{ is a function}\}$$

is also a set; that of all total functions from X to Y. Often this set is written Y^X.

Exercise 4.4 Let $A_2 = \{1, 2\}$ and $A_3 = \{a, b, c\}$. List the elements of the four sets $(A_i \to A_j)$ for $i, j \in \{2, 3\}$. Annotate those elements which are injections, surjections and bijections. □

Exercise 4.5 Let X and Y be sets. Show there is a bijection between the set of functions $(X \to \mathcal{P}(Y))$ and the set of relations $\mathcal{P}(X \times Y)$. □

When investigating the behaviour of a function $f \in (X \to Y)$ we apply it to arguments. Earlier in Proposition 3.3 we saw that equality of functions $f, f' \in (X \to Y)$ amounts to their giving the same result on an arbitrary argument x in X. We can treat functions as sets and so might introduce a function by describing the property satisfied by its input-output pairs. But this would ignore the fact that a function is most often introduced as an expression e describing its output in Y in terms of its input x in X. For this manner of description lambda notation (or λ-notation) is most suitable.

Lambda notation

Lambda notation provides a way to describe functions without having to name them. Suppose $f : X \to Y$ is a function which for any element x in X gives a value $f(x)$ described by an expression e, probably involving x. Sometimes we write

$$\lambda x \in X.\ e$$

for the function f. Thus

$$\lambda x \in X.\ e \ = \ \{(x, e) \mid x \in X\} \ .$$

So, $\lambda x \in X.\ e$ is an abbreviation for the set of input-output pairs determined by the expression e. For example, $\lambda x \in \mathbb{N}_0.\ x + 1$ is the successor function and we have $(\lambda x \in \mathbb{N}_0.\ x + 1) \in (\mathbb{N}_0 \to \mathbb{N}_0)$.

Exercise 4.6 Use lambda notation to describe bijections

$$[(A \times B) \to C] \cong [A \to (B \to C)] \ ,$$
$$[A \to (B \to C)] \cong [B \to (A \to C)] \ .$$

□

Exercise 4.7 Describe explicit bijections

$$[(A \uplus B) \to C] \cong (A \to C) \times (B \to C),$$
$$[A \to (B \times C)] \cong (A \to B) \times (A \to C) \ .$$

□

Characteristic functions

In set theory one starts with sets as primitive and builds functions. We built sets of functions $(X \to Y)$ with the help of powersets. There are alternative foundations of mathematics which work the other way round. They start with functions and "types" of functions $(X \to Y)$ and identify sets with special functions called *characteristic* functions to truth values. The correspondence between sets and characteristic functions is explored in the following exercise.[4]

Exercise 4.8 Let X be a set. The set $\{\mathtt{T}, \mathtt{F}\}$ consists of the truth values $\mathtt{T}$ and $\mathtt{F}$. Let $Y \subseteq X$. Define its *characteristic function* $\chi_Y : X \to \{\mathtt{T}, \mathtt{F}\}$ by taking

$$\chi_Y(x) = \begin{cases} \mathtt{T} & \text{if } x \in Y \\ \mathtt{F} & \text{if } x \notin Y \end{cases}$$

for all $x \in X$. Show the function taking Y to χ_Y is a bijection from $\mathcal{P}(X)$ to $(X \to \{\mathtt{T}, \mathtt{F}\})$. □

4.3.2 Sets of unlimited size

Cantor used a *diagonal argument* to show that X and $\mathcal{P}(X)$ are never in 1-1 correspondence for any set X. This fact is intuitively clear for finite sets but also holds for infinite sets. It implies that there is no limit to the size of sets.

Cantor's argument is an example of proof by contradiction. Suppose a set X is in 1-1 correspondence with its powerset $\mathcal{P}(X)$. Let $\theta : X \to \mathcal{P}(X)$ be the 1-1 correspondence. Form the set

$$Y = \{x \in X \mid x \notin \theta(x)\}$$

which is clearly a subset of X and therefore in correspondence with an element $y \in X$. That is $\theta(y) = Y$. Either $y \in Y$ or $y \notin Y$. But both possibilities are absurd. For, if $y \in Y$ then $y \in \theta(y)$ so $y \notin Y$, while, if $y \notin Y$ then $y \notin \theta(y)$ so $y \in Y$. We conclude that our first supposition must be false, so there is no set in 1-1 correspondence with its powerset.

[4] The seminal work on founding mathematics on functions is Alonzo Church's higher order logic. You can learn more on higher order logic and its automation in later courses of Mike Gordon and Larry Paulson.

Cantor's argument is reminiscent of Russell's paradox. But whereas the contradiction in Russell's paradox arises out of a fundamental, mistaken assumption about how to construct sets, the contradiction in Cantor's argument comes from denying the fact one wishes to prove.

As a reminder of why it is called a diagonal argument, imagine we draw a table to represent the 1-1 correspondence θ along the following lines. In the xth row and yth column is placed `T` if $y \in \theta(x)$ and `F` otherwise. The set Y which plays a key role in Cantor's argument is defined by running down the diagonal of the table interchanging `T`'s and `F`'s in the sense that x is put in the set iff the xth entry along the diagonal is `F`.

	$\cdots$	x	$\cdots$	y	$\cdots$
$\vdots$		$\vdots$		$\vdots$	
$\theta(x)$	$\cdots$	`T`	$\cdots$	`F`	$\cdots$
$\vdots$		$\vdots$		$\vdots$	
$\theta(y)$	$\cdots$	`F`	$\cdots$	`F`	$\cdots$
$\vdots$		$\vdots$		$\vdots$	

Exercise 4.9 This exercise guides you through to a proof that for any sets X and Y, with Y containing at least two elements, there cannot be an injection from the set of functions $(X \to Y)$ to X.

(i) Let X be a set. Prove there is no injection $f : \mathcal{P}(X) \to X$. [Hint: Consider the set $W =_{\text{def}} \{f(Z) \mid Z \subseteq X \;\&\; f(Z) \notin Z\}$.]

(ii) Suppose now that a set Y has at least two distinct elements. Define an injection $k : \mathcal{P}(X) \to (X \to Y)$, from the powerset of X to the set of functions from X to Y.

(iii) Prove that there is no injection from $(X \to Y)$ to X when the set Y has at least two distinct elements. [Hint: Recall that the composition of injections is an injection.]

□

Chapter 5

Inductive definitions

This chapter shows where induction principles come from. It is an introduction to the theory of inductively-defined sets. It provides general methods for defining sets recursively, and general induction rules to accompany inductively-defined sets.

5.1 Sets defined by rules—examples

Often a set is described in the following way. Some clauses stipulate that certain basic elements are to be in the set; then clauses are given stipulating further elements of the set in terms of elements already included. Implicitly, only elements produced in these stipulated ways are to be included in the set. This gives a kind of recipe for making a set: first put in the basic elements; then, conditional on certain elements being in, add others. Sets described in this way are called *inductively defined*.

Inductively defined sets are ubiquitous, but not always presented in the same way. To stress the commonality in their form of definition we'll present inductively-defined sets via rules. A rule instance comprises its premises and a conclusion

$$\frac{x_1, x_2, \cdots}{y}\ .$$

The intended interpretation is: if the premises $x_1, x_2, \cdots$ are in the set being defined, then so is the conclusion y. The premises may form an empty set, in which case the rule simply expresses that the conclusion y is in the set. The following examples only give a limited idea of the range of applications of inductive definitions—you'll meet them again and again, in applications as diverse as semantics, logic, verification, logic programming, datatypes, compiler construction, security, probability, ...

Example: The syntax of Boolean propositions has been described by

$$\mathtt{A}, \mathtt{B}, \ldots ::= \mathtt{a}, \mathtt{b}, \mathtt{c}, \cdots \mid \mathtt{T} \mid \mathtt{F} \mid \mathtt{A} \wedge \mathtt{B} \mid \mathtt{A} \vee \mathtt{B} \mid \neg \mathtt{A}$$

where $\mathtt{a}, \mathtt{b}, \mathtt{c} \cdots \in \mathtt{Var}$ belong to a set of propositional variables, $\mathtt{Var}$. We might instead describe the syntax of propositions by rules of the form:

$$\frac{}{\mathtt{a}}\ \mathtt{a} \in \mathtt{Var} \qquad \frac{}{\mathtt{T}} \qquad \frac{}{\mathtt{F}}$$

$$\frac{\mathtt{A} \quad \mathtt{B}}{\mathtt{A} \wedge \mathtt{B}} \qquad \frac{\mathtt{A} \quad \mathtt{B}}{\mathtt{A} \vee \mathtt{B}} \qquad \frac{\mathtt{A}}{\neg \mathtt{A}}$$

Each rule gives a step in the way of building up a Boolean proposition. Some rules say how to build propositions like $\mathtt{A} \wedge \mathtt{B}$ out of propositions $\mathtt{A}$ and $\mathtt{B}$ built earlier. Others assert that there are basic propositions like $\mathtt{T}$, or $\mathtt{a}$ where $\mathtt{a} \in \mathtt{Var}$. □

Example: The set of nonnegative integers $\mathbb{N}_0$ can be thought of as being generated in the following way: zero, 0, is a nonnegative integer; if n is a nonnegative integer, then so is $n + 1$. We can format these clauses in the form of rules:

$$\frac{}{0} \qquad \frac{n}{n+1}\ ,\ \text{where } n \in \mathbb{N}_0.$$

We can alternatively consider $\mathbb{N}_0$ as generated by the rules:

$$\frac{}{0} \qquad \frac{0, 1, \cdots, (n-1)}{n} \quad , \text{ where } n \in \mathbb{N}_0.$$

□

Example: The set of strings Σ^* over an alphabet of symbols Σ are defined by these clauses: ε is a string, the empty string; if x is a string and $a \in \Sigma$, then the concatenation ax is a string. Formatted as rules:

$$\frac{}{\varepsilon} \qquad \frac{x}{ax}\ a \in \Sigma$$

where we insist in the side condition that only symbols from the alphabet are used in making strings. (The set of *lists* over elements in Σ would be constructed in the same way, though generally writing [] for the empty list, and $a :: x$ for the concatenation of a in Σ to the front of a list x.)

The rules for producing nonnegative integers exampled above assumed the prior existence of such integers. At the dawn of thought, 'caveman numbers' could have been invented as strings of scratches '|' generated by the rules

$$\frac{}{|} \qquad \frac{x}{x|}$$

where x stands for a string of scratches. Repeated use of the rules would lead to

$$| \qquad || \qquad ||| \qquad |||| \qquad ||||| \qquad \cdots.$$

This shows how the natural numbers can be built from scratch! □

Example: Here are proof rules for establishing entailments between Boolean propositions:

$$\frac{}{\Gamma, \mathtt{A} \vdash \mathtt{A}} \qquad \frac{\Gamma \vdash \mathtt{A}}{\Gamma, \Delta \vdash \mathtt{A}} \qquad \frac{\Gamma \vdash \mathtt{A} \quad \Delta, \mathtt{A} \vdash \mathtt{B}}{\Gamma, \Delta \vdash \mathtt{B}}$$

$$\frac{}{\Gamma \vdash \mathtt{T}} \qquad \frac{\Gamma \vdash \mathtt{A}}{\Gamma, \mathtt{T} \vdash \mathtt{A}} \qquad \frac{}{\Gamma, \mathtt{F} \vdash \mathtt{A}}$$

$$\frac{\Gamma \vdash \mathtt{A} \quad \Gamma \vdash \mathtt{B}}{\Gamma \vdash \mathtt{A} \wedge \mathtt{B}} \qquad \frac{\Gamma \vdash \mathtt{A} \wedge \mathtt{B}}{\Gamma \vdash \mathtt{A}} \qquad \frac{\Gamma \vdash \mathtt{A} \wedge \mathtt{B}}{\Gamma \vdash \mathtt{B}}$$

$$\frac{\Gamma \vdash \mathtt{A}}{\Gamma \vdash \mathtt{A} \vee \mathtt{B}} \qquad \frac{\Gamma \vdash \mathtt{B}}{\Gamma \vdash \mathtt{A} \vee \mathtt{B}} \qquad \frac{\Gamma \vdash \mathtt{A} \vee \mathtt{B} \qquad \Gamma, \mathtt{A} \vdash \mathtt{C} \quad \Gamma, \mathtt{B} \vdash \mathtt{C}}{\Gamma \vdash \mathtt{C}}$$

$$\frac{\Gamma, \mathtt{A} \vdash \mathtt{F}}{\Gamma \vdash \neg \mathtt{A}} \qquad \frac{\Gamma \vdash \mathtt{A} \quad \Gamma \vdash \neg \mathtt{A}}{\Gamma \vdash \mathtt{F}} \qquad \frac{\Gamma \vdash \neg\neg \mathtt{A}}{\Gamma \vdash \mathtt{A}}$$

