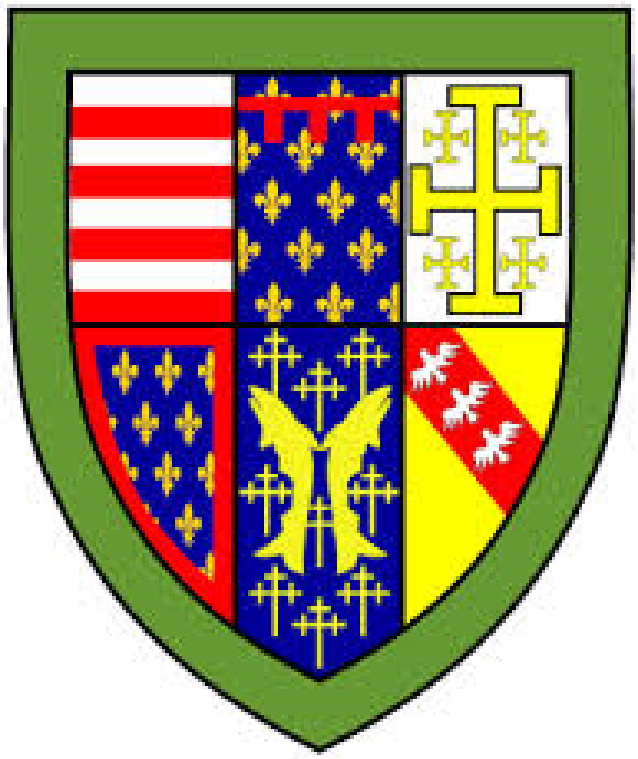
An Invitation to Discrete Mathematics

Thomas Forster

October 6, 2019



Contents

I should perhaps admit that in this file I have recycled some material from *Logic, Induction and sets*. It doesn't matter, and i'm mentioning it only in case there are people who have read that book and are waiting for an opportunity to say that I am guilty of autoplagiarism. "Why not rewrite it?" you might ask. Because it was perfect already. But *that* of course is a piece of plagiarism too (from Schönberg, as it happens). Funny, isn't it, that there is nothing about the action of copying/quoting that makes it wrong; it's just that you're supposed to say when you've done it. Perhaps the next time I rob a bank i'll just put my hand up to say i've done it and they'll let me keep the money.

Chapter 1

Introduction

Warning! Discrete Mathematics is a grabbag of tricks. It doesn't really have a unifying theme, being defined by what it excludes rather than what it contains: it's that part of first-year university mathematics that isn't *continuous*, that *isn't* the stuff that used to be called "calculus"—differentiation and integration. However, on the whole, what gets put into a course like this is not so much a body of knowledge and skills bound together by some internal logic (even negative logic) of its own, but rather the background mathematics that the lecturer thinks that the well-equipped computer science students have to have in their knapsacks. Quite what you need in your knapsack depends to a certain extent on what epoch you are going to live and work in. Nowadays you are less likely to need familiarity with different ways of representing numbers (binary, HEX, octal, decimal) than you would have done thirty years ago, but you are more likely to need to know about cryptography. Some topics which is some sense "ought" to be in a discrete maths course tend to get hived off into other courses; finite state machines often get treated separately and my notes on them are still in a separate file.

So this is a course on those bits of background mathematics that don't involve integration and differentiation, and which I think might be useful to you: it isn't supposed to have a consistent theme beyond that, so don't worry if you can't find one. One rather nice side-effect of this is that there is no obviously best book either, and you can do a lot worse than follow the habit I have had for many years of looking in the second-hand books section of charity shops and buying anything that looks like a book on first-year university mathematics. There are lots of such books (because lots of different communities need to do first-year mathematics: engineers, students doing psyschology, science, economics, computer science, medicine ...) and in places like that they can be quite cheap. There is of course always the danger that if you buy three of them you end up with three different systems of notation, and it has to be admitted that this is a pain. On the other hand, this pain is nothing more than the fact that there are lots of different and pairwise incompatible notations, and that is something—sadly—you are going to have to get used to!

In fact the single most important lesson for you to learn from this course is confidence in manipulating the mathematical symbols you will need later. We will banish mathsangst.

It is true that this material was assembled originally for the delectation of first-year students at Queen Mary. However it is now used by first students at Cambridge . . . and in their second term of studies. In their first term they will have been exposed to a certain amount of logic, and—in particular—to `ML`.

You should now read (or reread!) 'Alice in Wonderland' and 'Through the Looking Glass and what Alice found there', preferably the edition [4] with Martin Gardner's annotations, entitled 'The Annotated Alice'. Lewis Carroll was one of the nineteenth century writers who started the process that led later to the mathematisation of logic that in turn led to modern mathematical logic and computer science. In fact, while we are about it, anything else by Martin Gardner that comes your way should be snapped up and devoured.

The chief difficulty that students have with Discrete Mathematics is the lack of theorems, a lack of *deliverables*. The real lessons you will take away from this course is a not a knapsack full of theorems, but a new way of thinking about the materials. This lack of deliverables can be very disconcerting, and it often has the effect that students can be quite lost and yet not realise it. There is a difference between—on the one hand—thinking "Yes, I feel comfortable with this stuff: it looks OK" and—on the other—genuinely understanding it in the sense of being able to apply it to any purposes of yours that might crop up. If my experience of teaching this stuff is anything to go by there seems to be a very strong tendency to mistake the first for the second. The danger then is that once you realise that you hadn't, after all, understood the first three weeks of material, you suddenly find yourself three weeks behind and stuff is still coming at you at the same rate as before. The way to combat this tendency is to make sure you know what you are letting yourself in for, and know how it differs from things in which you have got embroiled in the past.

1.1 Some Puzzles to Get You Started

Don't look down on puzzles:

> *A logical theory may be tested by its capacity for dealing with puzzles, and It is a wholesome plan, in thinking about logic, to stock the mind with as many puzzles as possible, since these serve much the same purpose as is served by experiments in physical science.*
>
> Bertrand Russell

Exercise 1. *A box is full of hats. All but three are red, all but three are blue, all but three are brown, all but three are white. How many hats are there in the box?*

Exercise 2. *The main safe at the bank is secured with three locks, A, B and C. Any two of the three system managers can cooperate to open it. How many keys must each manager have?*

Exercise 3. *A storekeeper has nine bags of cement, all but one of which weigh precisely 50kg, and the odd one out is light. He has a balance which he can use to compare weights. How can he identify the rogue bag in only three weighings? Can he still do it if he doesn't know if the rogue bag is light?*

Exercise 4. *A father, a mother, a father-in-law, a mother-in-law, a husband, a wife, a daughter-in-law, a son-in-law, a niece, a nephew, a brother, a sister, an uncle and an aunt all went on holiday. There were only four people! How can this be?*[1]

Exercise 5. *There were five delegates, A, B, C, D and E at a recent summit.*

B and C spoke English, but changed (when D joined them) to Spanish, this being the only language they all had in common;
The only language A, B and E had in common was French;
The only language C and E had in common was Italian;
Three delegates could speak Portugese;
The most common language was Spanish;
One delegate spoke all five languages, one spoke only four, one spoke only three, one spoke only two and the last one spoke only one.

Which languages did each delegate speak?

Exercise 6. *People from Bingo always lie and people from Bongo always tell the truth.*

- *If you meet three people from these two places there is a single question you can ask all three of them and deduce from the answers who comes from where. What might it be?*
- *If you meet two people, one from each of the two places (but you don't know which is which) there is a single question you can ask either one of them (you are allowed to ask only one of them!) and the answer will tell you which is which. What is it?*

Exercise 7.

Brothers and sisters have I none
This man's father is my father's son

To whom is the speaker referring?

[1] I think you have to assume that the aunt is an aunt in virtue of being an aunt of another member of the party, that the father is a father of another member of the party, and so on.

Exercise 8. *You are told that every card that you are about to see has a number on one side and a letter on the other. You are then shown four cards lying flat, and on the uppermost faces you see*

E K 4 7

It is alleged that any card with a vowel on one side has an even number on the other. Which of these cards do you have to turn over to check this allegation?

Exercise 9. *A bag contains a certain number of black balls and a certain number of white balls. (The exact number doesn't matter). You repeatedly do the following. Put your hand in the bag and remove two balls at random: if they are both white, you put one of them back and discard the other; if one is black and the other is white, you put the black ball back in the bag and discard the white ball; if they are both black, you discard them both and put into the bag a random number of white balls from an inexhaustible supply that just happens to be handy.*

What happens in the long run?

Exercise 10.

	3	8						
	1	6		4		9	7	
4		7	1					6
		2	8		7			5
	5			1			8	
8			4			2		
7		5			1	8		4
	4	3		5		7	1	
						6		

Exercise 11. *Hilary and Jocelyn are married. One evening they invite Alex and Chris (also married) to dinner, and there is a certain amount of handshaking, tho' naturally nobody shakes hands with themselves or their spouse. Later, Jocelyn asks the other three how many hands they have shaken and gets three different answers.*

How many hands has Hilary shaken? How many hands has Jocelyn shaken?

The next day Hilary and Jocelyn invite Chris and Alex again. This time they also invite Nicki and Kim (also married). Again Jocelyn asks everyone how many hands they have shaken and again they all give different answers.

How many hands has Hilary shaken this time? How many has Jocelyn shaken?

Exercise 12. *You are shown 99 boxes, each of them containing some blue balls and some red balls, which you allowed to count. You are then told that you may take 50 of the boxes, and the idea is to select your boxes so that you end up with more than half the blue balls and more than half of the black balls.*

Can you do it?

These two are slightly more open-ended.

Exercise 13. *You are given a large number of lengths of fuse. The only thing you know about each length of fuse is that it will burn for precisely one minute. (They're all very uneven: in each length some bits burn faster than others, so you don't know that half the length will burn in half a minute or anything like that). The challenge is to use the burnings of these lengths of fuse to measure time intervals. You can obviously measure one minute, two minutes, three minutes and so on by lighting each fuse from the end of the one that's just about to go out. What other lengths can you measure?*

Exercise 14. *A Cretan says "Everything I say is false". What can you infer?*

Those exercises might take you a little while, but they are entirely do-able even before you have done any logic. Discuss them with your friends. Don't give up on them: persist until you crack them!

If you disposed of all those with no sweat try this one:

Exercise 15. *You and I are going to play a game. There is an infinite line of beads stretching out in both directions. Each bead has a bead immediately to the left of it and another immediately to the right. A* **round** *of the game is a move of mine followed by a move of yours. I move first, and my move is always to point at a bead. All the beads look the same: they are not numbered or anything like that. I may point to any bead I have not already indicated. You then have to give the bead a label, which is one of the letters* `a`–`z`*. The only restriction on your moves is that whenever you are called upon to put a label on the neighbour of a bead that already has a label, the new label must be the appropriate neighbour of the bead already labelled, respecting alphabetical order: the predecessor if the new bead is to the left of the old bead, and the successor if the new bead is to the right. For example, suppose you have labelled a bead with '*`p`*'; then if I point at the bead immediately to the right of it you have to label that bead '*`q`*'; were I to point to the bead immediately to the left of it you would have to label it '*`o`*'. If you have labelled a bead '*`z`*' then you would be in terminal trouble were I to point at the bead immediately to the right of it; if you have labelled a bead '*`a`*' then you would be in terminal trouble if I then point at the bead immediately to the left of it. If you have labelled some bead '*`j`*' and some bead to the right of it '*`q`*' then the beads between the two has to be given labels between '*`k`*' and '*`p`*'. We decide in advance how many rounds we are going to play. I win if you ever violate the condition on alphabetic ordering of labels. You win if you don't lose.*

Clearly you are going to win the one-round version, and it's easy for you to win the two-round version. The game is going to last for five rounds.

How do you plan your play?

How do you feel about playing six rounds?

It may surprise you (and it would probably surprise most people) to be told that all these puzzles are mathematical, even tho's they don't involve any numerical calculation .... Welcome to Discrete Mathematics!

Chapter 2

Read this first!

I am not going to expose you to any new Mathematics in this chapter, but that doesn't mean you should skip it. Read it first, to prepare you for what is to come. I am a great believer in *Naming the Devil*: philosophers have argued for a long time about the relation between thought and language but we do all agree that many things become easier to see and recognise once you have a name for them[1]. In the next few sections I shall be introducing you to some terminology. You won't have to prove anything using it, but it will help you once you come to proving other things, and it will help you get your bearings.

2.1 The Existential Other

Many of my students ask me how to write an exam answer. Robert Craft asked Stravinsky whom he wrote for; Stravinsky replied "For myself and the Hypothetical Other". That should be the audience for which you write your exam answers, and explanations. Who is your Hypothetical Other?

There are many approaches to this, but one that will help you in practising to write answers—or take notes—is to write something in the form of an explanation for a suspicious person of at least normal intelligence. If you have an annoying younger sibling who won't take anything on trust, then write it for them. Or it could be your supervision partner who slept in and missed the supervision at which you learnt these cool things that you now have to explain to them. Or you could try writing a message to your future self, for when you come to revise the material you are taking note son you will surely have forgotten at least *some* of it. The key idea is to write *for an audience*. The point of having an audience in mind is that—when you wonder "Shall I put this in . . . ?

[1] In [10] Oliver Sacks wrote "Muscular Dystrophy . . . was never seen until Duchenne described it in the 1850's. By 1860, after his original description, many hundreds of cases had been recognised and described, so much so that Charcot said "How come that a disease so common, so widespread, and so recognisable at a glance—a disease that has doubtless always existed—how come that it is only recognised now? Why did we need M. Duchenne to open our eyes?" "

Shall I leave that out?"—your knowledge of the audience tells you which way to jump.

Bearing this in mind will help you write answers to these questions which might enable me to get a clear picture of what you understand and what you don't.

2.2 Here are some things you will need

Exercise 16.

1. *Factorise* $x^2 - y^2$.

2. *What is the sum of the first* n *natural numbers?*

3. *If you toss a coin and roll a die*[2] *how many different results can you get? How about two coins and two dice? How about four dice? Make explicit to yourself the way you calculated these answers.*

4. *How many ways are there of arranging* n *things in a row? Can you explain why?*

5. *The expression '*$\binom{n}{r}$*'; what does it mean, and what is its value?*

6. *What is a function? Explain* injective *and* surjective. *Let* $A = \{a, b\}; B = \{1, 2, 3\}$*; What are the members of* $A \times B$*? Of* $B \times A$*?. Identify these members (to your annoying younger sibling). How many functions are there from* A *to* B*? From* B *to* A*? Identify these functions (to your annoying younger sibling)*

7. *What is a prime number?*

8. *(a)* $x^{(y^z)}$*: is this the same as* $(x^y)^z$*? Can you simplify either of these further?*

   (b) What is x^0*? Do you remember why?*

9. *We will also assume you know about matrix multiplication, tho' not much will hang on it.*

   Multiply the matrix

$$\begin{matrix} 1 & 2 & 0 \\ 2 & 1 & 3 \\ 0 & 3 & 1 \end{matrix}$$

   by the matrix

[2] Yes, 'die' is the singular of 'dice'!

$$\begin{matrix} 2 & 1 & 1 \\ 0 & 1 & 3 \\ 1 & 0 & 1 \end{matrix}$$

explaining the steps.

Much if not all of this will be explained below

2.3 Things you are not going to need and which we won't cover

Complex numbers, truth tables ...

2.4 Things you mightn't need

Some of the following you won't need directly in Cambridge 1a Discrete Mathematics. However you will need them for other courses, and if you haven't got them under control it might be an early sign of trouble ahead not just in DM but also in the other courses for which you will need them.

- Sums of Arithmetic Progressions, sums of Geometric Progressions,
- Consider the power series

  $$1 + x + x^2 + 2x^3 + 3x^4 + 5x^5 + 8x^6 + 13x^7 \ldots$$

  where the coefficient of 'x^n' is the nth Fibonacci number. Do you know how to sum it?
- Expansion of $(1 + x)^n$;

2.5 Philosophical Introduction: the Pædagogical Difficulties

It might seem odd to kick off a file of course materials on Discrete Mathematics with a section that has a title like this, but there is a reason. Some proofs just are hard, and people experience difficulty accordingly. But there are bottlenecks where people experience difficulties with the underlying concepts, and particularly with the notation.

The hard part of doing discrete maths isn't learning the proofs of the theorems. By and large the proofs are not particularly difficult at this level—though they can appear daunting. The hard part is making a certain kind of mental jump. Once you have made this jump, everything is easy. Let me explain.

Mathematicians often complain that lay people think that mathematics is about numbers. It isn't, and they are right to complain. Not just because it's a mistake, but because it's a mistake that throws people off the scent. Mathematics is a process of formalisation and abstraction that can be applied to all sorts of things, not just numbers. It just so happens that the only bits of mathematics that the average lay person encounters is mathematics as applied to *numerosity*, which is where numbers come from. In fact we can apply mathematical methods to all sorts of other ideas. (Geometry, for example).

2.5.1 Variables and Things

In applying mathematical methods to a topic we find that we take a number of steps. One of them is summarised in a famous remark of the twentieth century philosopher W.V.Quine: "To be is to be the value of a variable". You are probably quite happy if I say

> *"Let x be a number between 1 and 1000; divide it by 2 ..."*

or (if you are old enough to have done geometry at school)

> *"Let ABC be a triangle; extend the side AB ..."*.

In contrast you are almost certainly not happy if I say

> *"Let R be a binary relation on a set X; compose it with its inverse..."*.

Why is this? It is because numbers (and perhaps triangles) are mathematical objects in your way of thinking, whereas relations aren't. And what has this got to do with being the value of a variable? Quine's criterion for a species of object to be a *mathematical* object (in the way that numbers—or perhaps triangles—are) is that variables can range over mathematical objects. From your point of view, the utility of the observations of Quine's is that it enables you to tell which things you are comfortable thinking of mathematically.

Computer Scientists in their slang make the distinction (from the point of view of a programming language) between **first class objects** and the rest. First class objects are the kinds of things that the variables of the language can take as values. Typically, for a programming language, numbers such as integers or floating-point reals will be first-class objects, but *operations* on those numbers will not.

This distinction between first-class objects and the rest is echoed in ordinary language (well, in all the ordinary languages known to me, at least) by the difference between nouns and verbs.

Expand on this; burble nominalisation.

Let us take a live example, one that bothers many beginners in discrete mathematics. Relations are not mathematical objects for most people. ("Let R be a binary relation on a set X..."!) In consequence many people are not happy about being asked to perform operations on relations. The problem is not that they are unacquainted with the fact that—for example—the uncle-of

relation is the composition of the brother-of relation with the parent-of relation. This is, after all, something you can explain easily to any foreigner who asks you what the word 'uncle' means! The problem is that they don't know that this fact *is a fact about composition of relations*. This is because they don't think of relations as being the kind of things you perform operations on, and that in turn is because they don't think of relations as **things** at all!

You are quite happy applying operations to numbers. The conceptual leap you have to make here is to be willing to apply operations to relations. Although thinking of them as things (rather than as relations between things) and then thinking of them as the substrates of operations are two steps rather than one, it's probably best to think of them as two parts of a single move.

But relations are only one example of entities that you are now going to have to think of mathematically. Others are sets, functions and graphs.

This business of manufacturing new kinds of thing for us to think about and do things to has been much discussed by philosophers ever since the days of the Ancient Greeks. I first encountered the word for this process—"hypostasis" when I was a philosophy student. But it's not just philosophy students who need to think about it—as you can see!

Null objects

Another sign that a species of object (number, set, line ...) has become a mathematical object for you is when you are happy about degenerate or *null* objects of that species. You may remember being told in school that the discovery that zero was a number was a very important one. An analogous discovery you will be making here is that the empty set is a set. (Perhaps this is the same discovery, since numbers like 1, 2, 3, ... (though not $1{\cdot}5$, $5/3$, π ...) are answers to questions about how many elements there are in a set. "0" is the answer to "How many things are there in the empty set?".) If you think that the empty set "isn't there" it's because you don't think that sets have any existence beyond the existence of their members. Contrast this with the relaxed feeling you have about an empty folder or file in a directory on your computer (as it might be `Things_I_learnt_from_Twitter.xls`). Files and folders on your computer are things that–for you—are unproblematic objects of thought in a way that makes it possible for you to think of empty ones. You are happy to perform operations on them, after all... reading them, writing to them, copying them .... To *that* extent you are thinking of them as mathematical objects: the ability to be relaxed about the empty set is one of the things you will acquire when you start thinking of sets as mathematical objects.

duplicates material on p. **??**

We build formulæ by taking conjunctions or disjunctions of collections of formulæ. What is the conjunction of the empty set of formulæ? The disjunction of the empty set of formulæ? Never mind about the answer to this *just* yet (though we will soon); for the moment I am trying to impress you with (i) the novel idea that the question is a sensible one and (ii) that accepting the fact that it is a sensible question is part of thinking of sets as mathematical objects, which in turn is part of doing Discrete Maths.