Above, Γ and Δ stand for sets of propositions. A pair of the form $\Gamma \vdash \mathtt{A}$ where Γ is a set of propositions and $\mathtt{A}$ is a proposition is called a *sequent*. The intention is that when $\Gamma \vdash \mathtt{A}$ is derivable, then the conjunction of all the propositions in Γ entails $\mathtt{A}$ For example, consider the rule

$$\frac{\Gamma \vdash \mathtt{A} \wedge \mathtt{B}}{\Gamma \vdash \mathtt{A}}\ .$$

This rule is intuitive, in that, for any model, if Γ entails the conjunction $\mathtt{A} \wedge \mathtt{B}$, then certainly Γ entails the conjunct $\mathtt{A}$. When we write Γ, Δ we mean $\Gamma \cup \Delta$, and $\Gamma, \mathtt{A}$ means $\Gamma \cup \{\mathtt{A}\}$. The rules define a set of sequents $\Gamma \vdash \mathtt{A}$, so essentially a set of pairs $(\Gamma, \mathtt{A})$ where Γ is a finite set of propositions and $\mathtt{A}$ is a proposition; starting from basic sequents like $\Gamma, \mathtt{A} \vdash \mathtt{A}$ or $\Gamma \vdash \mathtt{T}$ we get further sequents by repeatedly applying the rules. Though it's far from obvious, the proof rules above allow us to derive all the entailments of propositional logic: a

sequent $\{\mathtt{A}_1, \cdots, \mathtt{A}_k\} \vdash \mathtt{A}$ is derivable iff $\mathtt{A}_1 \wedge \cdots \wedge \mathtt{A}_k \models \mathtt{A}$. In particular the derivable sequents $\vdash \mathtt{A}$, where the left-hand-side of the entailment is empty, coincide with the tautologies.[1]

All the above proof rules are finitary—all the premises have at most size 3. To give an idea of why it is sometimes useful to have infinitary rules, imagine extending propositions by those of the form $\forall \mathtt{x}.\ \mathtt{A}(\mathtt{x})$ where $\mathtt{A}(\mathtt{x})$ becomes a proposition $\mathtt{A}(\mathtt{n})$ whenever any natural number $\mathtt{n}$ replaces the variable $\mathtt{x}$. We could then adjoin the infinitary proof rule

$$\frac{\Gamma \vdash \mathtt{A}(\mathtt{1}), \cdots, \Gamma \vdash \mathtt{A}(\mathtt{n}), \cdots}{\Gamma \vdash \forall \mathtt{x}.\ \mathtt{A}(\mathtt{x})}\ .$$

Fortunately, there are also finitary rules to prove universal statements. There are however logics in computer science with infinitary rules. □

Example: The evaluation and execution of programs can be captured by rules. As an indication we show how to express the evaluation of Boolean propositions by rules. The evaluation proceeds in the presence of a truth assignment t giving truth values to propositional variables. A judgement

$$\langle \mathtt{A}, t\rangle \longrightarrow V$$

is read as "Boolean proposition $\mathtt{A}$ with truth assignment t evaluates to truth value $\mathtt{V}$," where $\mathtt{V}$ is either $\mathtt{T}$ or $\mathtt{F}$.

$$\frac{}{\langle \mathtt{a}, t\rangle \longrightarrow \mathtt{T}}\ \mathtt{aT} \in t \qquad \frac{}{\langle \mathtt{a}, t\rangle \longrightarrow \mathtt{F}}\ \mathtt{aF} \in t$$

$$\frac{}{\langle \mathtt{T}, t\rangle \longrightarrow \mathtt{T}} \qquad \frac{}{\langle \mathtt{F}, t\rangle \longrightarrow \mathtt{F}}$$

$$\frac{\langle \mathtt{A}, t\rangle \longrightarrow \mathtt{T} \quad \langle \mathtt{B}, \mathtt{t}\rangle \longrightarrow \mathtt{T}}{\langle \mathtt{A} \wedge \mathtt{B}, t\rangle \longrightarrow \mathtt{T}} \qquad \frac{\langle \mathtt{A}, t\rangle \longrightarrow \mathtt{F} \quad \langle \mathtt{B}, \mathtt{t}\rangle \longrightarrow \mathtt{F}}{\langle \mathtt{A} \wedge \mathtt{B}, t\rangle \longrightarrow \mathtt{F}}$$

$$\frac{\langle \mathtt{A}, t\rangle \longrightarrow \mathtt{F} \quad \langle \mathtt{B}, \mathtt{t}\rangle \longrightarrow \mathtt{T}}{\langle \mathtt{A} \wedge \mathtt{B}, t\rangle \longrightarrow \mathtt{F}} \qquad \frac{\langle \mathtt{A}, t\rangle \longrightarrow \mathtt{T} \quad \langle \mathtt{B}, \mathtt{t}\rangle \longrightarrow \mathtt{F}}{\langle \mathtt{A} \wedge \mathtt{B}, t\rangle \longrightarrow \mathtt{F}}$$

The reader can fill in the rules describing the evaluation of disjunctions and negations. □

Exercise 5.1 Write down rules to define the set of binary trees with leaves in an alphabet Σ. □

Exercise 5.2 Prove the *soundness* of the proof rules for Boolean propositions above. That is, for any model, for each proof rule, show that if all the entailments of the premises hold, then so does the entailment of the conclusion. For example, to show the soundness of the rule

$$\frac{\Gamma \vdash \mathtt{A}}{\Gamma \vdash \mathtt{A} \vee \mathtt{B}}$$

requires showing

$$\text{if } \ \Gamma \models \mathtt{A}, \ \text{ then } \ \Gamma \models \mathtt{A} \wedge \mathtt{B}\ ,$$

where $\{\mathtt{A}_1, \cdots, \mathtt{A}_k\} \models \mathtt{A}$ means $\mathtt{A}_1 \wedge \cdots \wedge \mathtt{A}_k \models \mathtt{A}$. □

5.2 Inductively-defined sets

We are interested in the general problem of defining a set by rules of the kind we have seen in the examples.

In essence, an instance of a rule has the form of a pair (X/y) consisting of a set X, the *premises*, and a *conclusion* y. In general X might be empty or even infinite. When there is rule of the form $(\emptyset/y)$ we will

[1] You'll learn more about such proof systems in the 2nd year CS course "Logic and proof."

call y an *axiom*. We say a rule (X/y) is *finitary* when the set X is finite; then the rule will be of the form $(\{x_1, \ldots, x_n\}/y)$, possibly with empty premises.

All examples of the previous section are associated with their own *set* of rule instances. For example, for strings Σ^* the rule instances form the set

$$\{(\emptyset/\varepsilon)\} \cup \{(\{x\}/ax) \mid x \in \Sigma^* \ \&\ a \in \Sigma\} \ ,$$

consisting of all instantiations of the rules used in building up strings. We gave two forms of rules for generating the nonnegative integers $\mathbb{N}_0$. For the first the set of rule instances is

$$\{(\emptyset/0)\} \cup \{(\{n\}/n+1) \mid n \in \mathbb{N}_0\} \ ,$$

while for the second the set is

$$\{(\{0, \cdots, (n-1)\}/n) \mid n \in \mathbb{N}_0\} \ .$$

A set of rule instances R specifies a way to build a set. A particular rule instance (X/y) is intended to say that if all the elements of X are in the set then so is y. We look for the least set with this property. If it exists this should be the set inductively defined by the rules.

Suppose we are given a set of rule instances R. We say a set Q is *closed* under the rule instances R, or simply *R-closed*, iff for all rule instances (X/y)

$$X \subseteq Q \Rightarrow y \in Q \ .$$

In other words, a set is closed under the rule instances if whenever the premises of any rule instance lie in the set so does its conclusion. In particular, an R-closed set must contain all the axioms.

Assume a set of rule instances R. Consider the collection of all R-closed sets:

$$\{Q \mid Q \text{ is } R\text{-closed}\} \ .$$

This collection is nonempty as, for example, the set

$$\{y \mid \exists X.\ (X/y) \in R\}$$

is clearly R-closed. Thus we can form its intersection

$$I_R = \bigcap \{Q \mid Q \text{ is } R\text{-closed}\} \ .$$

The important fact is that I_R is itself R-closed. To see this argue as follows: Let $(X/y) \in R$. Suppose $X \subseteq I_R$. Let Q be any R-closed subset. Then $X \subseteq Q$ and consequently $y \in Q$, as Q is R-closed. Hence $y \in I_R$.

Summarising what we have just shown:

Proposition 5.3 *With respect to a set of rule instances R,*
(i) *I_R is R-closed, and*
(ii) *if Q is an R-closed set then $I_R \subseteq Q$.*

The set I_R is often described as the set *inductively defined* by R. Proposition 5.3 will supply us with a very useful proof principle for showing a property holds of all the elements of I_R. The earlier examples give an idea of how widespread inductively-defined sets are.

5.3 Rule induction

Suppose we wish to show a property $P(x)$ is true of all elements $x \in I_R$, the set inductively-defined by a set of rule instances R. The conditions (i) and (ii) in Proposition 5.3 above furnish a method. Define the set

$$Q = \{x \in I_R \mid P(x)\} \ .$$

The property $P(x)$ is true of all $x \in I_R$ iff $I_R \subseteq Q$. By condition (ii), to show $I_R \subseteq Q$ it suffices to show that Q is R-closed. This requires that for all rule instances (X/y) that

$$(\forall x \in X.\ x \in I_R \ \&\ P(x)) \Rightarrow (y \in I_R \ \&\ P(y)) \ .$$

But I_R is R-closed by (i), so this will follow precisely when

$$(\forall x \in X.\ x \in I_R\ \&\ P(x)) \Rightarrow P(y)\ .$$

We have obtained an important, general proof principle.

The principle of rule induction
Let I_R be inductively-defined by R. Then $\forall x \in I_R.\ P(x)$ if for all rule instances (X/y) in R,

$$(\forall x \in X.\ x \in I_R\ \&\ P(x)) \Rightarrow P(y)\ .$$

(The property $P(x)$ is called the *induction hypothesis*.)

[In other words to prove $\forall x \in I_R.\ P(x)$ it suffices to show that $\forall x \in X.\ P(x)$ implies $P(y)$ only for all rule instances (X/y) *with* $X \subseteq I_R$.]

Notice for rule instances of the form (X/y), with $X = \emptyset$, the condition in the statement of rule induction is equivalent to $P(y)$. Certainly then $\forall x \in X.\ x \in I_R\ \&\ P(x)$ is vacuously true because any x in $\emptyset$ satisfies $P(x)$—there are none.

Supposing the rule instances R are finitary, the statement of rule induction amounts to the following. For rule instances R, we have $\forall y \in I_R.\ P(y)$ iff for all axioms

$$\frac{}{y}$$

$P(y)$ is true, and for all rule instances

$$\frac{x_1, \ldots, x_n}{y}$$

if $x_k \in I_R\ \&\ P(x_k)$ is true for all the premises, when k ranges from 1 to n, then $P(y)$ is true of the conclusion.

The principle of rule induction is very useful to show a property is true of all the elements in an inductively-defined set. It has many well-known instances.

Examples: Refer to the examples of rules beginning this chapter.

Nonnegative integers $\mathbb{N}_0$: The rules $(\emptyset/0)$ and $(\{n\}/(n+1))$, for a number n, yield *mathematical induction* as a special case of rule induction.

The alternative rules $(\emptyset/0)$ and $(\{0, 1, \cdots, (n-1)\}/n)$, for a number n, yield *course-of-values induction*, the principle that says: A property $P(n)$ holds for all nonnegative numbers n iff for all $n \in \mathbb{N}_0$

$$(\forall m < n.\ P(m)) \Rightarrow P(n)\ .$$

Notice what happens when $n = 0$. Then there are no $m \in \mathbb{N}_0$ with $m < 0$, so the condition of the implication is vacuously true and the implication amounts to $P(0)$. You'll recall, course-of-values induction is useful when truth of a property at n can depend on its truth at earlier values other than just its immediate predecessor.

Strings Σ^*: With the rules for strings over an alphabet Σ rule induction specialises to this principle: a property $P(x)$ holds of all strings $x \in \Sigma^*$ iff

$$P(\varepsilon) \text{ and}$$
$$\forall a \in \Sigma,\ x \in \Sigma^*.\ P(x) \Rightarrow P(ax)\ .$$

(Essentially the same induction principle works for lists.)

Boolean propositions: With the rules for the syntax of Boolean propositions, rule induction specialises to *structural induction*.

Proof rules for Boolean propositions: Consider now R to be the the proof rules for entailment between Boolean propositions. Rule induction yields an intuitive way to establish properties of all derivable judgments. To show a property holds of all derivable judgments $\Gamma \vdash A$ it suffices to show the property is preserved in going from the premises to the conclusions of all the rules. With Exercise 5.2 in hand it follows that for any derivable sequent $\mathtt{\Gamma} \vdash \mathtt{A}$ we have that $\mathtt{\Gamma} \models \mathtt{A}$.

Evaluation of Boolean propositions: That evaluation can be captured by rules enables the use of rule induction in proving facts about the evaluation of Boolean propositions. Boolean propositions have few surprises but rule induction can support proofs of properties such as agreement with truth tables, termination and uniqueness of value. Such properties about the evaluation of Boolean propositions can be established by showing that they are preserved in going from premises to conclusions of rules.

Exercise 5.4 Justify the following "special principle of rule induction." Let I_R be defined by a set of rule instances R. Let $A \subseteq I_R$. Then $\forall a \in A.\ Q(a)$ if for all rule instances (X/y) in R, with $X \subseteq I_R$ and $y \in A$,

$$(\forall x \in X \cap A.\ Q(x)) \Rightarrow Q(y).$$

[Hint: Take property $P(x)$ to be

$$P(x) \text{ iff } (x \in A \Rightarrow Q(x))$$

in the statement of rule induction.] □

Exercise 5.5 Based on your rules for binary trees with leaves in Σ (*cf.* Exercise 5.1), write down the corresponding principle of rule induction. □

Exercise 5.6 The set of well-bracketed strings is the subset of strings over symbols [and] defined inductively as follows:

[ ] is well-bracketed;

if x is well-bracketed, then $[x]$ is well-bracketed;

if x and y are well-bracketed, then xy is well-bracketed.

State the principle of rule induction for well-bracketed strings. Show the number of left brackets [equals the number of right brackets] in any well-bracketed string. □

Exercise 5.7 A simple language is defined with symbols a and b. The grammar of this language has the rules:

- ab is a word;
- if ax is a word, then axx is a word (where x is any string of symbols);
- if $abbbx$ is a word, then ax is a word.

(i) Is $abbbbb$ a word? Either exhibit a derivation, or prove there isn't one.

(ii) Is $abbb$ a word? Either exhibit a derivation, or prove there isn't one.

(iii) Characterise the strings which are words. Prove your characterisation is correct. □

Exercise 5.8 The set S is defined to be the least subset of natural numbers $\mathbb{N}$ such that:

$1 \in S$;

if $n \in S$, then $3n \in S$;

if $n \in S$ and $n > 2$, then $(n-2) \in S$.

Show that $S = \{m \in \mathbb{N} \mid \exists r, s \in \mathbb{N} \cup \{0\}. \quad m = 3^r - 2s\}$. Deduce that S is the set of odd numbers. □

Exercise 5.9 Let I be a nonempty subset of the natural numbers $\mathbb{N} = \{1, 2, 3, \cdots\}$.
The set S is defined to be least subset of $\mathbb{N}$ such that

$I \subseteq S$, and
if $m, n \in S$ and $m < n$, then $(n - m) \in S$.

Define h to be the least member of S. This question guides you through to a proof that h coincides with the *highest common factor* of I, written $hcf(I)$, and defined to be the natural number with the properties that

$hcf(I)$ divides n for every element $n \in I$, and
if k is a natural number which divides n for every $n \in I$, then k divides $hcf(I)$.

(a) The set S may also be described as the least subset of $\mathbb{N}$ closed under certain rules. Describe the rules. Write down a principle of rule induction appropriate for the set S.

(b) Show by rule induction that $hcf(I)$ divides n for every $n \in S$.

(c) Let $n \in S$. Establish that

$$\text{if} \quad p \cdot h < n \quad \text{then} \quad (n - p \cdot h) \in S$$

for all nonnegative integers p.

(d) Show that h divides n for every $n \in S$. [Hint: suppose otherwise and derive a contradiction.]

(e) Why do the results of (b) and (d) imply that $h = hcf(I)$.

□

5.3.1 Transitive closure of a relation

Suppose that R is a relation on a set U. Its *transitive closure*, written R^+, is defined to be the least relation T such that T includes R and T is transitive, *i.e.*

$$R \subseteq T \text{ and}$$
$$(a, b) \in T \;\&\; (b, c) \in T \Rightarrow (a, c) \in T \; .$$

An element of R^+, *i.e.* a pair $(a, b) \in R^+$, is either in the original relation R or put in through enforcing transitivity. This is captured by the following rules for pairs in $U \times U$:

$$\frac{}{(a, b)} \;\; (a, b) \in R \qquad\qquad \frac{(a, b) \quad (b, c)}{(a, c)} \; . \tag{1}$$

In other words the transitive closure R^+ is inductively defined by these rules.

You may have seen another way to characterise the transitive closure of R. Define an R-chain from a to b to consist of pairs $(a_1, a_2), (a_2, a_3), \cdots, (a_{n-1}, a_n)$ in R with $a = a_1$ and $b = a_n$. Then, $(a, b) \in R^+$ iff there is an R-chain from a to b.
To see this, let

$$S = \{(a, b) \mid \text{ there is an } R\text{-chain from } a \text{ to } b\} \; .$$

First observe that

$$R \subseteq S \text{ and } (a, b) \in S \;\&\; (b, c) \in S \Rightarrow (a, c) \in S \; ,$$

the former because pairs in R form 1-link R-chains, the latter because we can concatenate two R-chains to get an R-chain. It follows that

$$R^+ \subseteq S \; .$$

To show equality, we need the converse too. This follows by mathematical induction on $n \in \mathbb{N}$ with induction hypothesis:

$$\text{for all } R\text{-chains } (a_1, a_2), (a_2, a_3), \cdots, (a_{n-1}, a_n) \text{ we have } (a_1, a_n) \in R^+ \; .$$

The basis of the induction, when $n = 1$, follows directly as $R \subseteq R^+$. The induction step uses, in addition, the transitivity of R^+.

Exercise 5.10 Show the transitive closure of a relation R is the least relation T such that:

$$R \subseteq T \text{ and } (a,b) \in R \;\&\; (b,c) \in T \Rightarrow (a,c) \in T \ .$$

□

One way to define the *reflexive, transitive closure* R^* of a relation R on a set U is as

$$R^* = \mathsf{id}_U \cup R^+ \ .$$

Exercise 5.11 Let R be a relation on a set U. Show that R^* is the least relation that includes R and is reflexive and transitive. □

Exercise 5.12 Let R be a relation on a set U. Define $R^0 = \mathsf{id}_U$, the identity relation on the set U, and $R^1 = R$ and inductively, assuming R^n is defined, define

$$R^{n+1} = R \circ R^n \ .$$

So, R^n is the relation $R \circ \cdots \circ R$, obtained by taking n compositions of R. Show the transitive closure of R is the relation

$$R^+ = \bigcup_{n \in \mathbb{N}_0} R^{n+1} \ ,$$

and that the transitive, reflexive closure of a relation R on X is the relation

$$R^* = \bigcup_{n \in \mathbb{N}_0} R^n \ .$$

□

Exercise 5.13 Show $(R \cup R^{-1})^*$ is an equivalence relation. Show that it is the least equivalence relation including R. Show $R^* \cup (R^{-1})^*$ need not be an equivalence relation. □

Exercise 5.14 Show that the least equivalence relation containing two equivalence relations R and S on the same set is $(R \cup S)^+$. □

5.4 Derivation trees

Another important way to understand inductively-defined sets as generated by rules is via the notion of a *derivation tree*, or *derivation*. An inductively-defined set consists of precisely those elements for which there is a derivation. In this section we'll assume a set of rule instances R which are all finitary, though the ideas generalise straightforwardly to infinitary rules.

We are familiar with informal examples of derivations from games, like chess, draughts or bridge, where it's usual to discuss a particular play of a game leading up to a winning position. In the idealised view of mathematics, as a formal game of deriving theorems from rules, a proof is a derivation of a theorem from the rules of mathematics. As the example of proofs in mathematics makes clear, derivations can have much more informative structure than the, often simpler, things they are intended to derive; finding proofs in mathematics is highly nontrivial because the form an assertion takes rarely determines the structure of its possible proofs.

As a simple example here's a derivation for the Boolean proposition $\neg\mathtt{a} \wedge (\mathtt{b} \vee \mathtt{T})$ using the rules of syntax for forming Boolean propositions:

$$\dfrac{\dfrac{\dfrac{}{\mathtt{a}}}{\neg\mathtt{a}} \qquad \dfrac{\dfrac{}{\mathtt{b}} \qquad \dfrac{}{\mathtt{T}}}{\mathtt{b} \vee \mathtt{T}}}{\neg\mathtt{a} \wedge (\mathtt{b} \vee \mathtt{T})}$$

It has the form of a tree with the conclusion at the root and with axioms at the leaves. It is built by stacking rules together, matching conclusions with premises. A more interesting derivation is the following using the

proof rules of Boolean propositions:[2]

$$\dfrac{\dfrac{\dfrac{}{\mathtt{A} \wedge \neg \mathtt{A} \vdash \mathtt{A} \wedge \neg \mathtt{A}}}{\mathtt{A} \wedge \neg \mathtt{A} \vdash \mathtt{A}} \qquad \dfrac{\dfrac{}{\mathtt{A} \wedge \neg \mathtt{A} \vdash \mathtt{A} \wedge \neg \mathtt{A}}}{\mathtt{A} \wedge \neg \mathtt{A} \vdash \neg \mathtt{A}}}{\dfrac{\mathtt{A} \wedge \neg \mathtt{A} \vdash \mathtt{F}}{\vdash \neg(\mathtt{A} \wedge \neg \mathtt{A})}}$$

The example of proofs makes clear that, in general, there need not be a unique derivation associated with a particular conclusion.

Exercise 5.15 Identify the proof rules used in building the above derivation of $\vdash \neg(\mathtt{A} \wedge \neg \mathtt{A})$. Derive the sequents:

(i) $\neg \mathtt{A} \wedge \neg \mathtt{B} \vdash \neg(\mathtt{A} \vee \mathtt{B})$,

(i) $\neg(\mathtt{A} \vee \mathtt{B}) \vdash \neg \mathtt{A} \wedge \neg \mathtt{B}$, [Hard]

(iii) $\vdash \mathtt{A} \vee \neg \mathtt{A}$. [Hard]

□

These examples show how rules determine derivations. The idea is that rules lift to rules generating derivations; we build a new derivation by stacking derivations on top of matching premises of a rule. A derivation of an element y takes the form of a tree which is either an instance of an axiom

$$\dfrac{}{y}$$

or of the form

$$\dfrac{\dfrac{\vdots}{x_1}, \ldots, \dfrac{\vdots}{x_n}}{y}$$

which includes derivations of $x_1, \ldots, x_n$, the premises of a rule with conclusion y. In such a derivation we think of $\dfrac{\vdots}{x_1}, \cdots, \dfrac{\vdots}{x_n}$ as subderivations of the larger derivation of y.

In set notation, an *R-derivation* of y is either a rule

$$(\emptyset / y)$$

or a pair

$$(\{d_1, \cdots, d_n\} / y)$$

where $(\{x_1, \cdots, x_n\}/y)$ is a rule and d_1 is an R-derivation of x_1, ..., and d_n is an R-derivation of x_n.

As the formulation makes clear, the set of all R-derivations is inductively-defined. In this case rule induction specialises to another useful proof principle.

Induction on derivations
Let $P(d)$ be a property of R-derivations d. Then, $P(d)$ holds for all R-derivations d iff for all rule instances $(\{x_1, \cdots, x_n\}/y)$ in R and R-derivations d_1 of x_1, ..., and d_n of x_n,

$$P(d_1) \ \& \ \cdots \ \& \ P(d_n) \Rightarrow P(\{d_1, \cdots, d_n\}/y) \ .$$

(As usual, the property $P(d)$ is called the induction hypothesis.)

[2] As is coventional for such proofs we don't write the set brackets for the set on the left of a sequent and write nothing for the empty set.

In practice it is easier to apply induction on derivations than its rather formal statement might suggest. A proof by induction on derivations splits into cases according to the last rule in the derivation. In each case, it is required to show that a derivation

$$\frac{\frac{\vdots}{x_1}, \ldots, \frac{\vdots}{x_n}}{y}$$

inherits a desired property from its subderivations $\frac{\vdots}{x_1}, \cdots, \frac{\vdots}{x_n}$. Induction on derivations is illustrated in the proof of the following fundamental result.

Theorem 5.16 *An element $y \in I_R$ iff there is an R-derivation of y.*

Proof.
'only if': Consider the set

$$D = \{y \mid \text{ there is an } R\text{-derivation of } y\} \ .$$

The set D is R-closed as given any rule (X/y) and derivations for all its premises X we can construct a derivation of y. Thus $I_R \subseteq D$.
'if': A simple induction on derivations shows that $y \in I_R$ for any derivation of y. The argument: Take as induction hypothesis the property that holds of a derivation of y precisely when $y \in I_R$. Consider any rule $(\{x_1, \cdots, x_n\}/y)$ and derivations d_1 of x_1, ..., and d_n of x_n for which $x_1 \in I_R$, ..., and $x_n \in I_R$. Then, as I_R is R-closed, $y \in I_R$. □

It can sometimes be easier to attach a property to a derivation, which encodes the whole history of how an element came to be in an inductively-defined set, than to an element standing alone. If the rules are such that the conclusions determine the rules uniquely there is no advantage (or disadvantage) in using induction on derivations over straight rule induction.