Integrate this last para into the preceding para

(My Ph.D. thesis has the shortest title on record: "N.F." A title is a string of characters. So the shortest possible title is the empty string of characters. Note that having the empty string as your title is not the same as having no title!) TTBA "untitled". When we get on to Languages and Automata you will have to distinguish between the empty language (the language that has no formulæ in it) and the language whose sole formula is the empty string!! I want you to understand this distinction.

2.5.2 Envoi

I'm inflicting on you this brief digression on Philosophy and Foundations because every student, in mastering the skills and ideas of Computer Science, has to go through a hugely speeded-up version of the journey that Mathematics and Philosophy went through in dreaming up these objects in the first place.

2.6 The Type-Token distinction

The terminology 'type-token' is due to the remarkable nineteenth century American philosopher Charles Sanders Peirce. (You may have heard the tautology $((A \to B) \to A) \to A$ referred to as *Peirce's Law*). The two ideas of token and type are connected by the relation "is an instance of". Tokens are instances of types.

It's the distinction we reach for in situations like the following

- (i) "I wrote a *book* last year"
  (ii) "I bought two **books** today"

  In (ii) the two things I bought were physical objects, but the thing I wrote in (i) was an abstract entity. What I wrote was a *type*. The things I bought today with which I shall curl up tonight are *tokens*. This important distinction is missable because we typically use the same word for both the type and the token.

- A best seller is a book large numbers of whose *tokens* have been sold. There is a certain amount of puzzlement in copyright law about ownership of tokens of a work versus ownership of the type. James Hewitt owns the copyright in Diana's letters to him but not the letters themselves. (Or is it the other way round?)

- I remember being very puzzled when I was first told about printing. I was told that each piece of type could only be used once. Once for each book, in the sense of once for each print run Not once for each *copy* of a book. The copies from any one print run are all tokens of a type.

- I read somewhere that " ...next to Mary Woollstonecroft was buried Shelley's heart, *wrapped in one of his poems*." To be a bit more precise, it was wrapped in a *token* of one of his poems.

- You have to write an essay of 5000 words. That is 5000 word tokens. On the other hand, there are 5000 words used in this course material that come from latin. Those are word types.

- Grelling's paradox concerning the words *heterological* and *homological*. : a **heterological** word is one that is not true of itself. 'long' is heterological: it is not a *long* word. 'English' is not heterological but *homological*, for it is an English word. Notice that it is word *types* not word *tokens* that are heterological (or homological!) It doesn't make any sense to ask whether or not 'italicised' is heterological. Only word *tokens* can be italicised!

- What is the difference between "unreadable" and "illegible"? A book (type) is unreadable if it so badly written that one cannot force oneself to read it. A book (token) is illegible if it is so defaced or damaged that one cannot decypher the (tokens of) words on the page.

- We must not forget the difference between a program (type) and the tokens of it that run on various machines.

- Genes try to maximise the number of tokens of themselves in circulation. We attribute the intention to the gene *type* because it is not the action of any *one* token that invites this mentalistic metaphor, but the action of them all together. However it is the number of *tokens* that the type appears to be trying to maximise.

The type-token distinction was independently rediscovered by the people who brought us object-oriented programming. Their distinction between **class** and **object** is the same as the type-token distinction, and the divergence in notation is (presumably) caused solely by lack of communication between the various cultures that need this idea.

2.7 Copies

Buddhas

It is told that the Buddha could perform miracles. But—like Jesus—he felt they were vulgar and ostentatious, and they displeased him.

> A merchant in a city of India carves a piece of sandalwood into a bowl. He places it at the top of some bamboo stalks which are high and very slippery, and declares that he will give the bowl to anyone who can get it down. Some heretical teachers try, but in vain. They attempt to bribe the merchant to say they had succeeded. The merchant refuses, and a minor disciple of the Buddha arrives. (His name is not mentioned except in this connection). The disciple rises through the air, flies six times round the bowl, then picks it up and delivers it to the merchant. When the Buddha hears the story he expels the disciple from the order for his frivolity.

But that didn't stop him from performing them himself when forced into a corner. In *Siete Noches* [1] (from which the above paragraph is taken) J. L. Borges procedes to tell the following story, of a miracle of *courtesy*. The Buddha has to cross a desert at noon. The Gods, from their thirty-three heavans, each send him down a parasol. The Buddha does not want to slight any of the Gods, so he turns himself into thirty-three Buddhas. Each God sees a Buddha protected by a parasol he sent.[3]

Apparently he did this routinely whenever he was visiting a city with several gates, at each of which people would be waiting to greet him. He would make as many copies of himself as necessary to be able to appear at all the gates simultaneously, and thereby not disappoint anyone.

Minis

Q: How many elephants can you fit in a mini?
A: Four: two in the front and two in the back.

Q: How many giraffes can you fit in a mini?
A: None: it's full of elephants.

Q: How can you tell when there are elephants in the fridge?
A: Footprints in the butter.

Q: How can you tell when there are *two* elephants in the fridge?
A: You can hear them giggling when the light goes out.

Q: How can you tell when there are *three* elephants in the fridge?
A: You have difficulty closing the fridge door.

Q: How can you tell when there are *four* elephants in the fridge?
A: There's a mini parked outside.

Sets

If A is a set with three members and B is a set with four members, how many ordered pairs can you make whose first component is in A and whose second component is in B?

Weeeeell ... you pick up a member of A and you pair it with a member of B ... that leaves two things in A so you can do it again .... The answer must be three!

Wrong! Once you have picked up a member of A and put it into an ordered pair—it's still there!

One would tend not to use the word *token* in this connection. One would be more likely to use a word like *copy*. One makes lots of copies of the members of

[3] As is usual with Borges, one does not know whether he has a source for this story in the literature, or whether he made it up. And—again, as usual—it doesn't matter.

A. Just as the Buddha made lots of copies of himself rather than lots of *tokens* of himself. I suppose you could say that the various tokens of a type are copies of each other.

It is possible to do a lot of rigorous analysis of this distinction, and a lot of refinements suggest themselves. However, in the culture into which you are moving the distinction is a piece of background slang useful for keeping your thoughts on an even keel, rather than something central you have to get absolutely straight. In particular we will need it later (see page 84) when making sense of ideas like *disjoint union*.

2.8 The Use-Mention Distinction

We must distinguish words from the things they name: the word 'butterfly' is not a butterfly. The distinction between the word and the insect is known as the "use-mention" distinction. The word 'butterfly' has nine letters and no wings; a butterfly has two wings and no letters. The last sentence *uses* the word 'butterfly' and the one before that *mentions* it. Hence the expression 'use-mention distinction'.

Haddocks' Eyes

As so often the standard example is from [2].

> [...] The name of the song is called 'Haddock's eyes'."
>
> "Oh, that's the name of the song is it", said Alice, trying to feel interested.
>
> "No, you don't understand," the Knight said, looking a little vexed. "That's what the name is *called*. The name really is *'The agèd, agèd man'*."
>
> "Then I ought to have said, 'That's what the *song* is called'?" Alice corrected herself.
>
> "No you oughtn't: that's quite another thing! The *song* is called '*Ways and means*', but that's only what it is *called*, you know!"
>
> "Well, what *is* the song, then?" said Alice, who was by this time completely bewildered.
>
> "I was coming to that," the Knight said. "The song really is *'A-sitting on a Gate'* and the tune's my own invention".

The situation is somewhat complicated by the dual use of single quotation marks. They are used both as a variant of ordinary double quotation marks for speech-within-speech (to improve legibility)—as in "Then I ought to have said, 'That's what the *song* is called'?"—and also to make names of words or strings of words—'The agèd, agèd man'.... Even so, it does seem clear that the White

Knight has got it wrong. At the very least: if the name of the song really *is* 'The agèd agèd man' (as he says) then clearly Alice was right to say that was what the song was called. Granted, it might have more names than just that one—'Ways and means' for example—but that was no reason for him to tell her she had got it *wrong*. And again, if his last utterance is to be true, he should leave the single quotation marks off the title, or—failing that (as Martin Gardner points out in [4])—burst into song. These infelicities must be deliberate (Carroll does not make elementary mistakes like that), and one wonders whether or not the White Knight realises he is getting it wrong ... is he an old fool and nothing more? Or is he a paid-up party to a conspiracy to make the reader's reading experience a nightmare? The Alice books are one long nightmare, and perhaps not just for Alice.

Alphabet soup

People complain that they don't want their food to be full of E-numbers. What they mean is that they don't want it to be full of the things denoted by the E-numbers.[4]

Some Good Advice

Q: Why should you never fall in love with a tennis player?

A: Because 'love' means 'nothing' to them.

Apple Crumble

"Put cream on the apple crumble"
"But there isn't any cream!"
"Then put 'cream' on the shopping list!"

'Think'

"If I were asked to put my advice to a young man in one word, Prestwick, do you know what that word would be?"

"No" said Sir Prestwick.

" 'Think', Prestwick, 'Think' ".

"I don't know, R.V. 'Detail'?"

"No, Prestwick, 'Think'."

"Er, 'Courage'?"

"No! 'Think'!"

"I give up, R.V., 'Boldness'?"

"For heavan's sake, Prestwick, what is the matter with you? '*Think*'!"

[4]Mind you E-300 is Vitamin C and there's nothing wrong with *that*!

" 'Integrity'? 'Loyalty'? 'Leadership'?"

" 'Think', Prestwick! 'Think', 'Think', 'Think' 'Think'!"

Michael Frayn: *The Tin Men*. Frayn has a degree in Philosophy.

Ramsey for Breakfast

In the following example F.P. Ramsey[5] uses the use-mention distinction to generate something very close to paradox: the child's last utterance is an example of what used to be called a "self-refuting" utterance: whenever this utterance is made, it is not expressing a truth.

PARENT: Say 'breakfast'.
CHILD: Can't.
PARENT: What can't you say?
CHILD: Can't say 'breakfast'.

The Deaf Judge

JUDGE (*to PRISONER*): Do you have anything to say before I pass sentence?

PRISONER: Nothing

JUDGE (*to COUNSEL* : Did your Client say anything?

COUNSEL: 'Nothing' my Lord.

JUDGE: Funny . . . I could have sworn I saw his lips move. . .

The N-word

One standard way of creating a [token of a] name for a word is to take a token of it and put single quotes either side of it—as indeed we have been doing above . . . "My client said 'nothing' my lord". Naturally in these circumstances one wants to say that the word in question is mentioned not used. However there are circumstances in which it is felt (by some) that the word enclosed in single quotes is, nevertheless, in some sense, being used. This tends to happen when the word being mentioned is heavily taboo-ed and has to be kept at barge-pole distance, in particular with words that we are not permitted to *use*. Such words can of course still be *mentioned*; after all, how can one tell a new user of the language that they are not to use the word without somehow denoting it—that is to say, mentioning it? If you mention a word you need a name for it, but a name that we use to mention it cannot be one obtained by putting single quotes

[5] You will be hearing more of this chap.

round it. There is a slang American word for disparagingly denoting people with black skins; it has six letters and begins with 'n'. I can allude to this fact, and mention the word in so doing—as I have in fact just done. But if my way of mentioning it was by enclosing a token of it in single quotes the sky would probably land on my head. My defence would be that I am not using the word, but mentioning it. I'm not going to risk it, co's I want a quiet life. Call me a coward: I plead guilty as charged.

This is tied up with all sorts of complex and interesting issues in philosophy of language which you will have to come to grips with in the fullness of time. You could try googling 'referential opacity' if you want to read ahead.

The predicament of speakers in situations where the mentioned word is heavily taboo-ed is put to good comedic effect in Scene Five of *The Life of Brian* `http://montypython.50webs.com/scripts/Life_of_Brian/5.htm`, where Matthias, son of Deuteronomy of Gath, is to be stoned to death for *using* the name of Jehovah. In his defence he *mentions* the name:

> MATTHIAS: Look. I—I'd had a lovely supper, and all I said to my wife was, 'That piece of halibut was good enough for Jehovah.'

So: there are some words that one wishes to mention—if at all—only by using only those names of it that do not contain embedded occurrences of it—embedded within single quotes for example. *Prima facie* there is a question about how a word can acquire other safe names in this way, and there is presumably a literature on this question ... but I don't know any of it.

Remove this disclaimer

Banach-Tarski

There aren't many good mathematical jokes, God knows, but this is one of them...

> Give a good anagram of 'Banach-Tarski'.

and the answer is

> 'Banach-Tarski Banach-Tarski'

...the point being that the Banach-Tarski paradox (look it up) concerns the possibility of chopping up a solid sphere into finitely many pieces and reassembling the pieces to make two new spheres each the same size as the original.

Do we want subsubsection or subsection* here?

Fun on a Train

The use-mention distinction is a rich source of jokes. One of my favourites is the joke about the compartment in the commuter train, where the passengers have travelled together so often that they have long since all told all the jokes they know, and have been reduced to the extremity of numbering the jokes and reciting the numbers instead. In most versions of this story, an outsider arrives and attempts to join in the fun by announcing "*Fifty-six!*" which is met with

a leaden silence, and he is tactfully told "It's not the joke, it's the way you tell it". In another version he then tries "*Forty-two!*" and the train is convulsed with laughter. Apparently that was one they hadn't heard before.[6]

A good text to read on the use-mention distinction is the first six paragraphs (that is, up to about p. 37) of chapter 1 of Quine's [9].

Related to the use-mention distinction is the error of attributing powers of an object to representations of that object. I have always tended to think that this is a use-mention confusion, but perhaps it's a deliberate device, and not a confusion at all. So do we want to stop people attributing to representations powers that strictly belong to the things being represented? Wouldn't that spoil a lot of fun? Perhaps, but on the other hand it might help us understand the fun better. There was once a famous English stand-up comic by the name of *Les Dawson* who (did mother-in-law jokes but also) had a routine which involved playing the piano *very badly*. I think Les Dawson must in fact have been quite a good pianist: if you want a sharp act that involves playing the piano as badly as he seemed to be playing it you really have to know what you are doing[7]. The moral is that perhaps you only experience the full *frisson* to be had from use-mention confusion once you understand the use-mention distinction properly.

We make a fuss of this distinction because we should always be clear about the difference between a thing and its representation. Thus, for example, we distinguish between numerals and the numbers that they represent. (Notice that bus "numbers" are typically numerals not numbers, in the sense that the thing that identifies the bus is the character (the numeral) on it, rather than the number denoted by that numeral. Not long ago, needing a number 7 bus to go home, I hopped on a bus that had the string '007' on the front. It turned out to be an entirely different route! And this despite the fact that the numerals '007' and '7' denote the same number. Maybe this confusion in people's minds is one reason why this service is now to be discontinued.[8]) If we write numbers in various bases (Hex, binary, octal ...) the numbers stay the same, but the numerals we associate with each number change. Thus the numerals '`XI`', 'B', '11', '13' '1011' all represent the same number.[9]

Hotter Temperatures on the Way

No! Hotter *weather* on the way. Temperatures are not hot. They are numbers. Numbers are big or small, or lucky or unlucky.

Rockets

Missing: Number of children fleeing care in Cambridgeshire rockets

[6] For sophisticates: this is a joke about *dereferencing*.

[7] Wikipædia confirms this: apparently he was an accomplished pianist.

[8] But it's obvious anyway that bus numbers are not numbers but rather strings. Otherwise how could we have a bus with a "number" like '7A'?

[9] Miniexercise: What is that number, and under which systems do those numerals represent it?

Cambridge News, 17:44, 21 Feb 2017
`http://www.cambridge-news.co.uk/news/cambridge-news/missing-number-children-fleeing`

The [10] story is not that there are children fleeing their care home by Cambridgeshire rocket—concerning tho' that is (they *must* be in a hurry); the story is rather that the number of such children has been mislaid:

> $|\{x : \text{child}(x) \wedge x \text{ is fleeing care in a Cambridgeshire rocket}\}|$
> has been mislaid.

The person in charge of data capture has left it on a train. Or in a rocket, perhaps.

2.9 Intension and Extension

The intension-extension distinction is an informal device but it is a standard one which we will need at several places. We speak of **functions-in-intension** and **functions-in-extension** and in general of **relations-in-intension** and **relations-in-extension**. There are also 'intensions' and 'extensions' as nouns in their own right.

Consider two properties of being *human* and being a *featherless biped*–a creature with two legs and no feathers. There is a perfectly good sense in which these concepts are the same (or can be taken to be, for the sake of argument: one can tell that this illustration dates from before the time when the West had encountered Australia with its kangaroos!), but there is another perfectly good sense in which they are different. We name these two senses by saying that 'human' and 'featherless biped' are the same **property-in-extension** but are different **properties-in-intension**.

A more modern and more topical illustration is as follows. A piece of code that needs to call another function can do it in either of two ways.

> If the function being called is going to be called often, on a restricted range of arguments, and is hard to compute, then the obvious thing to do is compute the set of values in advance and store them in a look-up table in line in the code.
>
> On the other hand if the function to be called is not going to be called very often, and the set of arguments on which it is to be called cannot be determined in advance, and if there is an easy algorithm available to compute it, then the obvious strategy is to write code for that algorithm and call it when needed.

In the first case the embedded subordinate function is represented as a function-in-extension, and in the second case as a function-in-intension.

Functions-in-extension are sometimes called the **graph**s of the corresponding functions-in-intension: the graph of a function f is $\{\langle x, y\rangle : x = f(y)\}$, where we write '$\langle x, y\rangle$' for the ordered pair of x and y. One cannot begin to answer

Provide forward reference

[10] Thank you, Ted Harding!

exercise 39 on page 77 unless one realises that the question must be, "How many binary relations-*in-extension* are there on a set with n elements?" (There is no answer to "how many binary relations-in-intension ...?" Explain to the Hypothetical Other why this is so.)

One reason why it is a bit slangy is captured by an *aperçu* of Quine's: "No entity without identity". What this *obiter dictum* means is that if you wish to believe in the existence of a suite of entities—numbers, ghosts, functions-in-intension or whatever it may be—then you must have to hand a criterion that tells you when two numbers (ghosts, functions-in-intension) are the same number (ghost, etc.) and when they are different numbers (ghosts, etc). We need *identity criteria* for entities belonging to a suite if we are to reason rigorously about those entities. And sadly, although we have a very robust criterion of identity for functions-in-extension, we do not yet have a good criterion of identity for functions-in-intension. Are the functions-in-intension $\lambda x.x + x$ and $\lambda x.2 \cdot x$ two functions or one? Is a function-in-intension an algorithm? Or are algorithms even more intensional than functions-in-intension?

[Ooops, I mentioned λ-calculus there before telling you what it is. $\lambda x.F(x)$ is the function that, when given x, returns $F(x)$. (You may have seen the notation '$f : x \mapsto F(x)$' too.) When we apply a function $\lambda x. \ldots$ to an argument a we knock the 'λx' off the front and replace all the 'x's in the dots by 'a's. Thus $\lambda x.x^2$ applied to 2 evaluates to $2^2 = 4$.]

The intension-extension distinction turns up nowadays connection with the idea of evaluation. In recent times there has been increasingly the idea that intensions are the sort of things one *evaluates* and that the things to which they evaluate are extensions. Propositions evaluate to truth-values. Truth-values (`true` and `false`) are propositions-in-extension.

We do need both. Some operations are more easily understood on relations-in-intension than relations-in-extension (composition for example) Ditto ancestral ancestral??

Properties-in-extension are just sets. Relations-in-extension and functions-in-extension are sets of tuples

2.10 Semantic Optimisation and the Principle of Charity

When a politician says "We have found evidence of weapons-of-mass-destruction programme-related activities", you immediately infer that that have *not* found weapons of mass destruction (whatever they are). Why do you draw this inference?

Well, it's *so* much easier to say "We have found weapons of mass destruction" than it is to say "We have found evidence of weapons-of-mass-destruction programme-related activities" that the only conceivable reason for the politician to say the second is that he won't be able to get away with asserting the first. After all, why say something longer and less informative when you can say

something shorter and more informative? We here, doing a course in Discrete Mathematics, will tend to see this as a principle about maximising the amount of information you convey while minimising the amount of energy you expend in conveying it. We will be doing a teeny weeny bit of optimisation theory (in chapter 7) but only a very teeny-weeny bit (just enough for you to develop a taste for it) and certainly not enough to come to grips with all the complexities of human communication. But it's not a bad idea to think of ourselves as generally trying to minimise the effort involved in conveying whatever information it is that we want to convey.