Exercise 5.17 Go through this section generalising the definition of derivation and the results to the situation where the rules are not necessarily finitary. □

5.5 Least fixed points

Given a set of rule instances R, we defined I_R as the intersection of all R-closed sets. In a sense this is a way of restricting sets to obtain I_R. It perhaps doesn't match our first intuitions about how rules generate a set: start from the axioms and repeatedly apply the rules and only put something in the set if it can be derived in this way. There are alternative generative ways to construct the set I_R inductively defined by rules with instances R. The first leads us to an understanding of inductively-defined sets as least fixed points.

Assume a set of rule instances R. Given a set B, then

$$\widehat{R}(B) = \{y \mid \exists X \subseteq B.\ (X/y) \in R\}$$

is a set. Intuitively, the set $\widehat{R}(B)$ is got by applying the rule instances R to the set B. The rule instances R determines an operation $\widehat{R}$ on sets: given a set B it results in a set $\widehat{R}(B)$. Use of the operation $\widehat{R}$ gives another way of saying a set is R-closed, one that follows directly from the definitions.

Proposition 5.18 *A set B is R-closed iff $\widehat{R}(B) \subseteq B$.*

The operation $\widehat{R}$ provides a way of building up the set I_R which we will describe when the rule instances R are finitary. The operation $\widehat{R}$ is *monotonic* in the sense that

$$A \subseteq B \Rightarrow \widehat{R}(A) \subseteq \widehat{R}(B) \ .$$

If we repeatedly apply $\widehat{R}$ to the empty set $\emptyset$ we obtain a sequence of sets:

$$\begin{aligned} A_0 &= \widehat{R}^0(\emptyset) = \emptyset \ , \\ A_1 &= \widehat{R}^1(\emptyset) = \widehat{R}(\emptyset) \ , \\ A_2 &= \widehat{R}(\widehat{R}(\emptyset)) = \widehat{R}^2(\emptyset) \ , \\ &\vdots \\ A_n &= \widehat{R}^n(\emptyset) \ , \\ &\vdots \end{aligned}$$

The set A_1 consists of all the axioms, and in general the set A_{n+1} is all things which immediately follow by rule instances which have premises in A_n. Clearly $\emptyset \subseteq \widehat{R}(\emptyset)$, *i.e.* $A_0 \subseteq A_1$. By the monotonicity of $\widehat{R}$ we obtain $\widehat{R}(A_0) \subseteq \widehat{R}(A_1)$, *i.e.* $A_1 \subseteq A_2$. Similarly we obtain $A_2 \subseteq A_3$ *etc.*. Thus the sequence forms a chain

$$A_0 \subseteq A_1 \subseteq \cdots \subseteq A_n \subseteq \cdots \ .$$

Taking $A = \bigcup_{n\in\mathbb{N}_0} A_n$, we have:

Theorem 5.19 *Provided the rule instances R are finitary,*

(i) *A is R-closed,*

(ii) *$\widehat{R}(A) = A$,*

(iii) *A is the least R-closed set.*

Proof.
(i) Suppose $(X/y) \in R$ with $X \subseteq A$. Recall $A = \bigcup_n A_n$ is the union of an increasing chain of sets. As X is a finite set there is some n such that $X \subseteq A_n$. (The set X is either empty, whence $X \subseteq A_0$, or of the form $\{x_1, \ldots, x_k\}$. In the latter case, we have $x_1 \in A_{n_1}, \cdots, x_k \in A_{n_k}$ for some $n_1, \ldots, n_k$. Taking n bigger than all of $n_1, \ldots, n_k$ we must have $X \subseteq A_n$ as the sequence $A_0, A_1, \ldots, A_n, \ldots$ is increasing.) As $X \subseteq A_n$ we obtain $y \in \widehat{R}(A_n) = A_{n+1}$. Hence $y \in \bigcup_n A_n = A$. Thus A is closed under R.

(ii) By Proposition 5.18, since the set A is R-closed, we know that $\widehat{R}(A) \subseteq A$. We require the converse inclusion. Suppose $y \in A$. Then $y \in A_n$ for some $n > 0$. Thus $y \in \widehat{R}(A_{n-1})$. This means there is some $(X/y) \in R$ with $X \subseteq A_{n-1}$. But $A_{n-1} \subseteq A$ so $X \subseteq A$ with $(X/y) \in R$, giving $y \in \widehat{R}(A)$. We have established the required converse inclusion, $A \subseteq \widehat{R}(A)$. Hence $\widehat{R}(A) = A$.

(iii) We need to show that if B is another R-closed set then $A \subseteq B$. Suppose B is closed under R. Then $\widehat{R}(B) \subseteq B$. We show by mathematical induction that for all $n \in \mathbb{N}_0$

$$A_n \subseteq B \ .$$

The basis of the induction $A_0 \subseteq B$ is obviously true as $A_0 = \emptyset$. To show the induction step, assume $A_n \subseteq B$. Then

$$A_{n+1} = \widehat{R}(A_n) \subseteq \widehat{R}(B) \subseteq B \ ,$$

using the facts that $\widehat{R}$ is monotonic and that B is R-closed. □

Notice the essential part played in the proof of (i) by the fact that the rules are finitary.This restriction is needed—see the remark concluding this section.

Now (ii) in Theorem 5.19 says precisely that I_R is a fixed point of $\widehat{R}$. Moreover, (iii) implies that I_R is the *least fixed point* of $\widehat{R}$, *i.e.*

$$\widehat{R}(B) = B \Rightarrow I_R \subseteq B \ ,$$

because if any other set B is a fixed point it is closed under R, so $I_R \subseteq B$ by Proposition 5.3. The set I_R, defined by the rule instances R, is the least fixed point, written *fix* $(\widehat{R})$, obtained by the construction

$$\mathit{fix}\ (\widehat{R}) =_{def} \bigcup_{n\in\mathbb{N}_0} \widehat{R}^n(\emptyset) \ .$$

Exercise 5.20 Let U be a set. A function $\varphi : \mathcal{P}(U) \to \mathcal{P}(U)$ is *continuous* iff φ is monotonic, *i.e.* for all subsets S, S' of U,

$$S \subseteq S' \Rightarrow \varphi(S) \subseteq \varphi(S')$$

and, for any increasing chain of subsets of U $S_0 \subseteq S_1 \subseteq \cdots \subseteq S_n \subseteq \cdots$,

$$\varphi(\bigcup_{n\in\mathbb{N}_0} S_n) = \bigcup_{n\in\mathbb{N}_0} \varphi(S_n) \ .$$

Let R be a set of finitary rule instances of which all the conclusions lie in U. Show that the function $\widehat{R} : \mathcal{P}(U) \to \mathcal{P}(U)$ is continuous. Show that any continuous function $f : \mathcal{P}(U) \to \mathcal{P}(U)$ has a least fixed point. □

Remark There is a generalisation of Theorem 5.19 in which it is not required that the rules are finitary. But this involves having more than the $\mathbb{N}_0$ approximations $A_0, A_1, \cdots, A_n, \cdots$, and 'mysterious' ordinals like $\omega + 1, \omega + 2, \cdots$ to continue counting beyond $\mathbb{N}_0 = \omega$, so is outside the scope of this course. □

5.6 Tarski's fixed point theorem

Let U be a set. Then its powerset $\mathcal{P}(U)$ forms a partial order in which the order is that of inclusion $\subseteq$. We examine general conditions under which functions $\varphi : \mathcal{P}(U) \to \mathcal{P}(U)$ have canonical fixed points.

We provide a proof of Tarski's fixed point theorem, specialised to powersets. This concerns fixed points of functions $\varphi : \mathcal{P}(U) \to \mathcal{P}(U)$ which are *monotonic*, *i.e.* such that

$$S \subseteq S' \Rightarrow \varphi(S) \subseteq \varphi(S') \ ,$$

for $S, S' \in \mathcal{P}(U)$. Such monotonic functions have least (=minimum) and greatest (=maximum) fixed points.

Theorem 5.21 *(Tarski's theorem for minimum fixed points)*
Let $\mathcal{P}(U)$ be a powerset. Let $\varphi : \mathcal{P}(U) \to \mathcal{P}(U)$ be a monotonic function. Define

$$m = \bigcap \{S \subseteq U \mid \varphi(S) \subseteq S\}.$$

Then m is a fixed point of φ and the least prefixed point of φ, i.e. if $\varphi(S) \subseteq S$ then $m \subseteq S$. (When $\varphi(S) \subseteq S$ the set S is called a prefixed point *of φ.)*

Proof. Write $X = \{S \subseteq U \mid \varphi(S) \subseteq S\}$. As above, define $m = \bigcap X$. Let $S \in X$. Certainly $m \subseteq S$. Hence $\varphi(m) \subseteq \varphi(S)$ by the monotonicity of φ. But $\varphi(S) \subseteq S$ because $S \in X$. So $\varphi(m) \subseteq S$ for any $S \in X$. It follows that $\varphi(m) \subseteq \bigcap X = m$. This makes m a prefixed point and, from its definition, it is clearly the least one. As $\varphi(m) \subseteq m$ we obtain $\varphi(\varphi(m)) \subseteq \varphi(m)$ from the monotonicity of φ. This ensures $\varphi(m) \in X$ which entails $m \subseteq \varphi(m)$. Thus $\varphi(m) = m$. We conclude that m is indeed a fixed point and is the least prefixed point of φ. □

The proof of Tarski's theorem for minimum fixed points only makes use of the partial-order properties of the $\subseteq$ relation on $\mathcal{P}(U)$ and in particular that there is an intersection operation $\bigcap$. (In fact, Tarski's theorem applies equally well to a complete lattice with an abstract partial order and greatest lower bound.) Replacing the roles of the order $\subseteq$ and intersection $\bigcap$ by the converse relation $\supseteq$ and union $\bigcup$ we obtain a proof of the dual result for maximum fixed points.

Theorem 5.22 *(Tarski's theorem for maximum fixed points)*
Let $\mathcal{P}(U)$ be a powerset. Let $\varphi : \mathcal{P}(U) \to \mathcal{P}(U)$ be a monotonic function. Define

$$M = \bigcup \{S \subseteq U \mid S \subseteq \varphi(S)\}.$$

Then M is a fixed point of φ and the greatest postfixed point of φ, i.e. if $S \subseteq \varphi(S)$ then $S \subseteq M$. (When $S \subseteq \varphi(S)$ the set S is called a postfixed point *of φ.)*

Notation: The minimum fixed point is traditionally written $\mu X.\varphi(X)$, and the maximum fixed point as $\nu X.\varphi(X)$. Minimum and maximum fixed points are very important in verification, in particular in *model checking*—the automatic verification that a process satisfies a property.

Tarski's theorem for minimum fixed points provides another way to understand sets inductively defined by rules. Assuming that all the rule instances R have conclusions in the set U, we can turn R into a monotonic function $\widehat{R} : \mathcal{P}(U) \to \mathcal{P}(U)$, where for $S \in \mathcal{P}(U)$

$$\widehat{R}(S) = \{y \mid \exists X \subseteq S.\ (X/y) \in R\}\ .$$

Prefixed points of $\widehat{R}$ coincide with R-closed subsets of U and the least fixed point of φ_R with the set I_R inductively defined by the rule instances R.

Sets defined as maximum fixed points are often called coinductively defined sets.

Exercise 5.23 Supply a direct proof of Tarski's theorem (Theorem 5.22 above) for maximum fixed points. □

Exercise 5.24 Refer to Exercise 3.20 which defines the bisimilarity relation $\sim$ for a directed graph $(P, \longrightarrow)$. Define

$$\varphi : \mathcal{P}(P \times P) \to \mathcal{P}(P \times P)$$

—so φ is a function from relations on P to relations on P—by:

$p\ \varphi(R)\ q$ iff

- $\forall p' \in P.\ p \longrightarrow p' \Rightarrow \exists q' \in P.\ q \longrightarrow q'\ \&\ p'\,R\,q'$, and
- $\forall q' \in P.\ q \longrightarrow q' \Rightarrow \exists p' \in P.\ p \longrightarrow p'\ \&\ p'\,R\,q'$.

Show that φ is monotonic, has postfixed points precisely the bisimulations on P, and that the bisimilarity relation $\sim$ coincides with its maximum fixed point. □

Exercise 5.25 Streams, or infinite lists, of values in a set V can be represented as functions

$$V^\infty =_{def} (\mathbb{N}_0 \to V)\,.$$

The head and tail of a stream L are defined by

$$\begin{aligned} head(L) &= L(0) \\ tail(L)(n) &= L(n+1) \text{ for } n \in \mathbb{N}_0\,. \end{aligned}$$

Define a function $\varphi : \mathcal{P}(V^\infty \times V^\infty) \to \mathcal{P}(V^\infty \times V^\infty)$ by taking

$$L\ \varphi(R)\ M \ \text{ iff } \ head(L) = head(M) \text{ and } \ tail(L)\ R\ tail(M)\,.$$

Show that φ is monotonic. A stream-bisimulation is a postfixed point of φ, *i.e.* a relation R between streams such that

$$\text{If } \ L\ R\ M \ \text{ then } \ head(L) = head(M) \text{ and } \ tail(L)\ R\ tail(M)\,.$$

Show by mathematical induction that, for any stream bisimulation R,

$$\text{if } L\ R\ M \text{ then } L = M\ .$$

Deduce that the greatest fixed point of φ is id_{V^∞}. Deduce the principle of coinduction for : Streams $L, M \in V^\infty$ are equal if there is a stream-bisimulation relating them. □

Exercise 5.26 Let $\varphi : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ be the function given by

$$\varphi(U) = \{3n/2 \mid n \in U\ \&\ n \text{ is even}\} \cup \{n \mid n \in U\ \&\ n \text{ is odd}\}\ .$$

(i) Show φ is monotonic with respect to $\subseteq$.

(ii) Suppose that $U \subseteq \varphi(U)$, *i.e.* U is a postfixed point of φ. Show that

$$n \in U\ \&\ n \text{ is even} \Rightarrow 2n/3 \in U\ .$$

Deduce that all members of U are odd. [Hint: Assume there is an even member of U, so a least even member of U, to derive a contradiction.]

(iii) Deduce that the maximum fixed point of φ is the set of all odd numbers.

(iv) Characterise the prefixed points of φ. What is the minimum fixed point of φ?

□

Chapter 6

Well-founded induction

This chapter introduces the powerful general proof principle of well-founded induction and its associated method of definition called well-founded recursion. They are based on the concept of a well-founded relation. Well-founded induction has many applications but is especially important for defining and proving properties of terminating programs.

6.1 Well-founded relations

Mathematical and structural induction are special cases of a general and powerful proof principle called well-founded induction. In essence structural induction works because breaking down an expression into subexpressions cannot go on forever, eventually it must lead to atomic expressions which cannot be broken down any further. If a property fails to hold of any expression then it must fail on some minimal expression which when it is broken down yields subexpressions, all of which satisfy the property. This observation justifies the principle of structural induction: to show a property holds of all expressions it is sufficient to show that property holds of an arbitrary expression if it holds of all its subexpressions. Similarly with the natural numbers, if a property fails to hold of all natural numbers then there has to be a smallest natural number at which it fails. The essential feature shared by both the subexpression relation and the predecessor relation on natural numbers is that they do not give rise to infinite descending chains. This is the feature required of a relation if it is to support well-founded induction.

Definition: A *well-founded relation* is a binary relation $\prec$ on a set A such that there are no infinite descending chains $\cdots \prec a_i \prec \cdots \prec a_1 \prec a_0$. When $a \prec b$ we say a is a *predecessor* of b.

Note a well-founded relation is necessarily *irreflexive i.e.*, for no a do we have $a \prec a$, as otherwise there would be the infinite descending chain $\cdots \prec a \prec \cdots \prec a \prec a$. We shall generally write $\preceq$ for the reflexive closure of the relation $\prec$, *i.e.*

$$a \preceq b \iff a = b \text{ or } a \prec b.$$

(A relation $\prec$ for which $\preceq$ is a total order is traditionally called a *well-order*.)

Sometimes one sees an alternative definition of well-founded relation, in terms of minimal elements.

Proposition 6.1 *Let $\prec$ be a binary relation on a set A. The relation $\prec$ is well-founded iff any nonempty subset Q of A has a minimal element, i.e. an element m such that*

$$m \in Q \;\&\; \forall b \prec m.\ b \notin Q.$$

Proof.
"if": Suppose every nonempty subset of A has a minimal element. If $\cdots \prec a_i \prec \cdots \prec a_1 \prec a_0$ were an infinite descending chain then the set $Q = \{a_i \mid i \in \mathbb{N}_0\}$ would be nonempty without a minimal element, a contradiction. Hence $\prec$ is well-founded.
"only if": To see this, suppose Q is a nonempty subset of A. Construct a chain of elements as follows. Take a_0 to be any element of Q. Inductively, assume a chain of elements $a_n \prec \cdots \prec a_0$ has been constructed inside Q. Either there is some $b \prec a_n$ such that $b \in Q$ or there is not. If not , then stop the construction. Otherwise take $a_{n+1} = b$. As $\prec$ is well-founded the chain $\cdots \prec a_i \prec \cdots \prec a_1 \prec a_0$ cannot be infinite. Hence it is finite, of the form $a_n \prec \cdots \prec a_0$ with $\forall b \prec a_n.\ b \notin Q$. Take the required minimal element m to be a_n.□

Exercise 6.2 Let $\prec$ be a well-founded relation on a set B. Prove

(i) its transitive closure $\prec^+$ is also well-founded,

(ii) its reflexive, transitive closure $\prec^*$ is a partial order.

□

6.2 Well-founded induction

Well-founded relations support an important proof principle.

The principle of well-founded induction
Let $\prec$ be a well founded relation on a set A. To show $\forall a \in A.\ P(a)$ it suffices to prove that for all $a \in A$

$$[\forall b \prec a.\ P(b)] \Rightarrow P(a)\ .$$

The principle reduces showing that a property (the induction hypothesis) holds globally to showing that the property is preserved locally by the well founded relation.

We now prove the principle. The proof rests on the observation, Proposition 6.1, that any nonempty subset Q of a set A with a well-founded relation $\prec$ has a minimal element. To justify the principle, we assume $\forall a \in A.\ ([\forall b \prec a.\ P(b)] \Rightarrow P(a))$ and produce a contradiction by supposing $\neg P(a)$ for some $a \in A$. Then, as we have observed, there must be a minimal element m of the set $\{a \in A \mid \neg P(a)\}$. But then $\neg P(m)$ and yet $\forall b \prec m.\ P(b)$, which contradicts the assumption.

Example: If we take the relation $\prec$ to be the predecessor relation

$$n \prec m \text{ iff } m = n + 1$$

on the non-negative integers the principle of well-founded induction specialises to mathematical induction. □

Example: If we take $\prec$ to be the "strictly less than" relation $<$ on the non-negative integers, the principle specialises to course-of-values induction: To show $P(n)$ for all nonnegative integers n, it suffices to show

$$(\forall m < n.\ P(m)) \Rightarrow P(n)$$

for all nonnegative integers n. □

Example: If we take $\prec$ to be the relation between expressions such that $a \prec b$ holds iff a is an immediate subexpression of b we obtain the principle of structural induction as a special case of well-founded induction. □

Proposition 6.1 provides an alternative to proofs by the principle of well-founded induction. Suppose A is a well-founded set. Instead of using well-founded induction to show every element of A satisfies a property, we can consider the subset of A for which the property fails, *i.e.* the subset Q of counterexamples. By Proposition 6.1, to show Q is $\emptyset$ it is sufficient to show that Q cannot have a minimal element. This is done by obtaining a contradiction from the assumption that there is a minimal element in Q. Whether to use this approach or the principle of well-founded induction is largely a matter of taste, though sometimes, depending on the problem, one approach can be more direct than the other.

A special instance of Proposition 6.1 is well-known to be equivalent to mathematical induction. It is the principle that every nonempty subset of natural numbers has a least element.

Exercise 6.3 For a suitable well-founded relation on strings, use the "no counterexample" approach described above to show there is no string u which satisfies $au = ub$ for two distinct symbols a and b. □

Well-founded induction is the most important principle in proving the termination of programs. Uncertainties about termination arise because of loops or recursions in a program. If it can be shown that execution of a loop or recursion in a program decreases the value in a well-founded set then execution must eventually terminate.

6.3 Building well-founded relations

Applying the principle of well-founded induction often depends on a judicious choice of well-founded relation.

6.3.1 Fundamental well-founded relations

We have already made use of well-founded relations like that of proper subexpression on syntactic sets, or $<$ on natural numbers. More generally, in Section 5.4 we saw that any set inductively-defined by rule instances R was associated with a set of R-derivations. The subderivation relation is a well-founded relation on R-derivations—see Exercise 6.4 below. In many cases each element of an inductively-defined set I_R has a unique derivation (*e.g.* the case for $\mathbb{N}_0$ and simple syntax, such as that of Boolean propositions). Then the well-founded relation on R-derivations transfers directly to a well-founded relation on I_R.

Exercise 6.4 Let R be a collection of finitary rule instances. For R-derivations d, d' define

$$d' \prec d \text{ iff } \exists D, y.\ d = (D/y)\ \&\ d' \in D\ .$$

By using induction on derivations with a suitable induction hypothesis, show $\prec$ is well-founded. □

Here are some ways to construct further well-founded relations. Recall that we use $x \preceq y$ to mean ($x \prec y$ or $x = y$).

6.3.2 Transitive closure

If $\prec$ is well-founded relation on A, then so is its transitive closure $\prec^+$. Clearly any infinite descending chain

$$\cdots \prec^+ a_n \prec^+ \cdots \prec^+ a_1 \prec^+ a_0$$

with respect to $\prec^+$ would induce an infinite descending chain with respect to $\prec$. (This was part of an earlier exercise!)

6.3.3 Product

If $\prec_1$ is well-founded on A_1 and $\prec_2$ is well-founded on A_2 then taking

$$(a_1, a_2) \preceq (a_1', a_2') \Leftrightarrow_{def} a_1 \preceq_1 a_1' \text{ and } a_2 \preceq_2 a_2'$$

determines a relation $\prec = (\preceq \setminus \mathsf{id}_{A_1 \times A_2})$ in $A_1 \times A_2$ called the product relation:

Proposition 6.5 *The product relation of well-founded relations is well-founded.*

Proof. Suppose $\prec_1$ is well-founded on A_1 and $\prec_2$ is well-founded on A_2. Assume their product relation $\prec$ is not well-founded, *i.e.* that there is an infinite descending chain

$$\cdots \prec (x_n, y_n) \prec \cdots \prec (x_1, y_1) \prec (x_0, y_0)\ .$$

But then, from the definition of the product relation $\prec$, either

$$\cdots \prec_1 x_{n_k} \prec_1 \cdots \prec_1 x_{n_1} \prec_1 x_{n_0}$$

or

$$\cdots \prec_2 y_{n_k} \prec_2 \cdots \prec_2 y_{n_1} \prec_2 y_{n_0}\ ,$$

which contradicts the well-foundedness of $\prec_1$ and $\prec_2$. □

We'll see applications of the product of well-founded relations in the next section. However product relations are not as generally applicable as those produced by lexicographic products.

6.3.4 Lexicographic products

Let $\prec_1$ be well-founded on A_1 and $\prec_2$ be well-founded on A_2. Define their lexicographic product by

$$(a_1, a_2) \prec_{lex} (a_1', a_2') \text{ iff } a_1 \prec_1 a_1' \text{ or } (a_1 = a_1' \;\&\; a_2 \prec_2 a_2') \ .$$

Proposition 6.6 *The lexicographic product of well-founded relations is well-founded.*

Proof. Suppose $\prec_1$ is well-founded on A_1 and $\prec_2$ is well-founded on A_2. Assume their lexicographic product $\prec_{lex}$ is not well-founded, *i.e.* that there is an infinite descending chain

$$\cdots \prec (x_n, y_n) \prec \cdots \prec (x_1, y_1) \prec (x_0, y_0) \ .$$

From the definition of the lexicographic relation $\prec_{lex}$

$$\cdots \preceq_1 x_n \preceq_1 \cdots \preceq_1 x_1 \preceq_1 x_0 \ .$$

But $\prec_1$ is well-founded so from some stage on, say $m \geq n$, this chain is constant. But then from the definition of the lexicographic relation $\prec$,

$$\cdots \prec_2 y_{n+i} \prec_2 \cdots \prec_2 y_{n+1} \prec_2 y_n \ ,$$

which contradicts the well-foundedness of $\prec_2$. □

Exercise 6.7 Let $\prec$ be a well-founded relation on a set X such that $\preceq$ is a total order. Show it need not necessarily make the set

$$\{x \in X \mid x \prec y\}$$

finite for all $y \in X$.
[Recall a total order is a partial order $\leq$ such that $x \leq y$ or $y \leq x$ for all its elements x, y. Hint: Consider the lexicographic product of $<$ and $<$ on $\mathbb{N}_0 \times \mathbb{N}_0$.] □

6.3.5 Inverse image

Let $f : A \to B$ be a function and $\prec_B$ a well-founded relation on B. Then $\prec_A$ is well-founded on A where

$$a \prec_A a' \Leftrightarrow_{def} f(a) \prec_B f(a')$$

for $a, a' \in A$.