Quine used the phrase "The Principle of Charity" for the assumption one makes that the people one is listening to are trying to minimise effort in this way. It's a useful principle, in that by charitably assuming that they are not being unneccessarily verbose it enables one to squeeze a lot more information out of one's interlocutors' utterances than one otherwise might, but it's dangerous. Let's look at this more closely.

Suppose I hear you say

> We have found evidence of weapons-of-mass-destruction programme-related activities. (1)

Now you *could* have said

> We have found weapons of mass destruction. (2)

. . . but you didn't—*even though it's shorter*. Naturally I will of course put two and two together and infer that you were not in a position to say (2), and therefore that you have *not* found weapons of mass destruction. However, you should notice that (1) emphatically does *not* imply that

> We have *not* found weapons of mass destruction. (3)

After all, had you been lucky enough to have found weapons of mass destruction then you have most assuredly found evidence of weapons-of-mass-destruction programme-related activities: the best possible evidence indeed. So what is going on?

What's going on is that (1) does not imply (3), but that (4) does!

> We have chosen to say "We have found evidence of weapons-of-mass-destruction programme-related activities" instead of "We have found weapons of mass destruction ". (4)

Notice that (1) and (4) are not the same!

Now the detailed ways in which this optimisation principle is applied in ordinary speech do not concern us here—beyond one very simple consideration. **I want you to understand this optimisation palaver well enough to know when you are tempted to apply it, and to lay off. The formal languages we use in mathematics and computer science are languages of the sort where this kind of subtle reverse-engineering of interlocutors' intentions is a hindrance not a help. Everything is to be taken literally.**

2.10.1 Overloading

Not quite the same as ambiguity.

+ on reals and on natural numbers are different operations. They look sort-of similar, because they obey some of the same rules, so there is a temptation to think are the same thing—and certainly to use the same symbol for them. A symbol used in this way is said to be **overloaded**, and it's not quite the same as the symbol being **ambiguous** because there is a connection of meaning between the two uses which there might not be when a symbol is being used ambiguously.

polymorphism?

Overloading is a way of being thrifty in our use of notation. The drawback is that it gets us into the habit of expecting to find ambiguities even in settings where there are none. This leads us to...

2.11 Fault-tolerant pattern-matching

My brother-in-law once heard someone on the bus say "My mood swings keep changing." He—like you or I on hearing the story—knew at once that what the speaker was trying to say was that they suffer from mood swings!

This is an example of something we do all the time. It's *fault-tolerant pattern matching*. There are things out there is the world that we need to recognise, for good or ill. Things we might want to eat or to mate with, things that might want to eat us. We need to be able to spot these things, and we need to be able to spot them even if they imperfectly presented to us on account of the signal to noise ratio being less than it should be. We need to be able to match the patterns that we see in the outside world to the template in our heads, and because real signals are noisy, we need to be tolerant of faults and noise. Hence the expression *fault-tolerant pattern matching*

Reinterpreting silly utterances like this so that they make sense is something that we are incredibly good at. And by 'incredibly good' I mean that this is one of the things we can do *vastly* better that computers do (in contrast to the things like multiplying 100-digit numbers in our head, which computers can do very much better than we can). In fact we are *so* good at it that nobody has yet quite worked out how we do it, though there is a vast literature on it, falling under the heading of what people in linguistics call "pragmatics". Interesting though that literature is I am mentioning it here only to draw your attention to the fact that learning to do this sort of thing *better* is precisely what we are *not* going to do. I want you to recognise this skill, and know when you are using it, in order not to use it *at all!*

Fault-tolerant pattern matching is very useful in everyday life but absolutely no use at all in the lower reaches of computer science. It is all too easy for fault-tolerant pattern matching to turn into *overenthusiastic* pattern matching—otherwise known as *syncretism*: the error of making spurious connections between ideas. A rather alarming finding in the early days of experiments on sensory deprivation was that people who are put in sensory deprivation tanks

start hallucinating: their receptors expect to be getting stimuli, and when they don't, they wind up their sensitivity until they start getting positives. Since they are in a sensory deprivation chamber, those positives are one and all spurious . . . we have been *overinterpreting*.

2.11.1 Overinterpretation

Why on earth might we *not* want to use it?? Well, one of the differences between the use of symbols in mathematics (*e.g.* in programming languages) and the use of symbols in everyday language is that in maths we use symbols formally and rigidly and we suffer for it if we don't. If you write a bit of code with a grammatical error in it the O/S will reject it: "Go away and try again." One of the reasons why we design mathematical language (and programming languages) in this po-faced fault-intolerant way is that that is the easiest way to do it. Difficult though it is to switch off the error-correcting pattern-matching software that we have in our heads, it is much more difficult still to discover how it works and thereby emulate it on a machine—which is what we would have to do if we were to have a mathematical or programming language that is fault-tolerant and yet completely unambiguous. In fact this enterprise is generally regarded as so difficult as to be not worth even attempting. There may even be some deep philosophical reason why it is impossible even in principle: I don't know.

Switching off our fault-tolerant pattern-matching is difficult for a variety of reasons. Since it comes naturally to us, and we expend no effort in doing it, it requires a fair amount of self-awareness even to realise that we *are* doing it. Another reason is that one feels that to refrain from sympathetically reinterpreting what we find being said to us or displayed to us is unwelcoming, insensitive, and somehow not fully human, and that one will be told off for it. Be that as it may, you have to switch all this stuff off all the same. Tough!

So we all need some help in realising that we do it. I've collected in section 2.12 a few examples that have come my way. I'm hoping that you might find them instructive.

2.11.2 Scope ambiguities

Years ago when I was about ten a friend of my parents produced a German quotation, and got it wrong. I corrected him[11] and he snapped "All right, everybody isn't the son of a German Professor") (My father was Professor of German at University College London at the time). Quick as a flash I replied "What you mean is 'Not everybody is the son of a professor of German'.".

I was quite right. (Let's overlook the German professor/professor of German bit). He said that Everybody Isn't the son of a professor of German. That's not true. Plenty of people are; I am, for one. What he meant was "Not everybody is . . . ". It's the difference between "$(\forall x)(\neg \ldots)$" and "$\neg(\forall x)(\ldots)$"—the difference is real, and it matters.

[11] I was a horrid child, and I blush to recall the episode.

The difference is called a matter of **scope**. 'Scope'? The point is that in "$(\forall x)(\neg \ldots)$" the "scope" of the '$\forall x$' is the whole formula, whereas in '$\neg(\forall x)(\ldots)$' it isn't.

For you, the moral of this story is that you have to identify with the annoying ten-year old rather than with the adult that he annoyed: it's the annoying 10-year-old that is your rôle model here!

It is a curious fact that humans using ordinary language can be very casual about getting the bits of the sentence they are constructing in the right order so that each bit has the right scope. We often say things that we don't literally mean. ("Everybody isn't the son of ..." when we mean "Not everybody is ...") On the receiving end, when trying to read things like $(\forall x)(\exists y)(x \text{ loves } y)$ and $(\exists y)(\forall x)(x \text{ loves } y)$, people often get into tangles because they try to resolve their uncertainty about the scope of the quantifiers by looking at the overall meaning of the sentence rather than by just checking to see which order they are in!

First occurrence of the word 'quantifier'. Should explain it

All that glisters is not gold

Every Frenchman is not racist.

Exercise 17. *Match up the formulæ on the left with their English equivalents on the right.*

(i) $(\forall x)(\exists y)(x \text{ loves } y)$	*(a) Everyone loves someone*
(ii) $(\forall y)(\exists x)(x \text{ loves } y)$	*(b) There is someone everyone loves*
(iii) $(\exists y)(\forall x)(x \text{ loves } y)$	*(c) There is someone that loves everyone*
(iv) $(\exists x)(\forall y)(x \text{ loves } y)$	*(d) Everyone is loved by someone*

In the real world people make mistakes and say things that aren't exactly what they mean ("Everybody isn't the son of a German Professor") so listeners have to get quite good at spotting these errors and correcting them. So good, in fact, that we don't notice we do it. In mathematics (and, in particular, with programming languages) errors of the kind we are so skillful at correcting are never allowed to occur in the texts in the first place, so there is no need to have lots of clever software to detect and correct them. The fault-tolerant pattern-matching skill is no longer an asset and its deployment merely distracts us from the task of reading the formula in question. The result is that when we encounter a formula with nasty alternations of quantifiers and tricky scoping (such as the Pumping Lemma from Languages-and-Automata which is lying in wait for you even as we speak) we think "This looks ghastly; it can't be what he means. Life isn't that bad: let's reach for the rescoping software" whereas what we should be doing is just trying to read it as it is. Sadly life—or at any rate the Pumping Lemma—really is that bad! The Pumping Lemma is less than completely straightforward to read even with the best of intentions (it has more quantifiers in it than we are used to) and attempting to read it without first switching off your rescoping software is a sure recipe for disaster.

2.12 Howlers of Overinterpretation

$\lambda x.\lambda y.2$

What is $\lambda x.\lambda y.2$? Overinterpretation will probably make you think this should simplify to 2; it doesn't. It really just is the function whose constant value is the function whose constant value is 2. What happens if you apply this function to 3? It's actually idiotically simple. It's the result of applying to 3 the function whose constant value is the function whose constant value is 2. And please do not make the mistake of thinking that the function with constant value 2 (the one that returns 2 whatever it is given as argument) is the same as the number 2. There is an important difference between a pint of Guinness and the magic Guinness glass (given by the Leprichaun to the Irishman who released him from a bottle wherein he'd been trapped since the Bronze Age) that automatically refills itself with Guinness every time anyone drinks from it. This difference is not *quite* the same as the difference between 2 and $\lambda x.2$ but it might help to remind you of that difference.

We do need to distinguish between an object and the unary functions whose constant value is that object. However I don't want to think about a nullary function (a function with no input) whose [constant] value is x: that would start to sound too much like mediæval theology. Nevertheless you might need to think about this kind of thing in the years to come.

The square of a relation

What is the square of the $<$ relation on $\mathbb{N}$? Well, one thing it ain't is $\{\langle x, y\rangle : x^2 < y^2\}$, which is the answer one of my students gave once. You get into this mess if you forget what the square of a relation is, and fault-tolerantly match to something you do know, such as squaring of numbers[12].

Actually it's a perfect example of the kind of mess you can get into if—before taking on board the idea of overinterpretation—you *free-associate* rather than actually *think*. You have to learn how not to overinterpret—and to be *born again* as a mathmo/compsci—before it is safe to free-associate like this.

The Power Set of the empty set

The *Power set* of a set is the set of all its subsets.

The empty set is the set with no elements. We write it as '$\emptyset$' or occasionally as '{}'. I say *the* empty set because there is in fact only *one* empty set. There is the criterion of identity for sets: two sets are the same set if they have the same members. (Not true for multisets or lists for example). So what is the power set of the empty set? Let's take this question slowly, in stages, and answer it carefully by reading *entirely literally* all the definitions we need to refer to.

[12] If you want to know what the square of a relation is, it's the result of composing (see section 3.2.1) a relation with itself. The square of the parent relation is the grandparent relation.

Recall that the power set, $\mathcal{P}(X)$, of a set X is the set of all subsets of X. Y is a subset of X (written '$Y \subseteq X$') if everything that is in Y is in X. So in order to devise the power set of the empty set we are going to open up a bag and put into it everything we can find that is a subset of the empty set.

Q: So what are the subsets of $\emptyset$?

Were you about to say "none"? If you were about to say that, then I want to smack your wrist. *Obviously* the empty set is a subset of itself–just look at the definition!! So why did you leave it out? Because it didn't sound like a sensible answer. Why didn't it sound like a sensible answer? Because somewhere in your mind is the unspoken assumption that if I ask you for the something-or-others of X, you should come up with something *new*. "He can't be wanting me to mention X co's he already knows *that*." But recall the definition of 'subset of': $x \subseteq y$ holds precisely when everything that is in x is also in y. So every set is (trivially) a subset of itself. So in particular the empty set is a subset of itself.

Very well, we are agreed there is one subset of $\emptyset$, namely $\emptyset$ itself. Also, by extensionality (see page 37 for a definition of extensionality) it is the *only* subset. (That bit, at least, is unproblematic.)

Q: OK, So what *is* the power set of the empty set?

Were you about to say $\emptyset$? If so I'm going to smack your wrist yet *again*. You were probably thinking something like "...the empty set is $\{\}$ so the set containing the empty set must be $\{\{\}\}$ and the curly brackets can't be doing anything so that must be the same as $\{\}$". This mistake arises from your overinterpretation of data that you feel to be suspect, namely $\{\{\}\}$. But if you are careful, you will see that it isn't suspect at all, and it won't be hard to explain this to yourself. The set that contains all the subsets of the empty set has as one of its members (its sole member as it happens) the empty set. So it isn't empty! So it *can't* be the same as the empty set. (This uses the Principle of the Indiscernibility of Identicals. Look it up.)

So the mistake of thinking that $\mathcal{P}(\emptyset) = \emptyset$ arises from thinking that the expression '$\{\emptyset\}$' is a bit of suspect data to which you need to apply your fault-tolerant pattern-matching software. The idea that '$\{\emptyset\}$' is a bit of suspect data is a separate mistake that deserves an analysis of its own. The trap is the trap of thinking that because the empty set *has nothing inside it* then it *actually isn't there at all*. Why do you think this? Because you are not happy with the idea of a set being *empty*. But—I put it to you—there is nothing any odder about the idea of an empty *set* than there is about the idea of an empty *folder*, or an empty *file*. It's no odder than the idea that zero is an integer. "How many strawberries have you got in that punnet?" "None, sadly!". Once you are thinking about sets properly this difficulty goes away.

duplicates material on p. **??**

The Power Set of $\{1, 2, 3\}$

I once had a student who, when asked in an exam to write down all the subsets of $\{1, 2, 3\}$, supplied only $\{1, 2\}$, $\{1, 3\}$ and $\{2, 3\}$. My guess is that

- She omitted $\{1, 2, 3\}$ on the grounds that 'subset' probably meant 'proper subset'. We saw this mistake earlier;

where?

- She omitted the singleton subsets because she was probably thinking something like "Why would anyone want to write down '{1}'? That's silly. Anyone writing that down probably really means '1', and that isn't a set, so I can leave it out".

- She left out the empty set because she didn't think it was there. We've seen this too.

Miniexercise: So just what exactly is the power set of $\{1, 2, 3\}$, Dear Reader?

Affirming the consequent

Years ago I was teaching elementary Logic to a class of first-year law students, and I showed them this syllogism:

> "If George is guilty he'll be reluctant to answer questions; George is reluctant to answer questions. Therefore George is guilty."

Then I asked them: Is this argument valid? A lot of them said 'yes'.

We all know that an obvious reason—the first reason that comes to mind—why someone might be reluctant to answer questions is that they might have something to hide. And that something might be their guilt. So if they are reluctant to answer questions you become suspicious at once. Things are definitely not looking good for George. Is he guilty? Yeah—string him up!

But what has this got to do with the question my first-years were actually being asked? Nothing whatever. They were given a premiss of the form $P \to Q$, and another premiss Q. Can one deduce P from this? Clearly not. Thinking that you can is the fallacy of *affirming the consequent*.[13]

There are various subtle reasons for us to commit this fallacy, and we haven't got space to discuss them here. The question before the students in this case was not: do the premisses (in conjunction with background information) give evidence for the conclusion? The question is whether or not the inference from the premisses to the conclusion is logically valid. And that it clearly isn't. The mistake my students were making was in misreading the question, and specifically in misreading it as a question to which their usual fault-tolerant pattern-matching software would give them a swift answer.

[13] Explain the terminology. Affirm the antecedent, infer the consequent. Need to explain antecedent and consquent and contrapositive. Perhaps a glossary at the end.

Cartesian Product with the Empty Set

$A \times B$ (the "cartesian product of A and B") is the set of all ordered pairs whose first component is in A and whose second component is in B. So $A \times \emptyset$ is the cartesian product of A (a set) with the empty set, which is to say the set of all ordered pairs whose first component is in A and whose second component is in the empty set.

What does $A \times \emptyset$ actually turn out to be?[14] Do the sensible thing: try to form ordered pairs whose first components are in A and whose second components are in $\emptyset$. So you pick up a member of A to put into your *chase* (that's the thing printers hold in their hand and into which they put bits of type when setting something up in type) and then you reach for a member of $\emptyset$. Ouch!

There is a danger of getting into a tangle by trying to find the correct description of the set of *defective* ordered pairs: things that should have become ordered pairs but never made it because—altho' they have a first component in A—they somehow lack a second component. $A \times \emptyset$ certainly *looks like* the obvious place to put these discards and offcuts. You might well think that such a discard-or-offcut is just a member of A, and conclude that $A \times \emptyset$ is A.

However there are *no* ordered pairs whose first component is in A and whose second component is in $\emptyset$: any attempt to assemble one fails. A long way of saying this is to say that the set of all such pairs—namely $A \times \emptyset$—is $\emptyset$, and that of course is the correct answer.

What do you do with the discards and offcuts? It doesn't matter. It's the ordered pairs you are interested in. Discards and offcuts? Discard them! You certainly don't put them into $A \times \emptyset$.

(When you encounter *regular languages* later you will meet the notation 'LM' for the set of strings consisting of a string from L `cons`ed onto the front of a string from M. What happens if L is empty? What happens if L contains only the empty string?)

Is the Identity Relation a Function?

> *"Well, functions give you back values when you feed them arguments. The identity relation obviously doesn't do anything so it can't be a function!"*

Is that what you were thinking? Shame on you! *Of course* the identity relation is a function. Look at the definition of a function: each argument is related to one and only one value. The identity relation relates each thing to one and only one thing, namely itself! Duh!

Is the Identity Relation a Partial Order?

Lots of people say: 'no'! They think that because it doesn't order things then it can't be a partial order. However, if you read the definitions you will see that

[14] one might like to frame this question as "What does $A \times \emptyset$ evaluate to?

it is.

Can a Relation be both Symmetrical and Antisymmetrical?

> "*Well, obviously not, because the words sound as if the conditions are mutually contradictory. On the other hand, why would they be asking me if it was that easy? Er . . . is this a multiple choice question . . . ? Will I be penalised for guessing? . . . Can I ask a friend?*"

How about just answering the question? It might be quicker!

Suppose R is both symmetrical and antisymmetrical. Then, whenever x is related to y by R, y is related to x by R—by symmetry. But if x is related to y and y to x, then $x = y$ by antisymmetry. So if x is related to y, $x = y$. So R must be a subset of the identity relation. So perhaps the identity relation itself might be both symmetrical and antisymmetrical, and so indeed it turns out.

And that is about all you can say!

Is the Empty Relation Transitive?

I have had students get in a tangle over the question whether or not the empty relation is transitive[15]. Or over the question of whether or not the relation $\{\langle 1, 2\rangle\}$ is transitive. It's the same tangle. The tangle is this: for a relation to be transitive, it's necessary for it to contain the ordered pair $\langle x, z\rangle$ whenever it contains the ordered pairs $\langle x, y\rangle$ and $\langle y, z\rangle$. "But what if it doesn't contain any ordered pair?" they wail. Or "What if it contains an ordered pair $\langle x, y\rangle$ but no pair $\langle y, z\rangle$?" This is overinterpretation. Nobody said it in order to be transitive it had to contain an ordered pair $\langle x, y\rangle$ or had to contain pairs $\langle x, y\rangle$ and $\langle y, z\rangle$. Merely that **if** it contained pairs $\langle x, y\rangle$ and $\langle y, z\rangle$ **then** it would have to contain the ordered pair $\langle x, z\rangle$. If it doesn't satisfy the antecedent of the conditional then the condition is trivially satisfied. It's mechanical to check that the empty relation and the relation $\{\langle 1, 2\rangle\}$ are transitive. Ian Stewart's example *If you pick a guinea pig up by its tail its eyes fall out*—is true. Conditionals whose antecedents are false are vacuously true: in the nature of things these conditionals are unlikely to be *useful* but that doesn't make the *false*.

Haven't defined conditional or antecedent

You only get into a tangle if you try to be too clever, and overinterpret.

Coda

If you get these questions wrong it's almost certainly not because you are ignorant or stupid, but because you are approaching them the wrong way.

[15] See section 3.2.4 if you do not yet know what *transitive* means.

Chapter 3

Sets and relations

which way round to write relational composition, and which way round to write pairs in functions

That was an introductory pep talk. Now we start on some mathematics!

Sets are extensions. Two sets with the same members have the same set. This is the *axiom of extensionality* for sets.

One could say that sets are properties-in-extension. The properties *human* and *featherless biped* (see page 26) are true of the same things but are not the same property. Lists and multisets are extensional too, but we have to be careful about how we express this fact. The lists $[1;3]$ and $[3;1]$ have the same members but are not the same list. For two lists to be the same list they not only have to have the same elements but have to have them in the same *slots*. For two multisets to be the same multiset it is necessary not only that they have the same members but that they have them with the same multiplicity.

	ordered	not ordered
repetitions allowed	lists	multisets
no repetitions	wellorderings	sets

Multisets in brief

The two most natural examples of things-that-we-want-to-think-of-as multisets are (i) the set of roots of a polynomial; and (ii) the set of factors of a natural number. The equation $x^2 - 3x + 2 = 0$ has two solutions, 1 and 2. That is to say, the set of its solutions is $\{1, 2\}$. The equation $x^2 - 2x + 1$ has two solutions, but they are both 1, so the set of its solutions is the set $\{1\}$. But thinking of the collection of answers as a *set* conceals the fact that (in some sense!) there are *two* roots, which just happen to be the same. A better way of presenting the same information would be to say that that the collection of roots is the *multiset* $\{1, 1\}$. What this means of course is that the appropriate datatype for the collection of solutions of $x^2 - 3x + 2 = 0$ is `multiset` not `set`. What about the set of prime factors of 60? That is the multiset $\{2, 2, 3, 5\}$. Notice that the

collection of roots-of-a-polynomial, or of prime-factors-of-a-natural-number is *always* best thought of as a multiset rather than as a set, *even if every member has multiplicity one!* The multiset $\{1, 2\}$ is not the same object as the set $\{1, 2\}$, any more than the list [1] is the same as the set $\{1\}$.

Although we are not going to see any more uses of multisets here, they are a useful data structure and you should not forget about them entirely. My chief reason for mentioning them here is to bring out into the open the range of different datatypes that may come your way the better to understand what is going on. When you say that $x^2 - 4x + 4$ has two roots, the extensional data object whose cardinality is two is a multiset not a set. You need to always be clear about what you are dealing with[1].

There is no standard notation for lists and multisets, though some programming languages (`ML` for example) use square brackets for lists with semicolons as delimiters. Thus $[1; 2; 4]$ is the list whose elements are 1,2 and 4 *in that order*. (Why did I say *elements* of the list $[1; 2; 4]$ but *members* of the set $\{1, 2, 4\}$? I'm not sure: I don't think there is any significance to it.)

Pure Set Theory

There is a discipline of Pure Set theory, that studies sets that can only have other sets as members, but mostly we will be interested only in sets whose members are things other than sets, and which are therefore of more immediate interest. Sets of numbers, or sets of ordered pairs, or sets of matrices, or sets of sets of matrices. Generally we are not likely to be interested in sets-of-sets-of^n things for n more than about 2. The discipline of Pure Set Theory had a vogue in the twentieth century as a purported foundation of mathematics. The philosophers of the early twentieth century were very struck by the fact that it seemed to be possible (and it still does) to find (what compscis like us would call) *implementations* of all manner of mathematical objects into set theory. Ordered pairs, naturals, reals, complexes, functions etc etc. It was felt that the mere possibility of these implementations into set theory told us somehow what these various mathematical gadgets *were*. Thus there came the idea that set theory could be a foundation for mathematics. This idea has been a long time dying, and it's far from dead even now ... tho' it is on the way out.

That's not to say that the theory of pure sets is not an interesting and exciting branch of mathematics—it is. It's just that it doesn't do what is claimed for it. For us the point is that you don't need to know any.

While we are about it, there is a distinction Pure Set Theorists make which won't much concern us here. Some sets are *paradoxical*—such as the set of all sets that are not members of themselves. If you think about this you will tie yourself in a knot. Is it a member of itself or not? If it is, it isn't, and if it isn't it is!! Clearly it can't exist—but then I shouldn't have been talking about the **set** of all sets that aren't members of themselves. Set theorists tend to use the

[1] Say something about natural numbers as multisets of primes? Deals very nicely with hcf, lcm and divisibility—it's just $\subseteq$.

word **proper class** for dodgy collections like this one. This is because they use the word **class** in a noncommittal way to cover not only collections that are OK (which means the vast majority of them) but also the dangerous collections like the collection of all sets that aren't members of themselves. A proper class is a thing that's a bit like a set except that it's not allowed to be a member of anything—and that's because it's not really there! We also talk of **families** and **collections** when we want to be noncommittal. That explains why when we are considering a class that cannot be a set (like the Russell class $\{x : x \notin x\}$) we call it a *proper* class. This usage of the word 'proper' is like its usage in *proper* subset, which you may have seen earlier. (Miniexercise: what does "x is a proper subset of y" mean?)

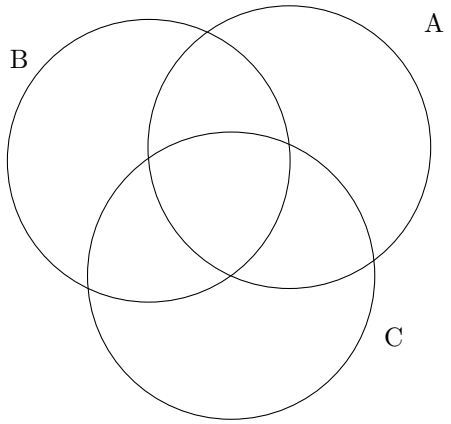
I am giving the paradoxes very short shrift in this material. They are very titillating, and although they do have connections with things you need to understand (Cantor's theorem and proofs of uncountability and—later on—the Unsolvability of the Halting Problem) they are of little direct relevance and they will most assuredly do your head in should you spend any time on them. There is a theory that it's not possession of language or tools or representational art that distinguishes *Homo sapiens sapiens* from *Homo sapiens neanderthalensis* (let alone the other species of the genus *Homo*) but the ability to lose sleep over the paradoxes. However you shouldn't make that a reason to study them. After all, if you're reading this, the chances are that you are already known to be a specimen of *Homo sapiens sapiens*.

3.1 Sets for Discrete Mathematics

There are hardly any theorems in set theory that we need to know, and, of those that we are going to prove, one of them (Cantor's theorem) is very easy. (In fact, one of the things that disconcerts first-year students is that there are no tangible *deliverables* in first-year Discrete Mathematics). What is going to take up most of our time is the task of coming to grips with the notation, which was never designed to make it easy for new entrants to feel at home! (In fact it was never *designed* at all.) The other theorem that I'm going to prove (the Inclusion-Exclusion Principle in section 3.3.5) isn't really very hard either, but if one tries to prove it properly one has to absorb a lot of notation, which makes working through it a very useful discipline. That's the peak, and the rising ground starts here.

3.1.1 Representing Sets Graphically

You have probably encountered Venn diagrams. They are a useful way of illustrating equalities and inequalities between unions, intersections and complements of sets. $A \cup B \cup C = \overline{\overline{A} \cap \overline{B} \cap \overline{C}}$ and suchlike.



Venn diagrams won't *prove* equalities and inequalities, but they can be a useful picture to draw if you are in doubt about them, and can sometimes help you visualise things and think them through. Of course if you are going to use them to check an equation or inequation, you have to draw your sets properly: the circle for A must overlap with the circle for B, and neither must be included in the other, lest you appear to have a proof that $A \subseteq B$ or *vice versa*. If you are drawing a picture to check an equation that mentions three sets A, B and C, you will want to draw the three circles so that all eight unions/intersections/differences[2] are present, and this can in fact be done. Here we come to our first cautionary tale. You can draw *three* circles so that all unions/intersections/differences are present, but you can't draw *four*. Try it and see. The appearance of the number 3 here is to do with the fact that the surface of the paper on which you are drawing your picture has two dimensions. If you want to draw four regions for A, B, C and D then they can't all be *convex*. (Type 'Anthony Edwards' (or 'A.W.F. Edwards') and 'Venn diagram' into your favourite search engine. Yes, and if you happen to be in Cambridge go and look at the windows in Caius dining hall.) If you want four convex regions showing all unions/intersections/differences then you have to go up a dimension. You can imagine four spheres in space intersecting in all possible ways. Although this fact is a very cute piece of Discrete Mathematics in its own right I am dragging it in here as an illustration to warn you about the way in which notations which are handy and attractive sometimes fail to give you the whole picture.

While we are about it there is another reason why you shouldn't think of Venn diagrams as the be-all and end-all of Set theory. Venn diagrams are no good for representing more than two levels of sets. One can indicate a point c inside a region called C but one cannot indicate members of c—should c be a

[2]Why are there eight? If you don't already know, this will be explained in section 3.2.8.

set in its own right—nor can one indicate things that C is a member of. This is a grave limitation on Venn diagrams as ways of illustrating facts in set theory. Examples of important constructs using three levels which therefore cannot be represented: $\bigcap X$, $\bigcup X$, $\mathcal{P}(X)$. (We define these in section 3.1.7 p 48.)

Notice that I have just—quite automatically—exploited a common convention that we use a lower case letter (in this case 'c') for a variable to range over elements of a set denoted by the corresponding uppercase variable (in this case 'C'). Some people use a convention that if we are to have a third level—things that C is to be a member of—we use a calligraphic font: $\mathcal{C}$.

Moral: Venn diagrams are fine, but don't allow the comfortable feelings you have about them to constrain your imagination when it comes to sets.

3.1.2 Notating Sets

The simplest of the many notations we have uses curly brackets in connection with commas: thus $\{1, 2, 4\}$ is the set whose members are 1, 2 and 4.

Please note: **these curly brackets are not mere punctuation**, as '(' and ')' are in English. They are part of the grammar. '{' and '}' cannot be replaced by '(' and ')' to make things easier to read. And they cannot be omitted without changing the meaning. Remember the discussion of the power set of the empty set on page 33.

There are actually two fundamentally different ways of notating sets. The system of the previous paragraph takes notations for the individual members of a set, puts them in a row, and places curly brackets round them. The result is a notation for the set whose members we have listed. The other notation for sets also uses curly brackets, but doesn't mention the members explicitly in the way we have just seen: instead it mentions them indirectly (in a way that assembler programmers might recognise as bearing a haunting similarity to *indirect addressing*). The notation

$$\{x : F\} \tag{3.1}$$

denotes the set of all things satisfying the condition F that we find after the colon. This piece of notation is a **set abstract**. Thus the set abstract

$$\{x : x = 2 \vee x = 4\}$$

denotes the set of all things that are equal to 2 or to 4—so we could have written it as

$$\{2, 4\}$$

Similarly the set abstract

$$\{x : x \in A \vee x \in B\}$$

has a shorter notation too, and one that you know: $A \cup B$. This indirect notation, using variables and the colon, enables us to write down notations for

infinite sets—in contrast to the first notation (curly brackets plus commas), which obviously enables us to point to finite sets only.

We saw the intension/extension distinction in section 2.9. One could say that the curly bracket notation in '$\{2, 3, 5\}$' is extensional and the notation in '$\{x : \phi\}$' is intensional, but this observation is impressionistic. If the reader finds it unhelpful (s)he is free to ignore it... the intension/extension distinction is always slightly slangy.

Colons and Vertical Bars

Some people write '$\{x|F\}$' for the set of things that are F. The vertical bar has uses already—we will write '$|x|$' for the number of things[3] in the set x, and '$|x|$' for the absolute value of a complex number x—indeed also the length of the vector x—and all these are uses you will most certainly encounter (there is yet another use: some people write '$f|X$' for the restriction of the function f to the set X, though this is probably an attempt to write '$f \restriction X$' when they haven't got the '$\restriction$' sign) and we don't want to introduce further confusion by giving it a fourth use. I shall adhere to the alternative notation that uses a colon: $\{x : F(x)\}$ as above. However you should be warned that many people prefer the notation with the vertical bar.

3.1.3 Curly Brackets

So far we have seen two styles of notation for sets:

(i) The *extensional* ("explicit") notation, as in '$\{a, b\}$' for the pair of a and b;

(ii) The *intensional* ("indirect") notation, as in '$\{x : x > 2\}$'.

There isn't a huge amount to say about the first notation, but the second needs quite a lot of discussion and explanation, because there are enhancements of it which put things other than variables to the left of the colon. These enhancements are in widespread use, and you will have to master them ... and they take a bit of getting used to!

The following expression turned up in an example sheet. Or was it a exam question(?) I forget. Anyway, it's a(n admittedly fairly extreme) example of the kind of notation you are going to have to be happy with. I'm going to gradually work through it, using it as a peg on which to hang various definitions.

$$\{2z \in \mathbb{N} \mid z \leq 5 \ \wedge \ 20/z \in \{w \in \mathbb{N} : w \leq z\}\} \tag{3.2}$$

Notice that in formula 3.2 the thing between the left '{' and the '|' isn't just a naked variable as it was in formula 3.1. This exploits two conventions which we explain in the two sections following, sections 3.1.4 and 3.1.5

[3] Make a note of this announcement, co's I am not planning to repeat it!

3.1.4 (Booleans to the left of a colon)

Often instead of writing things like

$$\{x : x \in A \wedge \Phi\}$$

(where Φ is some condition or other) we move the membership part of the condition to the left of the colon thus:

$$\{x \in A : \Phi\}$$

For example, where $A = \mathbb{N}$ and Φ is $(\exists y)(x = 2 \cdot y)$: instead of writing

$$\{x : x \in \mathbb{N} \wedge (\exists y)(x = 2 \cdot y)\} \tag{3.3}$$

we can write

$$\{x \in \mathbb{N} : (\exists y)(x = 2 \cdot y)\} \tag{1}$$

which means exactly the same thing. The difference is merely one of emphasis: the second notation suggests somehow that the natural numbers is a kind of context or environment for what happens after the colon. And, yes—in case you were wondering—this last thing is, indeed, the same as

$$\mathbb{N} \cap \{x : (\exists y)(x = 2 \cdot y)\} \tag{2}$$

This looks perverse, but there is reason to it, which we have just alluded to. We use it in situations where one feels that the set A provides a *context*. For example the expression $\{x : x^2 < 26\}$ denotes the set of all numbers whose square is less than 26. We might be interested in the set of all *real* numbers whose square is less than 26, in which case one would write

$$\{x : x^2 < 26 \wedge x \in \mathbb{R}\} \tag{3.4}$$

or one might be interested in the set of all natural numbers whose square is less than 26, in which case one would write

$$\{x : \ x^2 < 26 \wedge x \in \mathbb{N}\} \tag{3.5}$$

but—because in circumstances like this one usually is thinking of $\mathbb{R}$ or $\mathbb{N}$ as a *universe of discourse*, the place where it is happening—one would write these as

$$\{x \in \mathbb{R} \ : \ x^2 < 26\} \tag{3.6}$$

and

$$\{x \in \mathbb{N} \ : \ x^2 < 26\} \tag{3.7}$$

where the thing to the left of the colon is not a variable but a boolean with a free variable in it.

Before you go any further check your understanding by doing this exercise:

Exercise 18.
Write out the set in formula 3.7 in primary ("extensional") notation.
Why do I not ask you to write out the set from formula 3.6 in extensional notation?

There is a similar notation for the quantifiers: often one writes '$(\forall n \in \mathbb{N})(\ldots)$' instead of '$(\forall n)(n \in \mathbb{N} \to \ldots)$' and '$(\exists n \in \mathbb{N})(\ldots)$' instead of '$(\exists n)(n \in \mathbb{N} \wedge \ldots)$' and here, too, there is a suggestion that $\mathbb{N}$ is a kind of environment or local universe or context. We can use the same notational device in connection with λ-terms too. Thus, $\lambda x \in X.\{x\}$ is the function that sends members of X to their singletons. This function will reappear in the proof of Cantor's theorem in section 3.3.4.

Here is a live example from an actual example sheet

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

The thing to the left of the colon (well, the vertical bar) is a boolean, but not a boolean like '$x \in \mathbb{N}$'. It's an assertion that f is a function from $\mathbb{N}$ to $\{0,1\}$. This set abstract could more clearly be written as

$$\{f : (f : \mathbb{N} \to \{0,1\}) \wedge (\forall n \in \mathbb{N})(f(n) \leq f(n+1))\}$$

The convention my colleague was appealing to is that, if Φ and Ψ are both booleans with 'x' free, then $\{\Phi(x) : \Psi(x)\}$ is just $\{x : \Phi(x) \wedge \Psi(x)\}$. Notice too (and do not be alarmed by) the dual use of the colon!

This was on a first year example sheet, so you should aim to cope with this kind of notation. Incidentally, can you describe this set in words? How many members does it have?

3.1.5 Function symbols to the left of a colon!

We can even write formulæ (1) and (2) as

$$\{2y : y \in \mathbb{N}\} \tag{3.8}$$

This is possible because there is a convention that allows us to write

$$\{f(y) : y \in X\}$$

Explain double apostrophe notation here..?

to mean the set of values of the function f for aguments in X, and $\{2y : y \in \mathbb{N}\}$ is just a special case of this.

Thus we can write things like

$$\{x^3 \in \mathbb{N} \ : \ x^2 < 26\} \tag{3.9}$$

Careful! How do we know that 'x' ranges over $\mathbb{N}$?

Formula 3.9 denotes the set that you get from the set denoted by formula 3.7 by replacing every number in that set by its cube.

As we will see later, functions(-in-extension) are sets of ordered pairs. If we want a notation for the set of all ordered pairs satisfying a condition ϕ on its two components we would definitely prefer writing

$$\{\langle x, y\rangle : \phi(x, y)\}$$

to

$$\{p : (\exists x)(\exists y)(\text{fst}(p) = x \wedge \text{snd}(p) = y \wedge \phi(x, y))\}$$

and these two mean the same thing.

fst occurrence of 'fst'?

Index sets

Now, if $\{f(y) : y \in X\}$ is a set, so is

$$\bigcup\{f(y) : y \in X\}$$

and this is sometimes written

$$\bigcup_{y \in X} f(y)$$

and when we do this we often speak of X as an **index set**

say more about this

need lots more exercises here

$\in$-introduction and elimination

The last point the questionmaster was trying to get across is that $w \in \{x : \phi(x)\}$ is the same as $\phi(w)$. Being a member of the set of all green things is just the same as being a green thing[4].

Or rather, since the example is

$$20/z \in \{w \in \mathbb{N} \mid w \leq z\} \tag{3.10}$$

the point is that being a member of the set of green slimy things is the same as being green and slimy. Thus (by $\in$-elimination) formula 3.10 is just the same as

$$20/z \in \mathbb{N} \ \wedge \ 20/z \leq z \tag{3.11}$$

This will simplify formula 3.2 to

$$\{2z \in \mathbb{N} : z \leq 5 \ \wedge \ 20/z \in \mathbb{N} \ \wedge \ 20/z \leq z\} \tag{3.12}$$

So we ascertain what

[4]The study of proofs as mathematical objects is beyond the scope of these notes, but you will probably encouinter it in a Logic course in your second year. If $w \in \{x : \phi(x)\}$ is the same as $\phi(w)$ then we have a rule of inference that says we can deduce $w \in \{x : \phi(x)\}$ from $\phi(w)$. We will call this rule the *rule of $\in$-introduction* since it introduces the symbol '$\in$' into the conclusion. The equivalence can be used to draw an inference in the opposite direction: the rule of $\in$-*elimination*, whereby in inferring $\phi(w)$ from $w \in \{x : \phi(x)\}$ we have discarded an occurrence of '$\in$'. Hence the title of this subsection.

$$\{z \in \mathbb{N} : \ z \leq 5 \ \wedge \ 20/z \in \mathbb{N} \ \wedge \ 20/z \leq z\} \tag{3.13}$$

is and then multiply everything in it by 2. A simple case analysis shows that the only natural number z below 5 such that $20/z \leq z$ is 5. So the set of 3.13 is $\{5\}$. (Not 5 itself! That's another point the questionmaster was trying to make!)

Finally multiply everything in $\{5\}$ by 2 to obtain $\{10\}$.