Exercise 6.8 Show the inverse image of a well-founded relation is a well-founded relation. □

6.4 Applications

6.4.1 Euclid's algorithm for hcf

We can use well-founded induction to show the correctness of Euclid's algorithm for calculating the highest common factor (hcf) of a pair of natural numbers.[1] One way to formulate Euclid's algorithm is through a reduction relation $\longrightarrow_E$ on $\mathbb{N} \times \mathbb{N}$ defined as follows:

$$(m, n) \longrightarrow_E (m, n - m) \quad \text{if } m < n \ ,$$

$$(m, n) \longrightarrow_E (m - n, n) \quad \text{if } n < m \ .$$

So (m, n) reduces to $(m, n - m)$ if $m < n$ and to $(m - n, n)$ if $n < m$. Notice there is no reduction when $m = n$; in this case the reduction terminates.

It is easy to check that the following properties hold for the hcf of natural numbers:

Proposition 6.9

[1] Another name for highest common factor is *greatest common divisor* (gcd).

(a) $hcf(m,n) = hcf(m,n-m)$ *if* $m < n$,

(b) $hcf(m,n) = hcf(m-n,n)$ *if* $n < m$,

(c) $hcf(m,m) = m$.

Proof. The highest common factor of natural numbers m and n, $hcf(m,n)$, is characterised by:

(i) $hcf(m,n)$ divides m and n;

(ii) if k divides m and n, then k divides $hcf(m,n)$.

In all cases the proof proceeds by showing any divisor of the left is also a divisor of the right and *vice versa*; two natural numbers with the same divisors must be equal. As an example we show (a) $hcf(m,n) = hcf(m,n-m)$ assuming $m < n$. Suppose k divides the lhs $hcf(m,n)$. Then k certainly divides m and n by (i), and so divides m and $n-m$. Thus k divides the rhs $hcf(m,n-m)$ by (ii). Suppose now that k divides the rhs $hcf(m,n-m)$. Then k divides m and $n-m$ by (i). It follows that k divides m and n, and so the lhs $hcf(m,n)$ by (ii). □

Euclid's reduction terminates with the hcf of the natural numbers it starts with:

Theorem 6.10 *For all* $m, n \in \mathbb{N}$,

$$(m,n) \longrightarrow_E^* (hcf(m,n), hcf(m,n)) \ .$$

Proof. Let $\prec \subseteq \mathbb{N} \times \mathbb{N}$ be the well-founded relation constructed as the product of $<$ and $<$ on $\mathbb{N}$. Take

$$P(m,n) \Leftrightarrow_{def} \ (m,n) \longrightarrow_E^* (hcf(m,n), hcf(m,n))$$

as the induction hypothesis. We prove $P(m,n)$ for all $m, n \in \mathbb{N}$ by well-founded induction.

Let $(m,n) \in \mathbb{N} \times \mathbb{N}$. Assume $P(m',n')$ for all $(m',n') \prec (m,n)$. Consider the cases:

Case $m < n$. In this case $(m,n) \longrightarrow_E (m,n-m)$ and because $P(m,n-m)$ by the induction hypothesis,

$$(m,n-m) \longrightarrow_E^* (hcf(m,n-m), hcf(m,n-m)) \ .$$

Hence

$$(m,n) \longrightarrow_E^* (hcf(m,n-m), hcf(m,n-m)) \ ,$$

by the properties of the reflexive transitive closure $\longrightarrow_E^*$. Also $hcf(m,n) = hcf(m,n-m)$. Thus $P(m,n)$ in this case.

Case $n < m$. This case is very similar.

Case $m = n$. In this case

$$(m,n) \longrightarrow_E^* (m,n)$$

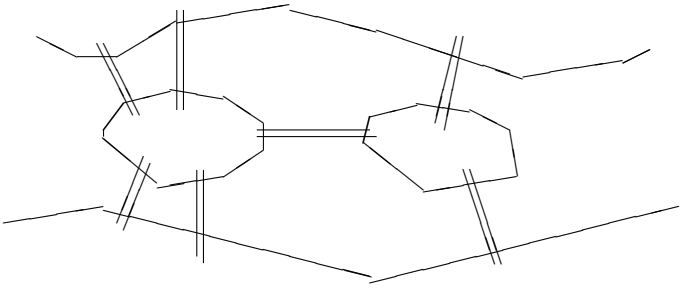
as $\longrightarrow_E^*$ is reflexive. Also $hcf(m,n) = m = n$. Thus $P(m,n)$ in this case.

In all possible cases for (m,n) we can derive $P(m,n)$ from the assumption that $P(m',n')$ holds for all $\prec$-predecessors (m',n'). Hence by well-founded induction we have established $P(m,n)$ for all $m, n \in \mathbb{N}$. □

6.4.2 Eulerian graphs

Well-founded induction is a proof principle of widespread applicability. Here's an example of its use in graph theory. A graph is a pair (V,E) consisting of a set of *vertices* V and a set of *edges* E—an edge between vertices v and v' is represented as an unordered pair $\{v,v'\}$. A graph is *connected* iff any two vertices v, v' are connected by a path of edges $\{v_0,v_1\},\{v_1,v_2\},\cdots,\{v_{n-1},v_n\}$ where $v = v_0$ and $v_n = v'$. A circuit of a graph consists of a path $\{v_0,v_1\},\{v_1,v_2\},\cdots,\{v_{n-1},v_n\}$ for which $v_0 = v_n$. A circuit is *Eulerian* iff it visits each edge exactly once. When does a finite connected graph have a Eulerian circuit? The answer to this

question, the theorem below, is due to the great mathematician Leonhard Euler (1707-1783). Reputedly he was asked by the townspeople of Königsberg whether it was possible to go for a walk in the town so as to cross each of its numerous bridges exactly once (Was it? See the figure and Theorem 6.11 below).



Theorem 6.11 *A finite connected graph has an Eulerian circuit iff every vertex has even degree, i.e. has an even number of edges connected to it.*

Proof.
"only if": Consider a finite connected graph. Assume it has an Eulerian circuit. Because the graph is connected, each vertex must appear at least once in the circuit (why?). Each occurrence of a vertex in the Eulerian circuit is accompanied by a pair of distinct edges—one going into the vertex and one going out. All edges appear precisely once in the Eulerian circuit, so each vertex has even degree.
"if": For finite connected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, define

$$G_1 \preceq G_2 \iff V_1 \subseteq V_2 \;\&\; E_1 \subseteq E_2 \; .$$

The relation $\prec$ between finite connected graphs is an example of the product of two well-founded relations, so is itself well-founded. We shall use well-founded induction to establish the following property of all finite connected graphs:

> if each vertex has even degree, then the graph has an Eulerian circuit.

We take the above as our induction hypothesis.

Let G be a finite connected graph in which each vertex has even degree. Assume that for all graphs G' with $G' \prec G$ if each vertex of G' has even degree, then G' has an Eulerian circuit. That is, we assume the induction hypothesis for all $G' \prec G$.

We first find a circuit C in the graph. Starting at some vertex (it doesn't matter which) form a maximal path along edges in which no edge appears more than once. Because the graph is finite such a path must contain a loop, the circuit C. Any occurrence of a vertex in C is accompanied by a pair of distinct edges—one ingoing and one outgoing. Remove all the edges of C from the graph G. This will result in one or more connected components G', where all vertices of G' have even degree and $G' \prec G$. Hence, each such connected component has an Eulerian circuit. Linking these into C we obtain an Eulerian circuit for G. □

6.4.3 Ackermann's function

Ackermann's function provides a counterexample to the conjecture that all computable functions are primitive recursive—it grows way too fast.[2] As a recursive program Ackermann's function looks like:

$$\begin{aligned} A(x,y) = \;& \textbf{if } x = 0 \textbf{ then } y+1 \textbf{ else} \\ & \textbf{if } y = 0 \textbf{ then } A(x-1,1) \textbf{ else} \\ & \qquad A(x-1, A(x, y-1)) \end{aligned}$$

But the fact that it can be written as a program leaves open the possibility of nontermination, that Ackermann's function is undefined for some input $x, y \in \mathbb{N}_0$. However:

Theorem 6.12 *There is a unique function* $ack : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ *such that*

$$ack(m,n) = \begin{cases} n+1 & \text{if } \; m = 0 \\ ack(m-1,1) & \text{if } \; m \neq 0, n = 0 \\ ack(m-1, ack(m, n-1)) & \text{otherwise} \end{cases}$$

for all $m, n \in \mathbb{N}_0$.

[2] Computable and primitive recursive functions are central topics in the second-year CS course on "Computability."

Proof. (See the next section for an alternative, simpler proof which uses the powerful principle of well-founded recursion.)

We first show that there is a partial function $ack : \mathbb{N}_0 \times \mathbb{N}_0 \rightharpoonup \mathbb{N}_0$ satisfying the equation

$$ack(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ ack(m-1,1) & \text{if } m > 0, n = 0 \ \&\ ack(m-1,1) \text{ is defined} \\ ack(m-1, ack(m,n-1)) & \text{if } m,n > 0 \ \&\ ack(m,n-1) \text{ and} \\ & ack(m-1,ack(m,n-1)) \text{ are both defined,} \end{cases}$$

for all $m, n \in \mathbb{N}_0$. Consider the following rules which capture the evaluation of Ackermann's function to a final value:

$$\frac{}{(0,n) \Downarrow n+1}$$

$$\frac{(m-1,1) \Downarrow k}{(m,0) \Downarrow k} \quad (m > 0)$$

$$\frac{(m,n-1) \Downarrow l \qquad (m-1,l) \Downarrow k}{(m,n) \Downarrow k} \quad (m,n > 0)$$

The relation $\Downarrow \subseteq (\mathbb{N}_0 \times \mathbb{N}_0) \times \mathbb{N}_0$, inductively defined by the rules, is a partial function. This can be shown by rule induction with induction hypothesis $P(m,n,k)$ defined by

$$P(m,n,k) \ \Leftrightarrow_{def} \ \forall k' \in \mathbb{N}_0.\ (m,n) \Downarrow k' \Rightarrow k = k' \ .$$

Define

$$ack(m,n) = k \ \Leftrightarrow_{def} \ (m,n) \Downarrow k \ .$$

The closure of $\Downarrow$ under the rules ensures that ack satisfies the equation above for all $m, n \in \mathbb{N}_0$.

Now we can show that the partial function ack is in fact total. That $ack(m,n)$ is defined for all $m, n \in \mathbb{N}_0$ is proved by well-founded induction on (m,n) ordered lexicographically. The induction hypothesis is

$$D(m,n) \ \Leftrightarrow_{def} \ ack(m,n) \text{ is defined.}$$

This shows the existence of a function $ack : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ satisfying the equation stated in the theorem.

To show uniqueness assume that $ack' : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ satisfies the same equation for all $m, n \in \mathbb{N}_0$. We can show $ack = ack'$. This is proved by well-founded induction on (m,n) ordered lexicographically with induction hypothesis

$$U(m,n) \ \Leftrightarrow_{def} \ ack(m,n) = ack'(m,n) \ .$$

□

In practice a program to calculate Ackermann's function won't terminate in a reasonable time on any machine for all but the smallest values.

Exercise 6.13 Complete the proof of Theorem 6.12 by filling in the details in: the proof by rule induction that the relation $\Downarrow$ is a partial function; the proofs of the existence and uniqueness of Ackermann's function by well-founded induction. □

Exercise 6.14 (McCarthy's 91 function) Show the relation $\prec$, where

$$n \prec m \Leftrightarrow m < n \leq 101,$$

for $n, m \in \mathbb{N}_0$, is well-founded.

Show by the technique in the proof of Theorem 6.12 that there is a partial function $f : \mathbb{N}_0 \rightharpoonup \mathbb{N}_0$ satisfying

$$f(x) = \begin{cases} x - 10 & \text{if } x > 100 \ , \\ f(f(x+11)) & \text{otherwise} \ , \end{cases}$$

for all $x \in \mathbb{N}_0$.

Show by well-founded induction with respect to $\prec$ that

$$f(x) = \begin{cases} x - 10 & \text{if } x > 100 \ , \\ 91 & \text{otherwise} \ , \end{cases}$$

for all $x \in \mathbb{N}_0$. □

6.5 Well-founded recursion

Earlier in the course we have used both definition by structural induction (*e.g.* in defining the model of truth assignments in Section 2.3) and definition by induction (*e.g.* in the proof of Lemma 3.25). Such definitions are a form of recursive definition: the result of a function on an argument is defined in terms of the results of the same function on strictly smaller arguments. For example, we can define the *length* of Boolean propositions by the following clauses:

$$\begin{aligned}
&\mathit{length}(\mathtt{a}) = 1 \ , \\
&\mathit{length}(\mathtt{T}) = 1 \ , \quad \mathit{length}(\mathtt{F}) = 1 \ , \\
&\mathit{length}(\mathtt{A} \wedge \mathtt{B}) = 1 + \mathit{length}(\mathtt{A}) + \mathit{length}(\mathtt{B}) \ , \\
&\mathit{length}(\mathtt{A} \vee \mathtt{B}) = 1 + \mathit{length}(\mathtt{A}) + \mathit{length}(\mathtt{B}) \ , \\
&\mathit{length}(\neg \mathtt{A}) = 1 + \mathit{length}(\mathtt{A}) \ .
\end{aligned}$$

A well-known example from mathematics is that of the Fibonacci numbers $0, 1, 1, 2, 3, 5, 8, 13, \cdots$, which we visited in Section 1.3. They are given by a recurrence relation

$$\mathit{fib}(0) = 0, \quad \mathit{fib}(1) = 1, \quad \mathit{fib}(n) = \mathit{fib}(n-1) + \mathit{fib}(n-2) \text{ for } n > 1 \ ,$$

in which the nth Fibonacci number is defined in terms of the two preceding numbers.

In a similar way we are entitled to define functions on an arbitrary well-founded set. Suppose B is a set with a well-founded relation $\prec$. Definition by well-founded induction, traditionally called *well-founded recursion*, allows the definition of a function f from B by specifying its value $f(b)$ at an arbitrary b in B in terms of $f(b')$ for $b' \prec b$. In more detail:

Definition by well-founded recursion
Suppose B is a set with a well-founded relation $\prec$. Suppose C is a set and $F(b, c_1, \cdots, c_k, \cdots)$ is an expression such that

$$\forall b \in B,\ c_1, \cdots, c_k, \cdots \in C. \quad F(b, c_1, \cdots, c_k, \cdots) \in C \ .$$

Then, a recursive definition of the form, for all $b \in B$,

$$f(b) = F(b, f(b_1), \cdots, f(b_k), \cdots) \ ,$$

where $b_1 \prec b, \cdots, b_k \prec b, \cdots$, determines a unique total function $f : B \to C$ (*i.e.*, there is a unique $f : B \to C$ which satisfies the recursive definition).

You can check that definitions by mathematical induction, structural induction, and, in particular of the Fibonacci numbers fit the general scheme of definition by well-founded recursion.[3]

Well-founded recursion and induction constitute a general method often appropriate when functions are intended to be total. For example, it immediately follows from well-founded recursion that that there is a unique total function on the nonnegative integers such that

$$\mathit{ack}(m,n) = \begin{cases} n+1 & \text{if } \ m = 0 \ , \\ \mathit{ack}(m-1, 1) & \text{if } \ m \neq 0, n = 0 \ , \\ \mathit{ack}(m-1, \mathit{ack}(m, n-1)) & \text{otherwise} \ , \end{cases}$$

for all $m, n \in \mathbb{N}_0$; observe that the value of *ack* at the pair (m, n) is defined in terms of its values at the lexicographically smaller pairs $(m-1, 1)$ and $(m, n-1)$. In fact, a great many recursive programs are written so that some measure within a well-founded set decreases as they are evaluated.

Not all recursive definitions are well-founded; it's a fact of life that programs may fail to terminate, and so in general determine *partial* functions from input to output. The techniques of semantics, domain theory (where least fixed points play a central role—see Sections 5.6, 5.5) or operational semantics (based on inductive definitions—see *e.g.* the proof of Theorem 6.12) apply in this broader situation.[4]

[3] The non-examinable proof justifying well-founded recursion is presented in Section 6.5.1.
[4] *Cf.* the Part IB course 'Semantics' and the Part II course 'Denotational Semantics.'

6.5.1 The proof of well-founded recursion

We need a little notation to justify well-founded recursion precisely and generally. Assume B is a set with a well-founded relation $\prec$. Each element b in B has a set of predecessors

$$\prec^{-1}\{b\} = \{b' \in B \mid b' \prec b\} \ .$$

For any $B' \subseteq B$, a function $f : B \to C$ to a set C, restricts to a function $f \restriction B' : B' \to C$ by taking

$$f \restriction B' = \{(b, f(b)) \mid b \in B'\} \ .$$

A very general form of definition by well-founded recursion is justified by the following powerful theorem:

Theorem 6.15 *(Well-founded recursion)*
Let $\prec$ be a well-founded relation on a set B. Let C be a set. Suppose $F(x,p) \in C$, for all $x \in B$ and functions $p : \prec^{-1}\{x\} \to C$. Then there is a unique function $f : B \to C$ such that

$$\forall x \in B.\ f(x) = F(x, f \restriction \prec^{-1}\{x\}) \ . \qquad (*)$$

Proof. Define an *approximant* to be a partial function $g : B \rightharpoonup C$ such that for all $x \in B$, if $g(x)$ is defined then

$$(\forall z \prec x.\ g(z) \text{ is defined}) \ \text{ and } \ g(x) = F(x, g \restriction \prec^{-1}\{x\}) \ .$$

The proof has two parts. We first show an *agreement* property. For any two approximants g, h,

$$\text{if } g(x) \text{ is defined } \&\ h(x) \text{ is defined, then } g(x) = h(x) \ ,$$

for any $x \in B$. The agreement property, $A(x)$, is proved to hold for all $x \in B$ by well-founded induction on $\prec$. For $x \in B$, assume $A(z)$ for every $z \prec x$. We require $A(x)$. Assume that $g(x)$ and $h(x)$ are both defined. If $z \prec x$, then

$$g(z) \text{ is defined } \&\ h(z) \text{ is defined}$$

as g and h are approximants. As $A(z)$ we obtain

$$g(z) = h(z) \ .$$

Hence

$$g \restriction \prec^{-1}\{x\} = h \restriction \prec^{-1}\{x\} \ .$$

It now follows that

$$g(x) = F(x, g \restriction \prec^{-1}\{x\}) = F(x, h \restriction \prec^{-1}\{x\}) = h(x) \ ,$$

again using the fact that g and h are approximants. Thus $A(x)$.

It follows that there can be at most one function f satisfying $(*)$. We now show that there exists such a function. We build the function by taking the union of a set of approximants $f_x : \prec^{*-1}\{x\} \to C$, for $x \in B$. To show suitable functions exist we prove the following property $E(x)$ holds for all $x \in B$ by well-founded induction on $\prec$:

$$\text{there exists an approximant } f_x : \prec^{*-1}\{x\} \to C$$

—note the approximant is necessarily unique by the agreement property.

Let $x \in B$. Suppose $\forall z \prec x.\ E(z)$. Then

$$h = \bigcup \{f_z \mid z \prec x\}$$

is a function, being the union of approximants which agree when defined on a common argument. Taking

$$f_x = h \cup \{(x, F(x,h))\}$$

gives an approximant $f_x : \prec^{*-1}\{x\} \to C$. This completes the well-founded induction, yielding $\forall x \in B.\ E(x)$.

Now we take $f = \bigcup_{x \in B} f_x$. By the agreement property, this yields $f : B \to C$, and moreover f is the unique function satisfying $(*)$. □

Appendix A

Exercises

The following weekly exercise sheets are assembled from selected exercises in the notes.

EXERCISES 1. Numbers and Induction

Exercise A.1 Show for any integer m that $\sqrt{m}$ is rational iff m is a square, *i.e.* $m = a^2$ for some integer a. □

Exercise A.2 Suppose 99 passengers are assigned to one of two flights, one to Almeria and one to Barcelona. Show one of the flights has at least 50 passengers assigned to it. □

Exercise A.3 Prove by mathematical induction that 7 divides $2^{4n+2} + 3^{2n+1}$ for all nonnegative integers n.

Exercise A.4 Prove $n^2 > 2n$ for all $n \geq 3$. □

Exercise A.5 There are five equally-spaced stepping stones in a straight line across a river. The distance d from the banks to the nearest stone is the same as that between the stones. You can hop distance d or jump $2d$. So for example you could go from one river bank to the other in 6 hops. Alternatively you might first jump, then hop, then jump, then hop. How many distinct ways could you cross the river (you always hop or jump forwards, and don't overshoot the bank)?

Describe how many distinct ways you could cross a river with n similarly spaced stepping stones. □

Exercise A.6 Let

$$\varphi = \frac{1 + \sqrt{5}}{2}$$

—called the golden ratio. Show that both φ and $-1/\varphi$ satisfy the equation

$$x^2 = x + 1 \ .$$

Deduce that they both satisfy

$$x^n = x^{n-1} + x^{n-2} \ .$$

Using this fact, prove by course-of-values induction that the nth Fibonacci number,

$$\mathit{fib}(n) = \frac{\varphi^n - (-1/\varphi)^n}{\sqrt{5}} \ .$$

[Consider the cases $n = 0$, $n = 1$ and $n > 1$ separately.] □

EXERCISES 2. Sets and Logic

Exercise B.1 Let $A = \{1, 3, 5\}$ and $B = \{2, 3\}$. Write down explicit sets for: $A \cup B$, $A \cap B$, $A \setminus B$, $B \setminus A$, $(A \cup B) \setminus B$ and $(A \setminus B) \cup B$. □

Exercise B.2 Describe the set $A \cup B \cup C$ as a union of 7 disjoint sets (*i.e.*, so each pair of sets has empty intersection). □

Exercise B.3 In a college of 100 students, 35 play football, 36 row and 24 play tiddlywinks. 13 play football and row, 2 play football and tiddlywinks but never row, 12 row and play tiddlywinks, while 4 practice all three activities. How many students participate in none of the activities of football, rowing and tiddlywinks? □

Exercise B.4 Using the set laws transform $(A \cap B)^c \cap (A \cup C)$ to a standard form as a union of intersections. □

Exercise B.5 Let A and B be sets. Prove $A \subseteq B \iff A \cup B = B$. □

Exercise B.6 Show in any model $\mathcal{M}$ that $[\![\neg \mathtt{B} \Rightarrow \neg \mathtt{A}]\!]_{\mathcal{M}} = [\![\mathtt{A} \Rightarrow \mathtt{B}]\!]_{\mathcal{M}}$. □

Exercise B.7 Using the set laws express $\neg(\mathtt{a} \vee \mathtt{b}) \vee (\mathtt{a} \wedge \mathtt{c})$ in conjunctive form. □

Exercise B.8 Simplify $[(\mathtt{a} \Rightarrow \mathtt{b}) \vee (\mathtt{a} \Rightarrow \mathtt{d})] \Rightarrow (\mathtt{b} \vee \mathtt{d})$ to the proposition $\mathtt{a} \vee \mathtt{b} \vee \mathtt{d}$. □

Exercise B.9

(i) Show that $\mathtt{A} \Leftrightarrow \mathtt{B} \equiv (\mathtt{A} \wedge \mathtt{B}) \vee (\neg \mathtt{A} \wedge \neg \mathtt{B})$.

(ii) Show that $\mathtt{A} \Leftrightarrow (\mathtt{B} \Leftrightarrow \mathtt{C}) \equiv (\mathtt{A} \Leftrightarrow \mathtt{B}) \Leftrightarrow \mathtt{C}$.
[The analogous result does not hold when $\Leftrightarrow$ is replaced by $\Rightarrow$—why not?]

(iii) Show that $\neg(\mathtt{B} \Leftrightarrow \mathtt{C}) \equiv ((\neg \mathtt{B}) \Leftrightarrow \mathtt{C})$. □

Exercise B.10 Sheffer's stroke is not an affliction but a logical operation $\mathtt{A}|\mathtt{B}$ out of which all the usual logical operations can be derived. It is defined by the following truth table:

A	B	A|B
F	F	T
F	T	T
T	F	T
T	T	F

Check that $\mathtt{A}|\mathtt{B} \equiv \neg(\mathtt{A} \wedge \mathtt{B})$ by showing that they have the same truth table. Describe how to define the operations of negation, conjunction and disjunction out of Sheffer's stroke. □

Exercise B.11 Define the length of a Boolean proposition by structural induction as follows:

$$
\begin{aligned}
&|\mathtt{a}| = 1, \quad |\mathtt{T}| = 1, \quad |\mathtt{F}| = 1, \\
&|\mathtt{A} \wedge \mathtt{B}| = |\mathtt{A}| + |\mathtt{B}| + 1, \\
&|\mathtt{A} \vee \mathtt{B}| = |\mathtt{A}| + |\mathtt{B}| + 1, \quad |\neg \mathtt{A}| = |\mathtt{A}| + 1 \ .
\end{aligned}
$$

Define a translation which eliminates disjunction from Boolean expressions by the following structural induction:

$$
\begin{aligned}
&tr(\mathtt{a}) = \mathtt{a}, \quad tr(\mathtt{T}) = \mathtt{T}, \quad tr(\mathtt{F}) = \mathtt{F}, \\
&tr(\mathtt{A} \wedge \mathtt{B}) = tr(\mathtt{A}) \wedge tr(\mathtt{B}), \\
&tr(\mathtt{A} \vee \mathtt{B}) = \neg(\neg tr(\mathtt{A}) \wedge \neg tr(\mathtt{B})), \quad tr(\neg \mathtt{A}) = \neg tr(\mathtt{A}) \ .
\end{aligned}
$$

Prove by structural induction on Boolean propositions that

$$
|tr(\mathtt{A})| \leq 3|\mathtt{A}| - 1 \ ,
$$

for all Boolean propositions $\mathtt{A}$. □

EXERCISES 3. Relations and functions

Exercise C.1 Prove

(i) $A \times (B \cup C) = (A \times B) \cup (A \times C)$

(ii) $A \times (B \cap C) = (A \times B) \cap (A \times C)$

(iii) $(A \times B) \cap (C \times D) = (A \cap C) \times (B \cap D)$

(iv) $(A \times B) \cup (C \times D) \subseteq (A \cup C) \times (B \cup D)$ [Show the converse inclusion does not hold in general.]