It gets worse

You will even see things like

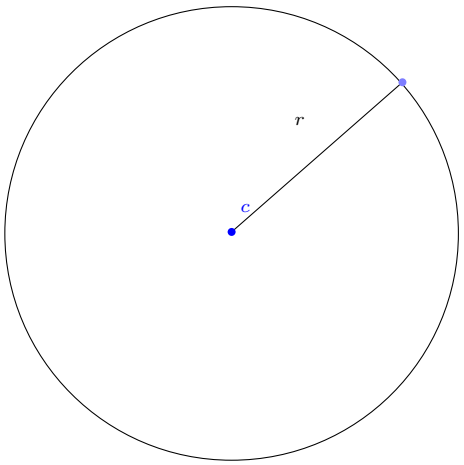
$$\{\pm n : \Phi(n)\} \tag{☠}$$

There are more casual notations to be seen, but I won't expose you to them. The general idea at this stage is: do not be casual; you shouldn't use slang until you have learned to talk proper!

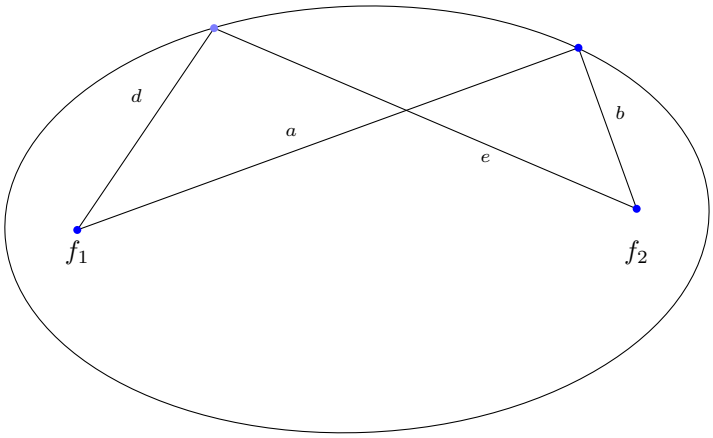
3.1.6 Exercises

Exercise 19. *Give two set abstracts for the set of natural numbers that are perfect squares, one using the existential quantifier $\exists$ and the other using the trick introduced in this section of moving some information to the left of the colon*

Exercise 20. *A circle is the set of points in the plane (that's $\mathbb{R}^2$ for the moment) that are a fixed distance from the centre. Write down a set abstract denoting a circle. It will have two free variables in it ("parameters"). (It is customary to write $d(x, y)$ for the distance between points x and y.)*



An ellipse is the set of points in the plane the sum of whose distances from two points (the foci—f_1 and f_2 in the picture) is a constant. (In the picture above $a+b=c+d$). Write down a set abstract denoting an ellipse. It will have three free variables in it ("parameters").



degrees of freedom?

Exercise 21. *A parabola is the set of points the sum of whose distances from a given point and a given line is a constant. Write down a set abstract denoting a parabola.*

3.1.7 Variables and binders

Some people make explicit the presence inside F of the variable that is to the left of the semicolon, the business variable (or *eigenvariable* if you want a fancy name for it). Thus you might see '$\{x : F(x)\}$' or (since some people don't) '$\{x : F\}$'. Both of these are shorthand for set abstractions. In the first case you are being told that the variable 'x' really does appear inside the expression that F represents, and in the second case you aren't being told this. But usually you will be given the formula in full rather than a shorthand for it.

Several subtleties arise at this point.

1. (Vacuous quantification and abstraction)

   What does '$\{x : A\}$' mean, if the variable 'x' does not appear in A at all? Well, it means either the empty set or the universe, depending on what A is. After all, if x is not mentioned by A, the yes/no answer to "Is x in the set?" does not depend on x; they're either all in, or none.

2. (alpha-conversion). There is no difference between $\{x : F(x)\}$ and $\{y : F(y)\}$. Those of you who have done a bit of logic will recognise this as the same phenomenon that $(\forall x)(F(x))$ and $(\forall y)(F(y))$ are in some sense the same formula. You may feel that this is a slight bug in the design of our language: it makes distinctions that we don't need. We will find the same

phenomenon arising in lambda calculus, and there there is a known alternative which does not have this bug: combinatory logic. There is no space to discuss these matters here. Sufficient unto the day is the evil thereof.

There is a connection here with two other ideas which are probably new to you with this course. Quantifiers and λ-terms. You can take a formula with an 'x' free in it (such as '$x > 3$') and prefix it by a quantifier, like '$\forall$' or '$\exists$'. Or you can take a term with 'x' free in it (such as '$x + 5$') and prefix it with 'λx'. This gives you expressions like $\{x : x > 3\}$ or $(\forall x)(x > 3)$ or $\lambda x.(x + 5)$ in which the variable 'x' is no longer free. So the curly-brackets-in-pairs, the quantifiers, and the letter λ are all called **binders**. Often having a word enables you to keep track of the connection of ideas.

'free' variable

Some exercises here

Exercise 22.
Can you explain the difference between '$f : x \mapsto y$' and '$f : x \to y$'?

Have a look at section 2.12.

Of course the thing to the right of the dot might be another lambda term. I shall also adhere to the universal practice of writing '$\lambda xy.(\ldots)$' for '$\lambda x.(\lambda y.(\ldots))$'. Lambda calculus is a great improvement on the old system, under which people would write things like '$y = F(x)$' and '$y = x^2$', relying on an implicit convention that—where 'x' and 'y' are the only two variables used—then y is the output (the vertical axis used to be called "ordinate") and x is the input (the horizontal axis used to be called "abcissa"). This convention—and others like it—have served us quite well, but in the information technology age, when one increasingly wants machines to do a lot of the formula manipulations that used to be done by humans, it turns out that lambda notation and notations related to it are more useful. Another reason for using lambda calculus rather than the assumption that x is an input and y an output is that life is simpler and syntax is easier to describe if all variables are equivalent in the sense that they don't come equipped with baggage. We want $(\forall x)(\Phi)$ to mean the same as $(\forall y)(\Phi)$ and we want this interchangeability to apply across the board. We will see more of λ-calculus in section 3.3.2.

Say something about β-reduction.

Sumset, Power set etc

We will now briefly go over some notations that you are probably familiar with, just in case you aren't! You know what '$\subseteq$' means. Care to guess what '$\supseteq$' means? (It's pronounced 'superset'.)

There are other notations along these lines that you will need to know if you do not know them already.

Sumset: $\bigcup x := \{y : (\exists z)(y \in z \wedge z \in x)\}$; and
Intersection $\bigcap x := \{y : (\forall z)(z \in x \to y \in z)\}$.
$\mathcal{P}(x)$ is the **power set** of x: $\{y : y \subseteq x\}$.
Set difference: $x \setminus y$ is the set of things that are in x but not in y.
The symmetric difference: $x \Delta y$, of x and y, is the set of things in

one or the other but not both: $(x \setminus y) \cup (y \setminus x)$. This is sometimes written '`XOR`'.

Exercise 23. *(Computer Science Tripos 2009 paper 1 question (c))*
If $\bigcup A \cap \bigcup B = \emptyset$ *does it follow that* $A \cap B = \emptyset$*?*

Definition 1.
A **partition** Π *of a set* x *is a family of pairwise disjoint nonempty subsets of* x *that collectively exhaust* x*, so that* $\bigcup \Pi = x$

The members of the partition are called its **pieces**.

If Π_1 *and* Π_2 *are partitions of* x*, we say that* Π_1 **refines** Π_2 *if every piece of* Π_1 *is a subset of a piece of* Π_2.

Do not confuse this capital 'Π' with the capital 'Π' used to denote the product of all numbers in a set!

Exercise 24. *Which of the sets in the left column are partitions, and of which set(s) in the right column (if any) are they partitions?*

(i) $\{\{1\},\{2,3\},\{1,2\}\}$	*(a)* $\{1,2,3\}$
(ii) $\{\{1\},\{2,4\},\{3\}\}$	*(b)* $\{1,2,3\}$
(iii) $\{\{1,2\},\{2,4\},\{3\}\}$	*(c)* $\{1,2,3,4\}$
(iv) $\{\{1,3\},\{2,4\}\}$	*(d)* $\{1,2,4,3\}$
(v) $\{\{1\},\{3\},\{2,4\}\}$	*(e)* $\{1,3,2,4\}$
(vi) $\{\{1\},\emptyset,\{3\},\{2,4\}\}$	*(f)* $\{2,1,3,4\}$
(vii) $\{\{1\},\{5\},\{3\},\{2,4\}\}$	*(g)* $\{2,1,3,4\}$

Exercise 25. *(*)*
List all partitions of $\{a,b,c\}$*. (You might find it helpful to draw a picture of each —a kind of Venn Diagram.)*

3.2 Relations and Functions

Relations-in-extension and functions-in-extension are sets of tuples. What is a tuple? An n-tuple is just a list of length n. [5]

For the rest of this section we are going to think of relations generally as relations-in-extension: sets of tuples. Quite a lot of what we are going to do makes sense when done to relations-in-intension as well, but the default will be that all relations under discussion are relations-in-extension.

The n-tuples that will be most important to us are those where $n = 2$. We call these tuples **ordered pairs**. An ordered pair has two different—we used the word *slot* earlier. If p is a pair, we can write `fst`(p) and `snd`(p) for the first and second components of p (the things in the first and second slots of

This seems to be the place where these things are defined

[5] Some programming languages—`ML` for example—distinguish between n-tuples and lists of length n. We won't make this distinction, and we don't at this stage need to go into why they do.

p). We will also write '$\vec{x}$' for the n-tuple '$\langle x_1 \dots x_n \rangle$'. This is because I write ordered pairs, triples, and so on with angle brackets: $\langle x, y \rangle$. However, the world being the deeply flawed place it is, you will find people using round brackets for this, and writing the ordered pair of x and y as '(x, y)'. I avoid it because this notation is used for several other things already, but not everybody feels like me, and you must not panic when you see it notated differently.

- (x, y) might be the open interval $\{z \in \mathbb{R} : x < z < y\}$.

  While we are about it $[x, y]$ is (in the same tradition) the closed interval $\{z \in \mathbb{R} : x \leq z \leq y\}$, and $(x, y]$ is (in the same tradition) the half-open interval $\{z \in \mathbb{R} : x < z \leq y\}$, and $[x, y)$ similarly.

- (x, y) might also be the permutation (sometimes called a *transposition*) that swaps x and y and fixes everything else.

- (x, y) is also the Highest Common Factor of x and y.

All this is in small print because mostly you don't really need it: it's here only to put things in context. However when we come to chapter 4 you will encounter the usage (x, y) for the highest common factor of x and y.

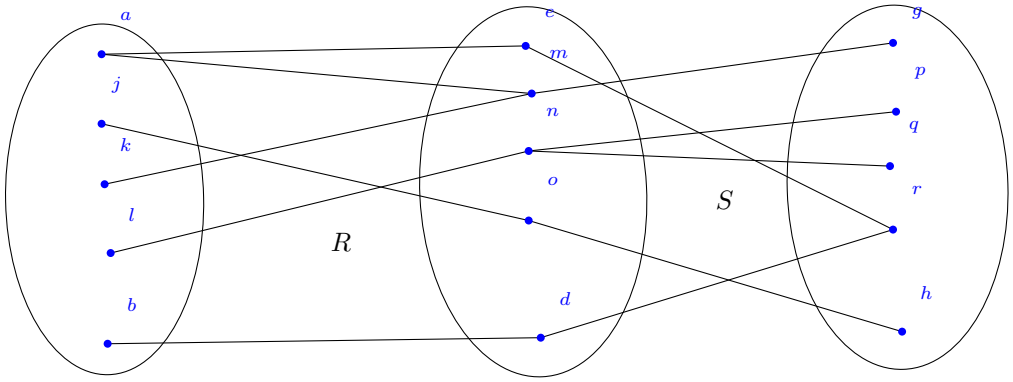
3.2.1 Relations

The **arity** of a function or a relation is the number of arguments it is supposed to have. It is a significant but generally unremarked fact that one can do most of mathematics without ever having to consider relations of arity greater than 2. Relations of arity two are **binary**.

We write $R(x, y)$ to mean that x and y are related by R. Sometimes we write it as $x\, R\, y$ instead. This is **infix** notation. Infix notation is universally used with order relations. We always write '$x \leq y$' rather than '$\leq (x, y)$', though there is no significance to this fact. I respect this tradition, but I tend not to use infix notation otherwise. One reason for this is that there is no way of writing ternary *etc* relations as infix!

Relations being sets, you can do anything to relations that you can do to sets: all the boolean operations: union, intersection, set difference and so on. But there are some extra operations you can do to relations which don't arise in this way. Composition and inverse; mere *sets* do not have composition and inverse, but *sets of ordered pairs* do. And just as there are special sets (the empty set, the universal set) there are also some special relations: one example is the identity relation. It's often written 'Δ_X' or '1_X' (where X is the intended domain) or even just '1'. For any set X there is also the universal relation on X, which is $X \times X$. There doesn't seem to be a standard notation for this construct. For my part i intend to stick to '1' or '1_X' when we need to be clear that it is X that we are restricting the identity relation to.

3.2.2 Composition and inverse



x is related to y by R-**composed-with-**S if there is a z such that x is related to z by R and z is related to y by S. In symbols, in infix notation

$$x\ R \circ S\ y \longleftrightarrow (\exists z)(xRz \wedge zSy)$$

So $R \circ S$ is the composition of R with S. Often written $R \cdot S$.

If we don't want to use infix we can write the biconditional as

$$R \circ S(x, y) \longleftrightarrow (\exists z)(R(x, z) \wedge S(z, y))$$

I will try to stick to '$\circ$' here because I want to go only using '$\cdot$' for multiplication of numbers. (Notice that we really do mean '$\exists$' not '$\forall$': for a woman to be your aunt it is sufficient for her to be the sibling of even *one* of your parents; she doesn't have to be a sibling of both of them!)

[Miniexercise: How can your aunt be a sibling of both your parents, without any laws being broken?]

Notice that I haven't mentioned ordered pairs here, and this definition of $R \circ S$ works for relations-in-intension just as well as for relations-in-extension. In fact it's probably more natural to think of composition as something one does to relations-in-intension.

Lots of examples here please

Exercise 26. *Write down a set abstract for the composition $R \circ S$ of two binary relations-in-extension R and S.*

If $R \subseteq A \times B$ and $A \cap B = \emptyset$, what is $R \circ R$?

You might wonder what the composition of two ternary relations is. Don't: we won't need it. But do bear in mind that $R \circ S$ is not in general the same as $S \circ R$: the sibling of your parent is probably not the parent of your sibling. Come to think of it, is it legally possible for them to be the same? There's a mini exercise!

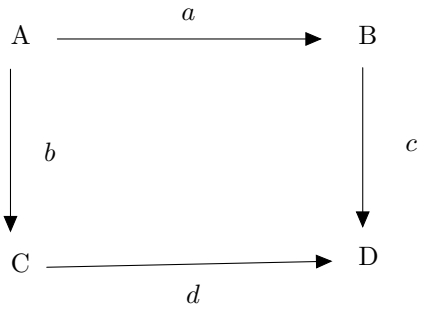
If $R \circ S = S \circ R$ we say that R and S **commute**. This word 'commute' is probably more usually used of functions:

If $(\forall x)(f(g(x)) = g(f(x)))$ we say f and g commute.

binary operations commute too!

(see the end of section 3.2.8.)

There is a style of picture which, although you will not explicitly need to know about it, you may find helpful. For example, when people say that the following diagram "commutes"



they mean that $c \circ a = d \circ b$.

$R \circ R$ is written R^2, and similarly R^{n+1} is $R \circ R^n$ for all natural numbers n. Remember that 1 is the identity relation, and $R \circ 1$ is just R, so one can think of 1 as R^0, which is pleasing. The **inverse** or **converse** of R, written 'R^{-1}', is $\{\langle x, y\rangle : \langle y, x\rangle \in R\}$. However, do not be misled by this exponential notation into thinking that $R \circ R^{-1}$ is the identity.

(What can you say about R if $R \circ R$ is the identity? On this last point see exercise 38 part (v). If you matched (v) up correctly by a process of elimination then you will have checked the correctness of your answer independently.)

Do not be confused by the difference between the converse R^{-1} of a relation R and its *complement*: $(X \times X) \setminus R$, the set of ordered pairs not in R. Just to check, have a quick look at the following

Exercise 27.
Is the complement of the converse of R the same as the converse of the complement?
Is the converse of the converse of R the same as R?
Is the complement of the complement of R the same as R?

3.2.3 Digraphs and Matrices for Relations-in-Extension

A digraph (short for "directed graph") is a set of vertices joined by lines that may have directions on them. So there might be an arrow from a to b and an arrow from b to a. Normally we do not allow there to be edges from a vertex back to itself, but in settings where we do allow such edges in digraphs, we call them "digraphs with loops".

Have a look at

`https://www.dpmms.cam.ac.uk/~tf/cam_only/discrete-trial-101-test.pdf`

It is sometimes convenient to think of a binary relation as a matrix whose entries are 1 and 0 (proxies for `true` and `false`). This has an advantage, namely that under this scheme the matrix product of the matrices for R and S is the matrix for $R \circ S$. (If you want the entries to be `true` and `false` (instead of their proxies of 1 and 0) you have to take multiplication to be $\wedge$ and addition to be $\vee$ (in your definition of matrix multiplication). More chat here

If this is the matrix for the relation $R \subseteq A \times B$

R	b_1	b_2	b_3
a_1	T	T	F
a_2	F	F	T
a_3	T	F	T

and this is the matrix for the relation $S \subseteq B \times C$

S	c_1	c_2	c_3
b_1	T	T	F
b_2	T	F	F
b_3	F	T	F

then we obtain the matrix for $R \circ S$ by matrix multiplication ...

You might like to fill in the question marks yourself ...

Exercise 28.

$R \circ S$	c_1	c_2	c_3
a_1	?	?	?
a_2	?	?	?
a_3	?	?	?

However, in principle it is not a good habit to think of binary relations as matrices in this way, because it forces one to decide on an ordering of the underlying set (after all, we have to decide on an order in which to write down the rows and columns) and this choice of an order makes this representation less general than the picture of binary relations-in-extension as sets of ordered pairs. (If you look at the table (3.2.5) of compatibility between blood groups, the suggestive distribution of crosses is suggestive only because the columns are written in the same order as the rows.)

Despite this, it can be useful at times, since it does give a nice picture of converses: the matrix of inverse/converse R^{-1} of R is the transpose of the matrix corresponding to R (so a symmetrical relation is one whose matrix is equal to its own transpose—and this is true however you order the elements of the domain) and this fact may help you break into this set of ideas.

This possibility of representing binary relations as matrices with values in $\{0, 1\}$ or in {`true`, `false`} serves also to make the point that you must always be prepared to re-think the data structures that you use in writing programs. Sometimes you might want to think of relations as matrices, sometimes as digraphs, sometimes as God-knows-what. And the same goes for other data structures too; this trick of trying to think of a object that is *prima facie* of one data type as actually belonging to another is essential if you are to exploit all your algorithms fully. If you can contort your problem into a problem about graphs then you can use a graph algorithm on it; if you can contort it into a problem about matrices then you can use a matrix algorithm on it.

Most of the applications of matrices belong in what one might loosely call 'Continuous Mathematics' rather than Discrete Mathematics (specifically in connection with Vector Spaces) and we do not cover Vector Spaces here.

3.2.4 Other properties of relations

A relation R is **transitive** if $\forall x \forall y \forall z\ R(x, y) \wedge R(y, z) \rightarrow R(x, z)$ (or, in brief, $R^2 \subseteq R$). A relation R is **symmetrical** if $\forall x \forall y (R(x, y) \longleftrightarrow R(y, x))$ or $R = R^{-1}$. Beginners often assume that symmetrical relations must be reflexive. They are wrong, as witness "rhymes with", "conflicts with", "can see the whites of the eyes of", "is married to", "is the sibling of" and so on.

A binary relation R is **extensional** if

$$(\forall x)(\forall y)(x = y \longleftrightarrow (\forall z)(R(x, z) \longleftrightarrow R(y, z)).$$

Notice that a relation can be extensional without its converse being extensional: think "square roots". An extensional relation on a set X corresponds to an injection from X into $\mathcal{P}(X)$, the power set of X. For us the most important example of an extensional relation will be $\in$, set membership. Two sets with the same members are the same set.

A binary relation on a set X is **reflexive** if it relates every member of X to itself. (A relation is **irreflexive** if it is disjoint from the identity relation: note that irreflexive is not the same as not-reflexive!) That is to say, R is reflexive iff[6] $(\forall x \in X)(\langle x, x\rangle \in R)$. Notice that this means that reflexivity is not a property of a relation, but of the structure $\langle X, R\rangle$ of which the relation is a component.

Exercise 29. *(*)*

Look up 'monophyletic'. Using only the auxilliary relation "is descended from" give a definition in first-order logic of what is is for a one-place predicate of lifeforms (reptile(x), whale(x) ...) to be monophyletic.

[6] I think this is the first place where i have used this standard abbreviation: 'iff' means if and only if'.

Exercise 30. *Show that, for all R, S and T,*

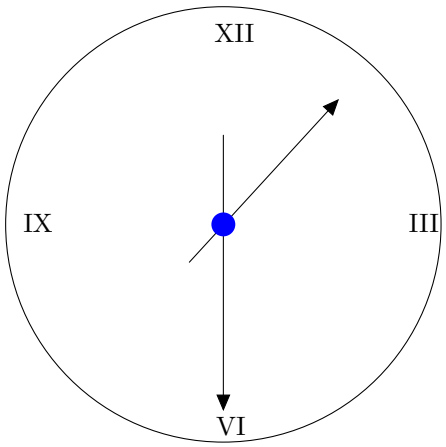
(i) $R \subseteq S \to R \circ T \subseteq S \circ T$;
(ii) $R \subseteq S \to T \circ R \subseteq T \circ S$;
(iii) $R \circ (S \cup T) = (R \circ S) \cup (R \circ T)$;
(iv) $(R \circ S)^{-1} = S^{-1} \circ R^{-1}$.

Which of the following are true?

(i) $R \subseteq S \to R^{-1} \subseteq S^{-1}$;
(ii) $R \subseteq S \to R^{-1} \supseteq S^{-1}$;
(iii) $R = R^{-1} \to 1 \subseteq R$.

Finally, it may be worth making the point that not all relations are binary relations. There is a natural three-place relation of *betweenness* that relates points on a line, but this doesn't concern us much. Of more interest (it lurks in the background in chapter 4 is the three-place relation of "later than" between hours on a clock. We cannot take this relation to be binary because if we do, it will simply turn out to be the universal relation. Every time on the clock is later than every other time if you wait long enough! However, with a three-place relation we can say "Starting at 12 o'clock we first reach 3 o'clock and then 6 o'clock" (which is true) and "Starting at 12 o'clock we first reach 6 o'clock and then 3 o'clock" (which isn't). Or we can think of it as "starting at x and reading clockwise we encounter y first and then z")

Exercise 31. *(*)*
Consider the clockface below. Write down the graph of the three-place order relation on the four positions on the face.

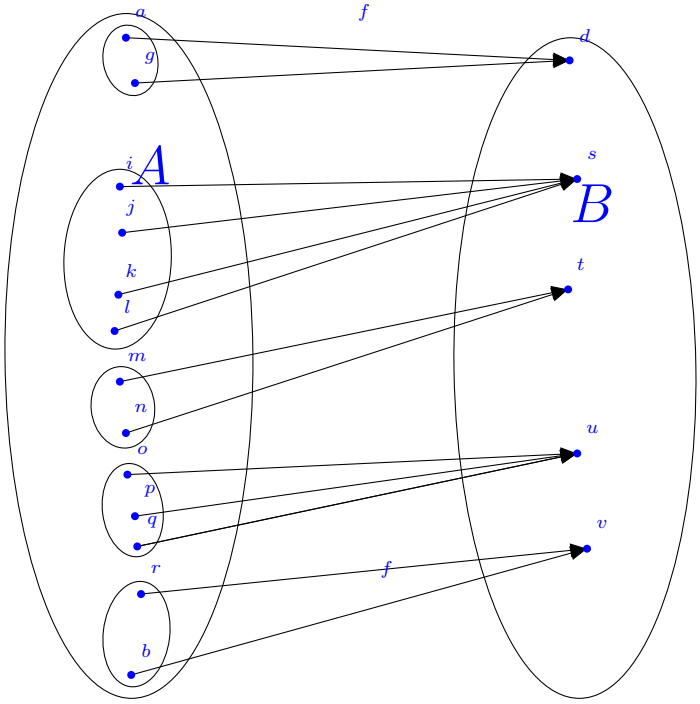


3.2.5 Equivalence relations

An **equivalence relation** is a relation that is symmetrical, transitive and reflexive. There is an important connection with the *partitions* that we met in definition 1 in section 3.1.7.

For example, the equivalence relation on the set $\{a, b, c, d, e\}$ that relates a, b and c to each other, and relates d and e to each other, corresponds to the partition $\{\{a, b, c\}, \{d, e\}\}$ of $\{a, b, c, d, e\}$. The pieces of this partition are called the **equivalence classes** of the equivalence relation. Similarly any partition of a set A gives us an equivalence relation on A. The relation that holds between two members of A when they belong to the same piece of the partition is an equivalence relation.

The following picture shows us some things we will want to define and describe.



Refer back to definition 1 on p 49

Here we have a picture of a surjection $f : A \twoheadrightarrow B$, where $B = \{d, s, t, u, v\}$ and $A = \{a, g, i, k, l, m, n, o, p, q, r, b\}$. The left-to-right arrows in the picture correspond (one-to-one) with the ordered pairs in [the graph of] f. The five things that look like parachutes falling left-to-right are examples of things often called **fibres**. Thus a fibre is a thing in the target (range, codomain, call it what you like) together with all the arrows that reach it, and the sources of the ar-

rows which can be found bundled into an ellipse on the left. Since the function going from left-to-right is f we say that these fibres are fibres **of** f. The set of (five) ellipses on the left is a partition of A, and each ellipse is a piece (of that partition). There is an equivalence relation lurking here, and that is the relation that holds between any two members of A that belong to the same piece (ellipse). It's perhaps a *wee* bit laborious to write out all the ordered pairs in this relation, co's there are quite a few of them. How many, exactly? We don't seem to have used the letter 'E' in this picture, so we can use it to denote this equivalence relation. (The letter 'E' is often used to denote an equivalence relation.) Finally we say that f is a **classifier** for E, by which we mean that $(\forall xy)(f(x) = f(y) \longleftrightarrow E(x, y))$.

Thus every surjection gives rise to an equivalence relation E, and it is a classifier for E. There are other classifiers for E, since $\pi \circ f$ is such a classifier whenever π is a permutation of $\{d, s, t, u, v\}$.

For example, the function that sends every set to its cardinality [notated in various ways] is a classifier for the relation of *being-in-bijection-with* aka *equipollence*. which i must discuss here

You are not going to be asked any time soon any questions in any exam that rely on your knowing the words 'fibre' or 'classifier' but in my experience the ability to use them properly helps get things clear in your mind.

Just as we often write '$\leq$' rather than 'R' or 'S' when we want to point to a partial order (and write it as an infix to boot) so we often write '$\sim$' rather than 'R' or 'S' when we want to point to an equivalence relation (and write that as an infix too)

It may be helpful here to introduce some notation:

If $\sim \ \subseteq \ X \times X$ (i.e., if $\sim$ is an equivalence relation on X) then ...

We write '$[x]_\sim$' for the equivalence class of x under $\sim$. In curly bracket notation this is '$\{x' : x' \sim x\}$'.

We write '$X/\sim$' for $\{[x]_\sim : x \in X\}$, the set of equivalence classes of members of X. It is sometimes called the *quotient* of X "over" $\sim$.

$X/\sim$ is a partition of X, a set of pairwise disjoint subsets of X that collectively exhaust X.

We have seen earlier how there is a correspondence between partitions of a set and equivalence relations on that set. If Π is a partition of a set X, then the binary relation that holds between members of X when they belong to thwe same piece of Π is an equivalence direction. For the other direction, if we are given an equivalence relation $\sim$ on X then we can obtain a partition Π of X ... how? Every x in X belongs to a piece of the partition (the pieces must cover the whole of X) so ... what else do we find in the piece that contains x? Everything that is related to x by $\sim$!

It has to be admitted that this takes some getting used to!

If an equivalence relation $\sim$ has n equivalence classes we say that it is "of index n"

Congruence relations

Definition 2. *An equivalence relation $\sim$ is a* **congruence relation for** *an n-ary function f if, whenever $x_i \sim y_i$ for $i \leq n$, then $f(x_1 \dots x_n) \sim f(y_1 \dots y_n)$.*

Congruence relations will crop up in other courses too but one particular example will be important to us here, and it's one that may be known to you already. The equivalence relation on natural numbers: "n and m have the same remainder on division by p" is a congruence relation for addition and for multiplication. We will deal with this in chapter 4.

If you know any chemistry you probably know the fact that the relation on isotopes of "having the same number of protons" is a congruence relation for all of chemistry. (Tho' you don't know that fact under that description!) This is why the name of a chemical element covers lots of different isotopes.

(Well, that's not entirely true. The difference between the two stable isotopes (protium and deuterium) of hydrogen are sufficient to give rise to genuine differences in chemistry. If you give up drinking H_2O and take to drinking D_2O instead it will eventually kill you. Wondering whether or not H and D are different elements is not a totally stupid question, but it's a question about what the identity criteria for chemical elements are to be, rather than a substantive question in the practice of chemistry.)

Blood Groups

Here is another real-life example of a congruence relation. Consider the relation between humans "It is safe for x to receive a transfusion of blood from y." Ignoring for the moment the fact that there are blood-borne diseases such as HIV, CJD, Hep C and so on, we find that if x can safely receive a transfusion of blood from y, and y' belongs to the same blood group as y, then x can safely receive a transfusion of blood from y'. That is to say, the equivalence relation of having-the-same-blood-group is a congruence relation for the binary relation "x can safely receive a transfusion of blood from y". In fact this is how blood groups were discovered.

That way we can think of the relation "x can safely receive a transfusion of blood from y" as really a relation between the blood groups, and summarise it in the following matrix.

Columns are donors, rows are recipients.

	$O-$	$O+$	$B-$	$B+$	$A-$	$A+$	$AB-$	$AB+$
$O-$	X							
$O+$	X	X						
$B-$	X		X					
$B+$	X	X	X	X				
$A-$	X				X			
$A+$	X	X			X	X		
$AB-$	X		X		X		X	
$AB+$	X	X	X	X	X	X	X	X

The blood groups themselves (O-, O+, B-, B+, A-, A+, AB- and AB+) are the equivalence classes under this equivalence relation.

Need some exercises on equivalence relations

You may be struck by the pleasing pattern made by the Xs: I certainly was. It looks a bit like a thing they call a "Sierpiński triangle" (no: don't google it!) What I want you to do is follow the same path I trod when I spotted this pattern, namely:

I) Does this pattern tell you anything significant about the properties of this compatibility relation? I'm thinking of reflexivity, transitivity, symmetry, antisymmetry etc. etc. When you've got that sorted out, procede to part II.

II) We have already seen other ways of representing binary relations. Try some of them on this data and see if you get any pretty pictures. I tried it and I got a three-dimensional shape. I want you to find that three-dimensional shape.

III) One moral I want to draw from this coursework is that Discrete Mathematics and mere thought can be useful in developing profitable hypotheses in the sciences. You probably know enough biology to know that characters are inherited by genes, and that we each have two copies of each gene. Each gene can come in several forms called *alleles*. Some alleles are *dominant* in that you express them even if you have only one copy (brown eyes); some are *recessive* in that you express them only if you have two copies (blue eyes, green eyes). The picture you have developed will suggest to you a hypothesis about how many genes there are that control your blood group, and how many alleles there are at each gene, and which are dominant. Formulate this hypothesis.

Should cut this back a lot

Do not forget that this is a discrete mathematics question, not a biology question. There is nothing to be gained—and time to be lost—in surfing the web for information about blood groups.

Any one subject matter may admit lots of congruence relations, not just one. The arithmetic of the integers invites us to look at the congruence relation of $+$ and $\times$ of having-the-same-remainder-mod-p, but the identity relation is a congruence relation for these operations (for *all* operations, indeed).

3.2.6 Partial orders

This section is a jumble of pieces all stuck together and needs to be sorted out

Order relations obviously have to be transitive, and they cannot be symmetrical because then they would not distinguish things, would they? Indeed transitive relations that are symmetrical are called *equivalence* relations (as long as they

are reflexive). So how do we capture this failure of symmetry? We start by noticing that, although an order relation must of course be transitive and cannot be symmetrical, it is not obvious whether we want it to be reflexive or want it to be irreflexive. Since orderings represent ways of *distinguishing* things, they do not have anything natural to say about whether things are related to themselves or not. Is x less than itself? Or not? Does it matter which way we jump? Reflection on your experience with $<$ and $\leq$ on the various kinds of numbers you've dealt with (naturals, integers, reals and rationals) will make you feel that it does not much matter. After all, in some sense $<$ and $\leq$ contain the same information about numbers (See exercise 39.41). These two ways give rise to two definitions.

Get this reference right

1. A **strict partial order** is irreflexive, transitive and asymmetrical. (A relation is **asymmetrical** if it cannot simultaneously relate x to y and y to x. This of course implies irreflexivity.)

2. A **partial order** is reflexive, transitive and ... well it cannot be asymmetrical because $x \leq x$. We need to weaken asymmetry to a condition that says that, if $x \neq y$, then not both $x \leq y$ and $y \leq x$. This condition, usually expressed as its contrapositive $(\forall xy)(x \leq y \wedge y \leq x \rightarrow x = y)$, is **antisymmetry** and is the third clause in the definition of partial order.

We need to define contrapositive

A relation that is reflexive and transitive (antisymmetry not guaranteed) is a **preorder** or **quasiorder**. For example the relation between humans "x can safely receive blood from donor y" is a preorder. The intersection of a preorder with its converse is always an equivalence relation. In this case the equivalence relation is the relation of having-the-same-blood-group.

You absolutely must have these definitions at your fingertips, co's these things crop up all the time. Merely knowing where to look them up isn't enough.

Exercise 32. *Are either of the following true?*

1. *The identity relation is a partial order.*

2. *The empty relation is a strict partial order.*

If R is a partial ordering of a set X, then $R \setminus \{\langle x, x\rangle : x \in X\}$ is a strict partial ordering of X, and if R is a strict partial ordering of a set X, then $R \cup \{\langle x, x\rangle : x \in X\}$ is a partial ordering of X. You obtain each from the other by unioning with the identity relation or substracting the identity relation. Thus each concept (partial order and strict partial order) can be defined in terms of the other. There is a scrap of logical slang that comes in handy here: we say that each can be defined if we take the other as **primitive**. A primitive is a concept in terms of which you define other concepts.

This interdefinability of partial orders with the corresponding strict partial orders is something that you are probably familiar with in an informal way. If

you want to talk about natural numbers it doesn't much matter whether you use $\leq_{\mathbb{N}}$ or $<_{\mathbb{N}}$ because of the equivalences

$$x \leq_{\mathbb{N}} y \iff x <_{\mathbb{N}} y \ \vee \ x = y$$

and

$$x <_{\mathbb{N}} y \iff x \leq_{\mathbb{N}} y \ \wedge \ x \neq y.$$

This line of thought can help clarify where the definition of antisymmetry comes from. If you think of $<_{\mathbb{N}}$ then it is clear that a strict partial order must be *asymmetrical*: $(\forall x, y)(\neg(x < y \wedge y < x))$, transitive: $(\forall xyz)(x < y \wedge y < z. \rightarrow x < z)$ and *irreflexive*: $(\forall x)(\neg(x < x))$ [tho' irreflexivity actually follows from asymmetry]. The way to understand the definition of *antisymmetrical* is to ask yourself: "If i have an *asymmetrical* relation, and i take the union of it with the identity relation, what condition-like-asymmetry does the new relation satisfy?" It can't be straight-up asymmetry beco's x is always related to x (co's we took the union with the identity relation) but it will say that x cannot be related to y at the same time as y is related to x *unless* $x = y$. And this is antisymmetry.

Total orders are a special kind of partial order. (Do not overinterpret and assume that partial orders cannot be total!) Again, they come in two flavours:

1. A **strict total** order is a strict partial order that satisfies the extra condition $(\forall xy)(x < y \vee y < x \vee x = y)$. Because this condition says there are no more than three possibilities, it is called **trichotomy** (from two Greek words meaning *three* and to *cut* as in a-*tom*, lobo-*tomy*.)

2. A **total order** is a partial order with the extra condition $(\forall xy)(x \leq y \vee y \leq x)$. This property is called **connexity**, and relations bearing it are said to be **connected**. ("connected" also has a meaning in graph theory, so beware)

Thus trichotomy and connexity are related to each other the way antisymmetry and asymmetry are.

> A **poset** $\langle X, \leq_X \rangle$ is a set X with a partial ordering $\leq_X$.
>
> A **monotone** function from a poset $\langle A, \leq_A \rangle$ to a poset $\langle B, \leq_B \rangle$ is a function $f : A \to B$ such that $\forall xy(x \leq_A y \to f(x) \leq_B f(y))$.

The word 'monotone' in mathematics refers to functions f which satisfy conditions like

$$x \leq y \to f(x) \leq f(y).$$

We say such a function is *monotone increasing* with respect to $\leq$. (If instead f satisfies $x \leq y \to f(x) \geq f(y)$ we say f is monotone *de*creasing with respect to $\leq$.) Of course, it may be (as it is in fact the case here) that the partial order in the antecedent of the condition is not the same partial order as in the consequent, so ideally we would need a more complex form of words along the lines of "f is monotone [increasing] with respect to $\leq$ and $\leq'$". However this

ideal notation is never used, being sacrificed by ellipses to the form of words "f is monotone [increasing]".

We use it here because the function F that takes a set of assumptions A and returns the set $F(A)$ of its logical consequences is monotone with respect to set-inclusion :

$$A \subseteq B \rightarrow F(A) \subseteq F(B).$$

The definition of a partial ordering as a relation that is transitive, reflexive and antisymmetrical applies equally well to relations-in-intension and relations in extension. A partial-order-in-extension is a set of ordered pairs. Generally relations-in-extension are sets of ordered tuples. But certain kinds of relations have other representations as extensional objects. We have seen that binary relations can be pictured as digraphs-with-loops as well as sets of ordered pairs. But there is more. A total ordering of a (finite) set can be represented as a list (without repetitions) of all the members of the set. Notice that this is a more economical representation of a total order than its representation as a set of ordered pairs: the representation of a total ordering of a finite set X is of length $|X|$ whereas its representation as a set of ordered pairs has... You tell me!

Exercise 33. *How many ordered pairs are there in a total ordering of a set with n elements?*

This trick doesn't work for an arbitrary partial order that isn't total. We can code a partial ordering as the set of its initial or terminal segments—as a set of sets. Think of the knot of people round an airplane lavatory on a long-haul flight at dawn. Each person knows only the set of people who were there before they were. (And it is a *set* that each knows not a *list*, co's they don't know what order the people ahead of them arrived). This way of representing a total order—as a nest of subsets—is sometimes called an *ordernesting*. (In fact, if you think of it, you will see that the usual Wiener-Kuratowski pair[7] is an ordernesting!). This serves to point the useful moral that many objects of interest to us can be coded or represented in more than one way. (This point was made also in connection with the matrix representation of binary relations in section 3.2.3.)

Transitive closure

For any binary relation-in-extension R whatever, the relation $R \cup 1$ (remember 1 is the identity relation) is a reflexive relation. By now you will have noticed also that for any binary relation R whatever, the relation $R \cup R^{-1}$ is a symmetrical relation. $R \cup 1$ is the **reflexive closure** of R, and is sometime written $r(R)$ to commemorate this fact. Similarly $R \cup R^{-1}$ is the **symmetric closure** of R and written '$s(R)$' similarly. The work being done by the word 'closure' here is not psychobabble: you should use it to remind yourself that what you are doing in these two cases is adding to R precisely the ordered pairs needed to make

[7] which you may not have met yet...

it reflexive, or symmetric. (Of course you can add more still: the idea here is to add the minimum necessary). An important feature of this idea is that this process is *deterministic*: there is a unique minimal way to add ordered pairs to R to obtain a reflexive relation, or a symmetric relation. In contrast if you want to add ordered pairs to a partial order to obtain a total order there is no obvious right way to do it. If the partial order is indifferent between Tweedledum and Tweedledee there is nothing about the partial order to tell you what to do. Whereas if a relation that isn't transitive relates a to b and b to c then we know to relate a to c in the transitive closure.

To be formal about it

Definition 3.