□

Exercise C.2 Show that a set $\{\{a\}, \{a, b\}\}$ behaves as an ordered pair should, *i.e.*

$$\{\{a\}, \{a, b\}\} = \{\{a'\}, \{a', b'\}\} \iff a = a' \ \& \ b = b' \ .$$

[This is trickier than you might at first think. Consider the two cases $a = b$ and $a \neq b$.] □

Exercise C.3 If A has k elements and B has m elements, how many relations are there between A and B? □

Exercise C.4 If A and B are finite sets with m and n elements respectively, how many functions and how many partial functions are there from A to B? □

Exercise C.5 Let $A = \{1, 2, 3, 4\}$, $B = \{a, b, c, d\}$ and $C = \{x, y, z\}$. Let $R = \{(1, a), (2, d), (3, a), (3, b), (3, d)\}$ and $S = \{(b, x), (b, z), (c, y), (d, z)\}$. What is their composition $S \circ R$? □

Exercise C.6 Let D be the set $\{x \in \mathbb{R} \mid x > 1\}$. Define a binary relation $g \subseteq D \times D$ by taking

$$(u, v) \in g \ \text{ iff } \ \frac{1}{u} + \frac{1}{v} = 1 \ .$$

(i) Express v as a formula in u for $(u, v) \in g$. Deduce that g is a function $g : D \to D$.

(ii) Define an inverse function to g and prove that it has the desired properties. Deduce that $g : D \to D$ is a bijection.

[The formula $\frac{1}{u} + \frac{1}{v} = 1$ expresses the relation between the distance of an object u and the distance of its image v from a lens with focal length 1.] □

Exercise C.7 Suppose $f : X \to Y$ is a function. Show f^{-1} preserves the Boolean operations of union, intersection and complement, *i.e.* for all $B, C \subseteq Y$,

$$\begin{aligned} f^{-1}(B \cup C) &= (f^{-1}B) \cup (f^{-1}C) \ , \qquad & f^{-1}\emptyset &= \emptyset \ , \\ f^{-1}(B \cap C) &= (f^{-1}B) \cap (f^{-1}C) \ , \qquad & f^{-1}Y &= X \ , \\ f^{-1}(B^c) &= (f^{-1}B)^c \ . \end{aligned}$$

What analogous properties hold of the direct image under f? Suppose now $f : X \rightharpoonup Y$ is a partial function. Describe how to modify the above identities to make them hold in this case. Which identities will hold if f is assumed only to be a relation? □

Exercise C.8 Show that if $a \equiv b \pmod{k}$ and $a' \equiv b' \pmod{k}$, then $a + a' \equiv b + b' \pmod{k}$, $a - a' \equiv b - b' \pmod{k}$ and $a.a' \equiv b.b' \pmod{k}$. □

Exercise C.9 Let $A = \{1, 2, 3\}$. List all the partitions of A. How many equivalence relations are there on A? How many relations are there on A? □

Exercise C.10 Draw the Hasse diagram of the partial order $(P, \subseteq)$ where P consists of all subsets of $\{a, b, c\}$. □

Exercise C.11 Let $(\mathbb{N}, \leq)$ be the set of natural numbers with the relation $m \leq n$ meaning m *divides* n. Show $(\mathbb{N}, \leq)$ is a partial order with lubs and glbs of all pairs. What are these lubs and glbs in more traditional terms? If $\mathbb{N}$ is replaced by $\mathbb{Z}$, does the divides relation still yield a partial order? □

Exercise C.12 Show that if a partial order has all lubs, then it necessarily also has all glbs and *vice versa*. □

EXERCISES 4. Countable and Uncountable Sets

Exercise D.1 Prove that the set $\mathbb{Z}$ of integers and the set $\mathbb{Q}$ of all rational numbers are countable. [Both proofs involve the same idea.] □

Exercise D.2 Show that the set of all *finite* subsets of $\mathbb{N}$ is countable. □

Exercise D.3 Show that $\mathbb{Q} \times \mathbb{Q}$ is countable. Deduce that any set of disjoint discs (*i.e.* circular areas which may or may not include their perimeter) in the plane $\mathbb{R} \times \mathbb{R}$ is countable. Is the same true if "discs" is replaced by "circles" (*i.e.* just the perimeters of the circles)? □

Exercise D.4 Show that a nonempty set A is countable iff there is a surjection $f : \mathbb{N} \to A$. □

Exercise D.5 Prove that the set of irrational numbers is uncountable. □

Exercise D.6 By using a variation on the diagonal argument, show that the powerset, $\mathcal{P}(\mathbb{N}) =_{def} \{S \mid S \subseteq \mathbb{N}\}$, is uncountable. □

Exercise D.7 Which of the following sets are finite, which are infinite but countable, and which are uncountable?

- $\{f : \mathbb{N} \to \{0,1\} \mid \forall n \in \mathbb{N}.\ f(n) \leq f(n+1)\}$
- $\{f : \mathbb{N} \to \{0,1\} \mid \forall n \in \mathbb{N}.\ f(2n) \neq f(2n+1)\}$
- $\{f : \mathbb{N} \to \{0,1\} \mid \forall n \in \mathbb{N}.\ f(n) \neq f(n+1)\}$
- $\{f : \mathbb{N} \to \mathbb{N} \mid \forall n \in \mathbb{N}.\ f(n) \leq f(n+1)\}$
- $\{f : \mathbb{N} \to \mathbb{N} \mid \forall n \in \mathbb{N}.\ f(n) \geq f(n+1)\}$

[You may use the result of the exercise above.] □

EXERCISES 5. Constructions on Sets

Exercise E.1 Let $A_2 = \{1, 2\}$ and $A_3 = \{a, b, c\}$. List the elements of the four sets $(A_i \to A_j)$ for $i, j \in \{2, 3\}$. Annotate those elements which are injections, surjections and bijections. □

Exercise E.2 Let X and Y be sets. Show there is a bijection between the set of functions $(X \to \mathcal{P}(Y))$ and the set of relations $\mathcal{P}(X \times Y)$. □

Exercise E.3 Use lambda notation to describe bijections

$$[(A \times B) \to C] \cong [A \to (B \to C)] \ ,$$
$$[A \to (B \to C)] \cong [B \to (A \to C)] \ .$$

□

Exercise E.4 Let X be a set. The set $\{\mathtt{T}, \mathtt{F}\}$ consists of the truth values $\mathtt{T}$ and $\mathtt{F}$. Let $Y \subseteq X$. Define its *characteristic function* $\chi_Y : X \to \{\mathtt{T}, \mathtt{F}\}$ by taking

$$\chi_Y(x) = \begin{cases} \mathtt{T} & \text{if } x \in Y \\ \mathtt{F} & \text{if } x \notin Y \end{cases}$$

for all $x \in X$. Show the function taking Y to χ_Y is a bijection from $\mathcal{P}(X)$ to $(X \to \{\mathtt{T}, \mathtt{F}\})$. □

Exercise E.5 This exercise guides you through to a proof that for any sets X and Y, with Y containing at least two elements, there cannot be an injection from the set of functions $(X \to Y)$ to X.

(i) Let X be a set. Prove there is no injection $f : \mathcal{P}(X) \to X$. [Hint: Consider the set $W =_{\text{def}} \{f(Z) \mid Z \subseteq X \ \& \ f(Z) \notin Z\}$.]

(ii) Suppose now that a set Y has at least two distinct elements. Define an injection $k : \mathcal{P}(X) \to (X \to Y)$, from the powerset of X to the set of functions from X to Y.

(iii) Prove that there is no injection from $(X \to Y)$ to X when the set Y has at least two distinct elements. [Hint: Recall that the composition of injections is an injection.]

□

EXERCISES 6. Inductive Definitions

Exercise F.1 The set of well-bracketed strings is the subset of strings over symbols [and] defined inductively as follows:

[ ] is well-bracketed;

if x is well-bracketed, then $[x]$ is well-bracketed;

if x and y are well-bracketed, then xy is well-bracketed.

State the principle of rule induction for well-bracketed strings. Show the number of left brackets [equals the number of right brackets] in any well-bracketed string. □

Exercise F.2 The set S is defined to be the least subset of natural numbers $\mathbb{N}$ such that:

$1 \in S$;

if $n \in S$, then $3n \in S$;

if $n \in S$ and $n > 2$, then $(n-2) \in S$.

Show that $S = \{m \in \mathbb{N} \mid \ \exists r, s \in \mathbb{N} \cup \{0\}. \quad m = 3^r - 2s\}$. Deduce that S is the set of odd numbers. □

Exercise F.3 A simple language is defined with symbols a and b. The grammar of this language has the rules:

- ab is a word;
- if ax is a word, then axx is a word (where x is any string of symbols);
- if $abbbx$ is a word, then ax is a word.

(i) Is $abbbbb$ a word? Either exhibit a derivation, or prove there isn't one.

(ii) Is $abbb$ a word? Either exhibit a derivation, or prove there isn't one.

(iii) Characterise the strings which are words. Prove your characterisation is correct. □

Exercise F.4 Let I be a nonempty subset of the natural numbers $\mathbb{N} = \{1, 2, 3, \cdots\}$.
The set S is defined to be least subset of $\mathbb{N}$ such that

$I \subseteq S$, and
if $m, n \in S$ and $m < n$, then $(n-m) \in S$.

Define h to be the least member of S. This question guides you through to a proof that h coincides with the *highest common factor* of I, written $hcf(I)$, and defined to be the natural number with the properties that

$hcf(I)$ divides n for every element $n \in I$, and
if k is a natural number which divides n for every $n \in I$, then k divides $hcf(I)$.

(a) The set S may also be described as the least subset of $\mathbb{N}$ closed under certain rules. Describe the rules. Write down a principle of rule induction appropriate for the set S.

(b) Show by rule induction that $hcf(I)$ divides n for every $n \in S$.

(c) Let $n \in S$. Establish that

$$\text{if} \ \ p \cdot h < n \ \ \text{then} \ \ (n - p \cdot h) \in S$$

for all nonnegative integers p.

(d) Show that h divides n for every $n \in S$. [Hint: suppose otherwise and derive a contradiction.]

(e) Why do the results of (b) and (d) imply that $h = hcf(I)$.

□

Well-founded Induction

Exercise F.5 Let $\prec$ be a well-founded relation on a set X such that $\preceq$ is a total order. Show it need not necessarily make the set

$$\{x \in X \mid x \prec y\}$$

finite for all $y \in X$.
[Recall a total order is a partial order $\leq$ such that $x \leq y$ or $y \leq x$ for all its elements x, y. Hint: Consider the lexicographic product of $<$ and $<$ on $\mathbb{N}_0 \times \mathbb{N}_0$.] □

Exercise F.6 Let $f : X \to Y$ be a function. Show the inverse image w.r.t. f of a well-founded relation on Y is a well-founded relation on X. □

Exercise F.7 (McCarthy's 91 function) Show the relation $\prec$, where

$$n \prec m \Leftrightarrow m < n \leq 101,$$

for $n, m \in \mathbb{N}_0$, is well-founded.

The 91 function of McCarthy $f : \mathbb{N}_0 \to \mathbb{N}_0$ satisfies

$$f(x) = \begin{cases} x - 10 & \text{if } x > 100\ , \\ f(f(x + 11)) & \text{otherwise}\ , \end{cases}$$

for all $x \in \mathbb{N}_0$. Show by well-founded induction with respect to $\prec$ that

$$f(x) = \begin{cases} x - 10 & \text{if } x > 100\ , \\ 91 & \text{otherwise}\ , \end{cases}$$

for all $x \in \mathbb{N}_0$. □