- *The reflexive closure of a binary relation-in-extension R is the least (with respect to $\subseteq$) set of ordered pairs that is both a reflexive relation and a superset of R:*

$$r(R) := \bigcap \{S : R \subseteq S \wedge 1 \subseteq S\}$$

and

- *The symmetric closure of a binary relation-in-extension R is the least (with respect to $\subseteq$) set of ordered pairs that is both a symmetric relation and a superset of R:*

$$s(R) := \bigcap \{S : R \subseteq S \wedge S = S^{-1}\}$$

and finally

- *The transitive closure of a binary relation-in-extension R is the least (with respect to $\subseteq$) set of ordered pairs that is both a transitive relation and a superset of R:*

$$t(R) := \bigcap \{S : R \subseteq S \wedge S^2 \subseteq S\}$$

It occurs to me that it just might be a good idea to check at this juncture that the intersection of two transitive relations-(in-extension) is another transitive relation. Here goes:

If R and S are transitive relations, so is $R \cap S$.

I take it we are all agreed that if $X \subseteq Y$ and $X' \subseteq Y'$ then $X \circ X' \subseteq Y \circ Y'$. Applying this to $R \cap S \subseteq R$ and $R \cap S \subseteq S$ gives us the two inclusions

$$(R \cap S) \circ (R \cap S) \subseteq R \circ R \subseteq R$$
$$(R \cap S) \circ (R \cap S) \subseteq S \circ S \subseteq S$$

whence

$$(R \cap S) \circ (R \cap S) \ \subseteq \ R \cap S$$

as desired.

■

Notice that the same argument shows that the intersection of any number of transitive relations is a transitive relation: i.e., transitivity is an intersection-closed property of relations.

Relational algebra in this style goes back to Russell and Whitehead (1919).

Symmetric and reflexive closures of relations one can build in one hit, as above, since they are $R \cup 1$ and $R \cup R^{-1}$ respectively. Transitive closures are a bit more of a mouthful, which is why we left them until last.

We will show that $t(R)$ is in fact $\bigcup_{n \in \mathbb{N}} R^n$; something slightly easier to understand.

To do this it will be sufficient to show

1. $\bigcup_{n \in \mathbb{N}} R^n$ is transitive;
2. If S is a transitive relation $\supset R$ then $\bigcup_{n \in \mathbb{N}} R^n \subseteq S$.

For (1) we need to show that if $\langle x, y \rangle$ and $\langle y, z \rangle$ are both in $\bigcup_{n \in \mathbb{N}} R^n$ then $\langle x, z \rangle \in \bigcup_{n \in \mathbb{N}} R^n$. If $\langle x, y \rangle \in \bigcup_{n \in \mathbb{N}} R^n$ then $\langle x, y \rangle \in R^k$ for some k, and if $\langle y, z \rangle \in \bigcup_{n \in \mathbb{N}} R^n$ then $\langle y, z \rangle \in R^j$ for some j. Then $\langle x, z \rangle \in R^{j+k} \subseteq \bigcup_{n \in \mathbb{N}} R^n$.

For (2) let $S \supset R$ be a transitive relation. So $R \subseteq S$. We prove by induction on $\mathbb{N}$ that for all $n \in \mathbb{N}$, $R^n \subseteq S$. Suppose $R^n \subseteq S$. Then

$$R^{n+1} = \ R^n \circ R \ \subseteq^{(a)} \ S \circ R \ \subseteq^{(b)} \ S \circ S \ \subseteq^{(c)} \ S.$$

Inclusions (a) and (b) hold because $\circ$ is *monotone*: if $X \subseteq Y$ then $X \circ Z \subseteq Y \circ Z$. Inclusion (c) holds because S is transitive. (See the first two parts of exercise 9.)

A picture that might help

Several levels:

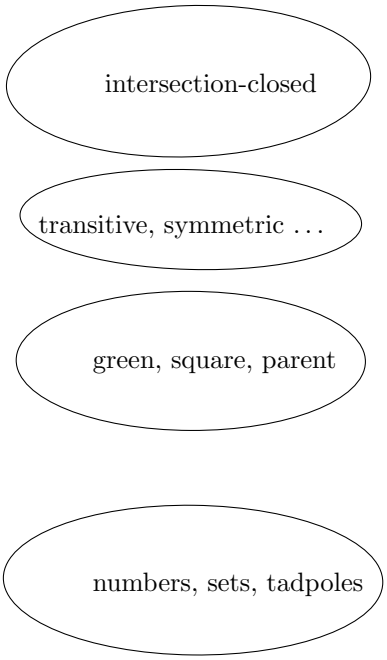
At the bottom level we have objects: *things that have properties*. These might be numbers, sets, people, tadpoles, booleans ...

Then, sitting above them, we have the properties that those things might have, and the relations that might hold between them. Being even, being-divisible-by, being green, being a parent-of, and so on.

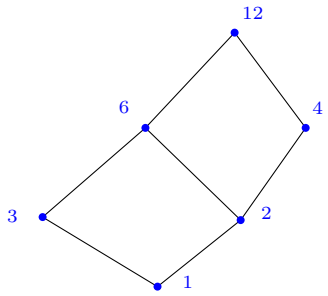
Above them are the properties those relations (and properties) and operations have: being transitive, symmetrical, reflexive .... And operations on those relations: converse, transitive closure etc.

Above them are properties of properties of properties of things: intersection-closed ... $F()$ is an intersection-closed property of relations if an intersection of any number of relations that have property F also has property F. Transitivity is an intersection-closed property of a relation, because an arbitary intersection of transitive relations is a transitive relation.

display properly the definitioin of intersection-closed



Hasse diagrams



The poset of the factors of 12 under divisibility - a Hasse diagram

The digraph picture gives rise to **Hasse diagrams**.

He's German, pronounced 'Husser' (unless you are from North of the river Trent!) And it's definitely two syllables not one. German 'Hass' with one syllable means *hatred*.

When drawing a digraph of a transitive relation R [at least on a finite set—let's not worry about Hasse diagrams for infinite relations!] one can safely leave out a lot of arrows and still display the same information: all one has to draw is the arrows for a relation whose transitive closure is R. One could restore all the missing arrows (should one wish to) because transitivity tells one where to put them in. Thus the relation represented by a lot of dots joined by arrows is the relation "I can get from x to y by following the arrows".

To be *slightly* more formal about it one can say that one obtains the Hasse diagram for a transitive relation R by first drawing a digraph with one directed edge for each pair in R. One then leaves out the loops at each vertex and—further—whenever one has both an edge from a to b and an edge from b to c one can delete the edge from a to c. The result is a digraph picture of a relation whose transitive closure in R. Thus it comes to pass that if R is a partial order on a finite set then there is a *minimal* relation whose transitive closure is R. For each x in the domain of R put an edge from x to each of its immediate successors—but put in no other edges! (This will even work for $\mathbb{N}$... one can draw a Hasse diagram for $\mathbb{N}$—yes i know i said we wouldn't consider infinite relations!—but not for $\mathbb{Q}$.)

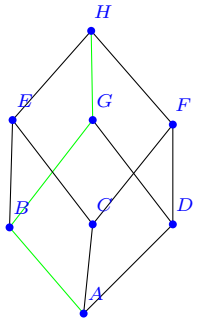
In fact we can even leave out the heads on the arrows (so we draw in edges rather than arrows) by adopting the convention that the end of the edge on which the arrowhead belongs is the end that is further up the page. The result of doing this is the Hasse diagram of that transitive relation. [one effect of this is that no Hasse diagram ever has a horizontal line anywhere!] The appeal of Hasse diagrams relies on and to some extent reinforces an unspoken (and false!) assumption that every partial order can be embedded somehow in the plane. Related to this is the weaker (but nevertheless still nontrivial) assumption that all total orders can be embedded in the real line, as instance, the image of Justice, blindfolded with a pair of weighing scales.

Although this is clearly a false assumption that might perhaps push our intuitions in wrong directions, it is not such a crazy idea in computer science, where linearity of time and of machine addresses compel us to assume that all partial orders can be refined to total orders. Any representation of a set in the bowels of a computer must always be as a list!

Chains

"carrier set"

The **restriction** of a relation R to a carrier set X (which is $R \cap X^n$, where n is the arity of R) is denoted by '$R \restriction X$'. (We saw restrictions of *functions* on page 42; restrictions of relations is a natural generalisation.) A **chain** in a poset $\langle X, \leq_X \rangle$, is a total ordering $\langle X', \leq_X \restriction X' \rangle$, where $X' \subseteq X$. In words: a chain in a poset is a subset totally ordered by the restriction of the order relation. An **antichain** in a poset is a subset of the carrier set such that the restriction of the order relation to it is the identity relation.



The subset $\{A, B, G, H\}$ (in green, equipped with the obvious restriction of the partial order) is a chain, as is $\{A, C, E, H\}$ similarly equipped. $\{B, C, D\}$ is an antichain.

Dilworth's theorem?

3.2.7 Products of orders

There is a general notion of product of structures, and you may well need to learn it eventually. However for the moment we will restrict ourselves to the case of most immediate interest: the products of two partial orderings.

ordered pair notation for structures

If $\langle X, \leq_X \rangle$ and $\langle Y, \leq_Y \rangle$ are two partial orders, then we can define partial orders on $X \times Y$ in several ways. The product defined above is called

Ollie Black tells me i should say more about pointwise product here

the **pointwise** product. In the **lexicographic** order of the product we set $\langle x, y\rangle \leq_{lex} \langle x', y'\rangle$ if $x <_X x'$ or $x = x'$ and $y \leq_Y y'$. Although straightforward examples of lexicographic products are scarce, there are a number of combinatorial devices that have the flavour of a lexicographic product. The intuition behind lexicographic product is that you are trying to order things by looking at the values they take under certain parameters, and that some parameters are more important than others for this purpose. The expression 'tie-breaker' can be helpfully evocative here. You are trying to choose between two things on the basis of the values they get with respect to some parameter. If two things get the same value, what do you do? You look at their value under some other (less important!) parameter. For example: the Olympic league table: one grades nations in the first instance by the number of gold medals their athletes have won, then by the number of silvers and only if these fail to discriminate between them does one count the number of bronzes. To be a bit more formal about it: we associate to each country an ordered triple $\langle g, s, b\rangle$ and we then order these triples lexicographically. We then copy the ordering on the triples back to the countries.

Other examples include the devices used to determine which team goes forward from a qualifying group in world cup football. *Prima facie* this should be the team with the largest number of points, but if two teams have the same number of points, one looks at the number of goals the two teams have scored, and so on, examining the values the two teams take under a sequence of parameters of dwindling importance until we find one with respect to which they differ. In cricket the analysis of a bowler who takes x wickets while conceding y runs is preferred to that of a bowler who takes x' wickets while conceding y' runs as long as $x > x'$ or $x = x' \wedge y < y'$. However, in none of these naturally occurring cases is one ordering *tuples* of things: rather, one is trying to order things by combining in various ways various preorders of the things. However, the underlying intuition is the same.

carrier set?

Notice that the lexicographic product is a superset of the pointwise product. If we have two partial orders with the same carrier set and (the graph of, or extension of) one is a superset of (the graph of, or extension of) the other, we say the first **extends** or **refines** the second.

A *total* or *linear* order is one that has no proper refinement (Can't add any ordered pairs to obtain a partial ordering of the same domain).

The **colex** ordering of $X \times Y$ orders pairs according to *last* difference. The colex ordering too is a superset (extension, refinement) of the pointwise product ordering.

Exercise 34. *Check that the pointwise product ordering is the intersection of the lexicographic ordering and the colex ordering.*

One naturally tends to think of preorders as preference orders, as the preorders in the illustrations above of course are. Although naturally not all preorders are preference orders, thinking of them as preference orders enables us to motivate the distinction between the pointwise product of $\mathcal{P} \times \mathcal{Q}$ of two preference orderings $\mathcal{P}$ and $\mathcal{Q}$ (which corresponds to impartiality between parameters

$\mathcal{P}$ and $\mathcal{Q}$) and the lexicographic product (according to which any increase in $\mathcal{P}$ is more important than any increase in $\mathcal{Q}$). Naturally occurring preference orderings on products of posets tend to be complicated. Lexicographic products are extremely unlikely to represent your views on baskets of apples and oranges because even if you prefer apples to oranges, you would be unlikely to prefer any increase (however small) in the number of apples you are offered to any increase (however large) in the number of oranges—unless, that is, you had no use for oranges in the first place. And in those circumstances you would hardly bother to express a preference for an-apple-and-*two*-oranges over an-apple-and-*one*-orange. (You would probably describe your tastes by giving apples more *utility*. Interesting stuff no doubt, but no concern of us in first-year Discrete Mathematics.)

On the other hand, your preference ordering is likely nevertheless to be finer than the pointwise product ordering: according to the pointwise product ordering, you would be unable to decide between a single orange-with-a-pound-of-apples and two-oranges-with-one-apple. You would have to be very blasé indeed not to prefer the first. After all, to a certain extent apples and oranges are interchangeable: realistic product (preference) orders refine the product order but are typically not as refined as a lexicographic order. (We must not get too deeply into utility theory!) Note merely that it is a sensible motivation for the study of orderings and products of orderings.

But before leaving preference orderings altogether the reader should notice at least that preference orders have a rather odd feature not shared by partial orders in general. $A \not\leq B \not\leq A$ and $B > C$ does not imply $A > C$, though one expects it to if the ordering is a preference ordering. This makes a nice exercise.

Exercise 35. *Are the two following conditions on partial orders equivalent?*

$$(\forall xyz)(z < x \not\leq y \not\leq x \to z < y),$$
$$(\forall xyz)(z > x \not\leq y \not\leq x \to z > y).$$

(This exercise uses four common conventions that it takes a logician to spell out.

(i) When '$\leq$' and '$<$' appear in the same formula they denote a partial ordering and its strict part, respectively;

(ii) The relations $\leq$ and $\geq$ are converses of each other;

(iii) that '$x < y < z$' is short for '$(x < y) \wedge (y < z)$';

(iv) putting a slash through a symbol (as in "$\not\leq$") negates it.)

Exercise 36. *Define $x\,R\,y$ on natural numbers by $x\,R\,y$ iff $x \leq y + 1$.*
What are the following relations?[8]

(1) $R \cap R^{-1}$;
(2) $R \setminus R^{-1}$;
(3) The transitive closure of the relation in (1);
(4) The transitive closure of the relation in (2).

[8]The structure $\langle \mathbb{N}, R\rangle$ is known to students of modal logic as the *Recession Frame*.

Pareto

Given a subset $X \subseteq (P \times Q)$, the points in X that are maximal in the pointwise product $\mathcal{P} \times_{pw} \mathcal{Q}$ are called "**Pareto-efficient** points" (of X) by economists. They are sometimes called "Pareto-*optimal*" because if X is the set of points that are in some sense accessible, or possible, or something of that nature, then a Pareto-efficient point in X is one that, once one has reached it, one cannot find another point in X that makes one of the coordinates better without simultaneously making another one worse. Pareto was an Italian economist. Natural illustrations are defective in the way that we have seen that natural illustrations of lexicographic products are defective, but they might still help. Here are some.

(i) The critical point of a substance is that temperature and pressure at which the difference between liquid and gas disappears. When the substance is at a higher temperature and higher pressure than this it is said to be **supercritical**. Methane is the compound most easily put into a supercritical state: all other compounds require either a more extreme temperature or a more extreme pressure or both. Methane is a Pareto-efficient point.

(ii) Each of the isotopes in the table below is a Pareto-efficient point if you plot atomic weight against half-life. For each of these isotopes it is the case that every heavier isotope has shorter half-life, and everything with a longer half-life has lower atomic weight.

Isotope	Half-life (years)
Pb^{208}	∞;
Bi^{209}	4×10^{19};
Th^{232}	1.405×10^{10};
U^{238}	4.468×10^{9};
Pu^{244}	8.08×10^{7};
Cm^{247}	1.56×10^{7}.

Pareto-Optimal Isotopes

(Why is U^{235} not in this list?)

Exercise 37. *Go to*
http://nucleardata.nuclear.lu.se/nucleardata/toi/perchart.htm to see if you can find a Pareto-optimal isotope with greater atomic weight than Cm^{247}.

(iii) No portrait survives of the mathematician Green—after whom Green Street in Cambridge is named, and who invented Green functions. There are more famous people than Green of whom no portrait survives, but they are all of them more recent. There are more recent people than Green of whom no portrait survives, but they are all of them less famous than he is.

(iv) Robert Browning is buried in Westminster Abbey. This prompted Henry James to observe that "A good many oddities and a good many great writers

have been entombed in the Abbey; but none of the odd ones have been so great and none of the great ones so odd."

(v) The airport at Christchurch in New Zealand is Pareto-efficient with respect to size and southerlyness. There are airports to the south of it, but they are all smaller. There are larger airports, but they are all north of Christchurch.

However, we will not develop these ideas here, as they find their most natural expression in connection with convex optimisation rather than logic. We touch briefly on convex optimisation in chapter 7.

Once one has explained Pareto-efficiency one can make the point that if the values taken by your parameters are infinitely divisible then you can have infinitely many pareto-efficient points. If they are discrete (like $\mathbb{N}$) then you can't. A sleeper for WQO theory. Don't ask. No, really.

Exercise 38. *Match up the properties in the left column with those in the right.*

(i) $R^2 \subseteq R$;	*(a)* R *is symmetrical*
(ii) $R \cap R^{-1} = \emptyset$;	*(b)* R *is antisymmetric*
(iii) $R \cap R^{-1} = 1$;	*(c)* R *is asymmetrical*
(iv) $R = R^{-1}$;	*(d)* R *is a permutation*
(v) $R \circ R^{-1} = 1$;	*(e)* R *is connected*
(vi) $R \cup R^{-1} = U$;	*(f)* R *is transitive*

3.2.8 Functions

(One should start a section on functions by defining injective and surjective, but i'm guessing you know those expressions already. You can always ask Wikipædia, which knows everything. I might supply some definitions later, if only for the sake of completeness)

The annoying feature of reflexivity we saw on page 54—that you cannot tell by looking at the graph of a relation whether it is reflexive or not, because you need to know the intended domain—(which irreflexivity does not share) is also exhibited by **surjectivity**, which is a property not of a function but a function-with-a-range. A function is surjective if every element of the range is a value. **Totality** likewise is a property of a function-and-an-intended-domain. A function f on a set X is total if it is defined for every argument in X. Normally we will assume that our functions are total unless the possibility of their being partial is explicitly flagged. (This is not so in all CS cultures. For example in the theory of computable functions it is always assumed—unless the word 'total' is there in black and white—that our functions need not be total.) rewrite this para

Some mathematical cultures make this explicit, saying that a function is an ordered triple of domain, range and a set of ordered pairs. This notation has the advantage of clarity, but it has not yet won the day.

In contrast, injectivity of a function-in-extension is a property solely of the function-in-extension and not of the intended domain or range. A function is injective iff it never sends distinct arguments to the same value. You can tell whether or not a function-in-extension is *injective* simply by examining the ordered pairs within it. No such examination can ever tell you whether or not it is *surjective*: you have to know additionally what the intended range of the function is.

Functions of more than one variable are usually written in the style '$f(x_1 \dots x_n)$' but some functions (such as $+$ and $\times$) traditionally are written in the infix style that we saw earlier (page 50).

The properties of *associativity*, *commutativity* and *distributivity* that I am about to explain seem always to be stated for functions that are written in infix notation—like $+$ and $\times$ on numbers (of all kinds). You know the equations

$$(\forall x)(\forall y)(x + y = y + x)$$
$$(\forall x)(\forall y)(x \times y = y \times x).$$

Those said that multiplication and addition are **commutative**. These two:

$$(\forall x)(\forall y)(\forall z)(x + (y + z) = (x + y) + z)$$
$$(\forall x)(\forall y)(\forall z)(x \times (y \times z) = (x \times y) \times z)$$

say that multiplication and addition are **associative**. Finally

$$(\forall x)(\forall y)(\forall z)(x \times (y + z) = (x \times y) + (x \times z)))$$

says that multiplication **distributes over** addition. Observe that $\wedge$ distributes over $\vee$.

An operation $*$ is said to be **idempotent** if

$$(\forall x)(x * x = x)$$

$\wedge$ and $\vee$ are idempotent operations on propositions. $+$ and $\times$ are not idempotent on numbers (of any kind). HCF is idempotent, though we will not make much use of this fact.

Be aware that there is another use of this word 'idempotent'. We also say that a *function* f is idempotent if $f(f(x)) = f(x)$ for all x. Constant functions are the most straightforward example of idempotent functions. Other important examples are the operations of *transitive closure*, of *symmetric closure* and *reflexive closure* all of which are idempotent, and which we saw in section 3.2.6. The word 'closure' is the clue here. In all these cases you are doing something (adding ordered pairs as it happens) until some condition is satisfied. And once it's satisfied ... well, it's satisfied.

We now need the concept of a **unit** for a binary operation. Notice that—for example, the following hold for natural numbers, reals etc:

$$(\forall x)(x \cdot 1 = x) \text{ and } (\forall x)(x + 0 = x).$$

We express this by saying that 1 is a unit for multiplication (a "multiplicative unit") and 0 is a unit for addition (an "additive unit"). In general a constant `c` is a unit for a commutative operation $*$ if $(\forall x)(x * \texttt{c} = x)$. For example $\emptyset$, the empty set, is a unit for $\cup$: $(\forall x)(x \cup \emptyset = x)$, and V, the universal set, is a unit for $\cap$. The propositional constants `true` and `false` are units for $\wedge$ and $\vee$ respectively.

Along with units come inverses—sometimes! 0 is a unit for $+$, and $-x$ is the additive inverse of x, in the sense that $x + (-x) = 0$. Similarly $1/x$ is the multiplicative inverse of x (unless $x = 0$!). In general we say that $f(x)$ is the inverse of x from the point of view of $*$ if $(\forall x)(x * f(x) = \texttt{c})$. (Remember that `c` was the unit for $*$.) Sometimes but not always. Notice that $\cup$ and $\cap$ do not have inverse functions: there is no function f such that $(\forall x)(x \cup f(x) = \emptyset)$. And $\wedge$ and $\vee$ do not have inverses either.

If the binary relation on `widgets` has nice properties it can be extended to a binary relations on datatypes constructed out of widgets:

- A binary operation on widgets that is associative can be naturally extended to a function `widget-list -> widget`.
- If the function is also commutative then it can be naturally extended to a function from *multisets*-of-`widgets` to `widgets`.
- If it is (associative and) commutative and also idempotent then it can be naturally extended to a function from *sets*-of-`widgets` to `widgets`, as in the illustrations below.

For example both addition and multiplication of numbers are associative and commutative (but not idempotent) so

- You can add two numbers together to get a number, so you can add together all the numbers in a (multi)-set X of numbers to get a number.
- You can multiply together all the numbers in a multiset of numbers. (We write the sum of all numbers in X as 'ΣX' and the product as 'ΠX'.)
- The operations $\cap$ and $\cup$ are associative, commutative and idempotent, so you can form $\bigcap X$ and $\bigcup X$ if X is a set of sets.
- The operations $\wedge$ and $\vee$ are associative commutative and idempotent, so you can form the compound propositions $\bigwedge P$ and $\bigvee P$ when P is a set of propositions.

This raises an obvious question, or family of questions.

What happens if we apply Σ *or* Π *to the empty set of numbers?*
What happens if we apply $\bigvee$ *or* $\bigwedge$ *to the empty set of propositions?*
What happens if we apply $\bigcap$ *or* $\bigcup$ *to the empty set of sets?*

It's not hard to see that, for all these questions, the answer must be *the unit for the operation in question*: $\bigvee \emptyset = \texttt{false}$, $\bigwedge \emptyset = \texttt{true}$, $\Sigma\emptyset = 0$ and $\Pi\emptyset = 1$, $\bigcup \emptyset = \emptyset$ and $\bigcap \emptyset = V$. The correctness of these last two equalities can be checked by literal-mindedly unravelling the set abstracts.

$\Sigma(X \cup \{y\})$ is obviously $\Sigma(X) + y$. So $\Sigma(\emptyset)$ had better be 0.

$\Pi(X \cup \{y\})$ is obviously $\Pi(X) \cdot y$. So $\Pi(\emptyset)$ had better be 1.

If you are still not convinced, consider the situation where you are trying to calculate the sum of the first n elements of a list. At each stage you append an element to the list of things you have added up. Initially you have added up no items. And what is the sum? Clearly it must be 0. Now think about taking the product of a list of terms. Again, you start with the empty list. What is the product of the empty list of terms? Clearly it must be 1.

This matters, and for two reasons. The first is that by forcing yourself to think about what happens when you do these operations to the empty set you will make progress on the issues discussed in section 2.5.1. The second is that at least one of these facts—namely the fact that the disjunction of the empty set of propositions is the `false`—has genuine computational significance (in connection with *resolution* which you may encounter later when you do more Logic[9]).

What follows now is a worked example. The exercise is to show that relational composition is associative. I am writing it out in some considerable detail because altho' the result is pretty obvious it's not at all clear to first year students what a proof must look like.

We write '$x\,T\,y$' for "x is related to y by T". We saw how to define composition of relations around p 51, and we recap:

$$x\,(T \circ S)\,y \text{ iff } (\exists z)(x\,T\,z \wedge z\,S\,y)$$

We will show that $R \circ (S \circ T) = (R \circ S) \circ T$.

That is to say, for all x and y, $x\,R \circ (S \circ T)\,y$ iff $x\,(R \circ S) \circ T\,y$

(I am using capital Roman letters both as relation symbols and as variables in an algebra.)

Now, by definition of relational composition,

$$x\,R \circ (S \circ T)\,y$$

is

$$(\exists z)(x\,R\,z \wedge z\,(S \circ T)\,y)$$

and expand the second '$\circ$' to get

$$(\exists z)(x\,R\,z \wedge (\exists w)(z\,S\,w\ \wedge w\,T\,y))$$

[9]Cambridge CS students: you will encounter resolution in Ib Logic and Proof

We can pull the quantifiers to the front because—at least as long as 'u' is not free in A—'$(\exists u)(A \wedge \phi(u))$' is the same as '$A \wedge (\exists u)\phi(u)$'. (You haven't seen a proof of this but we assume that you can see that it's true.)

This gives us

$$(\exists z)((\exists w)(x\,R\,z \wedge (z\,S\,w\ \wedge w\,T\,y)))$$

and

$$(\exists z)(\exists w)(x\,R\,z \wedge (z\,S\,w \wedge w\,T\,y))$$

and we can certainly permute the two quantifiers '$\exists x$' and '$\exists w$' getting

$$(\exists w)(\exists z)(x\,R\,z \wedge (z\,S\,w \wedge w\,T\,y)).$$

We can permute the brackets in the matrix of the formula because '$\wedge$' is associative getting 'matrix'?

$$(\exists w)(\exists z)((x\,R\,z \wedge z\,S\,w) \wedge w\,T\,y).$$

We can import the existential quantifier again getting

$$(\exists w)((\exists z)(x\,R\,z \wedge z\,S\,w)\ \wedge\ w\,T\,y)$$

and reverse the first few steps by using the definition of $\circ$ to get

$$(\exists w)(x(R \circ S)w\ \wedge\ w\,T\,y)$$

and

$$x\,(R \circ S)\ \circ\ T\,y$$

as desired.

`\begin{grumble}` This is an old example sheet question. However, the more I think about it the odder it seems. Is this a proof? Is there really no more to the associativity of $\circ$ than the fact that $\wedge$ is associative? The proof above is certainly correct, but does it enlighten? What did the person who set this question expect by way of an answer? Did they think about it at all? I bet they didn't ... I think what happened is that the author thought it was a trivial fact which should accordingly be given to beginners to chew over—perhaps as part of a rite of passage. The pictures one draws of circles with dots inside them that are joined to dots in adjacent circles by lines corresponding to ordered pairs in S and T looks like the kind of notation that conceals a logical truth. Or is the associativity of relational composition something even more banal than a logical truth? Is it actually a *good thing* to have a notation that conceals it??

This example serves to remind me of how important it is for researchers to do some teaching. There are some things about your subject that you won't really understand properly until you have to think about how to explain them. The above is a case in point.

`\end{grumble}`

Indicator functions

Each set A has an **indicator** or **characteristic** function, written I_A or χ_A ('χ' is the first letter of the Greek word whence we obtained our word 'characteristic'). This is the function that, on being given an object, returns `true` if the object is in A and `false` otherwise. In lambda notation it is:

$$\lambda x.\ \texttt{if}\ x \in A\ \texttt{then true else false}.$$

Characteristic functions make it slightly easier (very slightly!) to explain why a set with n members has 2^n subsets. To notate the same fact a different way, using the vertical bars and the curly $\mathcal{P}$ we have just learned, $|\mathcal{P}(X)| = 2^{|X|}$. Indicator functions make it clear that $|\mathcal{P}(X)| = |X \to \{0,1\}|$.

You will need to know about them for a variety of reasons, for example in section 3.3.5. They will also crop up in second year computation theory, where they are invariably called **characteristic** functions rather than indicator functions. Yet another example of diverse cultures being interested in the same material.

Inverses of functions

If $f : A \to B$ is a function from A to B, and $g : B \to A$ is a function such that $(\forall a \in A)(g \circ f(a) = a)$ we say that g is a **right inverse** of f. If $(\forall b \in B)(f \circ g(b) = b)$ we say that f is a **left inverse** of g.

Warning: this is a completely different use of the word 'inverse' from the one in play when we were talking about additive and multiplicative inverse earlier.

Finally, do not forget when revising this material that the notation '$f\text{“}x$' was discussed earlier, in section 3.1.5 and not here; you might want to go back there too.

While on the subject of functions, a last notational point. In most mathematical usage the terminology '$f(x)$' is overloaded: it can denote either the value that the function f allocates to the argument x or the *set* of values that f gives to the arguments in the set x. Normally this overloading does not cause any confusion, because typically it is clear from the context which is meant. $f(\pi)$ is clearly a number and $f(\mathbb{R})$ a set of numbers. The give-away here is in the style of letter used for the argument. The font reveals to the user the ADT of the things the variable ranges over. As you can probably guess by now, I am a purist who doesn't like relying on contextual cues in this way, it prolongs our bad habit of fault-tolerant pattern-matching, and it can be avoided—because there is a notation that disambiguates these two styles of functional application without using information about the variable. $f\text{“}x$ is the set of values that f allocates to the arguments in the set x:

$$f\text{“}x = \{f(y) : y \in x\}$$

and $f(x)$ will continue[10] to be the value that f assigns to the argument x. This

[10] You need to be warned that the ambiguous use of the "$f(A)$" notation to mean both $f(A)$

double apostrophe notation is used for relations as well: $R\text{“}X$ is

$$\{y : (\exists x \in X)(R(y, x))\}.$$

This is why it's OK to write '$f^{-1}\text{“}X$' for $\{y : f(y) \in X\}$ *even though* f^{-1} might not be a function. Some people write $x.R$ for $R\text{“}\{x\}$. My first thought is that they should be shot, but there are excuses[11].

For us the commonest use of this notation is in settings like " $f\text{“}X \subseteq X$ ", which says that x is **closed under** f. Of course we can talk about sets being closed under n-ary functions with $n > 1$, and if f is an n-ary function ("a function of n variables"), then $g\text{“}(X^n) \subseteq X$ says that X is closed under g.

work these last two paragraphs in carefully. "move to the left" is a hangover from when this material was placed earlier

We have to use this move-to-the left trick to have a sensible notation for $A \times B$ as a set abstract. The usual answer is

$$A \times B := \{\langle x, y\rangle : x \in A \wedge y \in B\}$$

although both

$$\{u : (\exists x \in A)(\exists y \in B)(u = \langle x, y\rangle)\}$$

and

$$\{u : \mathtt{fst}(u) \in A \wedge \mathtt{snd}(u) \in B\}$$

are of course correct too.

3.2.9 Some Exercises

In the following questions assume the carrier set is a fixed set X, let $\mathbf{1}_X$ be the identity relation restricted to X and let U be the universal (binary) relation on X, namely, $X \times X$. The relations here are all relations-in-extension.

Exercise 39. *(a) Is $R \setminus R^{-1}$ antisymmetric? Asymmetric?*
(b) Is R XOR R^{-1} symmetrical? Antisymmetric? Asymmetric?
(c) Is the composition of two symmetrical relations symmetrical?
(d) Is the composition of two transitive relations transitive?
(e) Is the converse of a symmetrical relation symmetrical?
(f) Is the converse of a transitive relation transitive?

Exercise 40. *Can there be a function $f : X \to X$ whose graph is*
(i) a reflexive relation? or
(ii) a transitive relation? or
(iii) a symmetrical relation?

Exercise 41. *How many binary relations are there on a set of size n?*

How many of them are (a) reflexive? (b) fuzzies? (c) symmetrical? (d) antisymmetric? (e) total orders? (f) trichotomous? (g) antisymmetric and

and $f\text{“}A$ is widespread and you should expect to see it. Whether or not you propose to use it yourself is a matter between you and your conscience; i shall not pry.

[11] Not many, and certainly no good one. The root of the problem is that these ideas get used by lots of different communities each of which would rather reinvent the wheel than read a textbook written by a different community. It's all very annoying for the student.

trichotomous? (h) extensional? (i) partial orders? Do not answer part (i); (j) strict partial orders? Do not answer part (j); (k) permutations? (l) Circular orders (as in exercise 31)?

Without actually calculating the answers to (c), (d), (f), (g), (j) or (i) ...
(m) Explain why are the answers to (d) and (f) are the same;
(n) Explain why are the answers to (g) and (c) are the same;
(o) Explain why are the answers to (j) and (i) are the same.

EXERCISE 42.
If I give you a set of ordered pairs and tell you it is $(A \times B) \cup (B \times A)$ can you tell what A and B were?

This question is not particularly hard, but it might be a good idea to write out a very rigorous answer, even if only just to check that you are in complete control of the notation.

EXERCISE 43. *Let us assume the following:*

The enemy of my enemy is my friend
The friend of my enemy is my enemy
The enemy of my friend is my enemy.
The friend of my friend is my friend.

You might like to express these observations in first-order logic, using binary relation symbols like $F(\ ,\ )$ and $E(\ ,\ )$.

1. *If you have an enemy must you have a friend? If you have a friend are you friends with yourself?*

2. *Can you infer from the foregoing that two things cannot be simultaneously friends and enemies? Prove or find a countermodel.*

3. *Explain "congruence relation for ...." Assume 'friend-of' to be reflexive, so it is an equivalence relation. Think about the equivalence-classes-under-friendship. Let's also assume that—the expression "he's his own worst enemy" notwithstanding—'enemy-of' is irreflexive.*

   (i) How does 'enemy-of' "lift" to these equivalence classes? Is 'friend-of' a congruence relation for 'enemy-of'?
   (ii) How many equivalence classes can an equivalence class be hostile to?
   (iii) Explain how your answer to (ii) partitions the domain.

4. *Clearly 'enemy-of' is not transitive, but it does have a property that is rather like transitivity. Can you describe this feature exactly, and state it for a binary relation R in the style in which you know how to state that R is transitive? Look at the footnote if you need a hint[12] but try hard to do it without.*

5. *Does the feature (analogous to transitivity) from the previous part admit a notion of closure analogous to transitive closure, symmetric closure, etc.?*

[12] A binary relation R is transitive iff $R^2 \subseteq R$.

3.3 Cardinals

We will say that two sets **have the same cardinality** (= number of elements) if and only if ("iff") there is a bijection between them. We take this as the definition of cardinality. Cardinality is what two sets have in common iff there is a bijection between them.

Why do we use the fancy word 'cardinal' instead of 'number'? (I'm not doing it just to be difficult). There are different sorts of number (something that probably was never spelled out properly to you when you were little) and emphatically not all numbers are cardinals. There are all sorts of questions to which the answer is a number. For example:

(1) How many apples have you got in that basket?
(2) How much money have you got in your bank account?
(3) What is your resting pulse rate?
(4) What is your diastolic blood pressure?
(5) How long is the hypotenuse of a right-angled isoceles triangle whose other sides are of length 1?

The answer to (1) is a cardinal number. The answer to "How many ...?" is *always* a cardinal—even if it's infinite: that's what cardinals are. The answer to (2) is an integer. (An integer number of pennies: it might be negative!) We write the set of integers as $\mathbb{Z}$. The answer to (3) and (4) are real numbers, probably given to single precision. There are purists (and I am one of them) who insist that the complex number 1, the real number 1, the cardinal number 1 and the integer 1 are all in some sense different objects. You might not be a purist, but you'd better learn how purists think, for this distinction between (for example) the real number 1 and the cardinal number 1 is made by plenty of modern programming languages.

(I've left complex numbers out of this discussion not because they aren't important—they're **extremely** important—but because they are part of continuous mathematics not discrete mathematics.)

Rationals are so called because they are *ratio*s, and in fact not just any ratios but ratios of *integers*. They are answers to questions of the kind "How much bigger/nicer/higher is A than B?" (at least if niceness, size etc is measured by an integer!). We write the set of rationals as $\mathbb{Q}$.

Real numbers measure lengths of line segments, or areas, volumes; that sort of thing. You buy potatoes by real numbers not by cardinal number. Avocados you buy by cardinal number. (You buy potatoes by the kilo whereas avocados are so much *each*.) We write the set of reals as $\mathbb{R}$.

The Greeks discovered early on that not every real number is a ratio; specifically they showed quite early on that $\sqrt{2}$ is not rational. (You will have heard of e and π and they aren't rationals either. But proving that is harder than proving than $\sqrt{2}$ is irrational!)

THEOREM 1. *$\sqrt{2}$ is not a rational.*

We first assume that it can be expressed as a rational number (the opposite of what we believe), therefore:

$$\sqrt{2} = a/b \tag{3.14}$$

with a and b reduced to their lowest terms (no common factors)

We now square both sides of the equation to get:

$$2 = a^2/b^2 \tag{3.15}$$

We now multiply both sides by b^2 to get:

$$2b^2 = a^2 \tag{3.16}$$

From this we can see that a^2 must be an even number because it is equal to something multiplied by 2, and an even number multiplied by 2 is still even, as is an odd number multiplied by 2. Indeed: a is even since the square of an odd number is odd.

Even numbers are divisible by 2 so $a = 2 \cdot c$, for some other natural number c. Therefore $a^2 = 4c^2$, and we can substitute $4c^2$ for a^2 in 3.16) to get:

$$2b^2 = 4c^2 \tag{3.17}$$

The two sides of 3.17 have 2 as a common factor so we can divide through by 2:

$$b^2 = 2c^2 \tag{3.18}$$

But using the same deduction for b as we used for a in equation in 3.16, we can show that b is even, too!

And this is our contradiction: if a and b are both even then they share at least one common factor (namely 2), but we said that a/b had been simplified so that a and b had no common factors). This means that our original assumption must be wrong, and that the square root of 2 cannot be expressed as a ratio of two whole numbers.

■

Just to check that you understand this proof, do a similar proof that $\sqrt{3}$ is irrational.

There are other kinds of numbers that we haven't seen yet. Ordinals and integers mod p. Quaternions. You definitely do *not* need to know about quaternions, and you definitely *do* need to know about integers mod p. You may manage to get away without knowing about ordinals for quite a long while yet.

There are now some trivial observations we can make about cardinality. The relation "X and Y have the same cardinality" is an equivalence relation. Most textbooks leave this as an exercise but in my experience this is something that needs to be aired.

1. Transitivity. If X and Y have the same cardinal it is because there is a bijection between them, say $f : X \to Y$. Similarly if Y and Z have the same cardinal it is because there is a bijection between them, say $g : Y \to Z$. But then $g \circ f : X \to Z$ is a bijection between X and Z.

2. Symmetry. If X and Y have the same cardinal it is because there is a bijection between them, say $f : X \to Y$. But then $f^{-1} : Y \to X$ is a bijection in virtue of which Y has the same cardinal as X.

3. Reflexivity. The identity map $X \to X$ certifies that X is the same size as X.

Must establish that a composition of two injections is an injection, and of two surjections is a surjection

Notice that all we have defined is what it is for two sets to have-the-same-cardinality; we haven't said what cardinalities *are*. And what's more, we won't! And it doesn't matter! All you need to know about cardinality is that two sets have the same cardinal iff there is a bijection between them.

I think you can safely assume that the words 'cardinal' and 'cardinality' are synonymous.

So what are cardinals? You probably never worried about what the number 1 was, since all the things you wanted to do with it or to it you could do without worrying about what it actually *was*. Perhaps you are expecting—now that you are doing rather more proper maths than hitherto—that you have to start worrying about these things. Interestingly you don't. You can think of cardinals as equivalence classes of sets under the equivalence relation of having-the-same-cardinal if you want to; but you don't have to. It's not a good idea to start worrying about what cardinals are, it can do your head in. Incredibly, there is a community of people who worry about whether or not the number 1 might be Julius Cæsar. (I'm not making this up.) You don't want to end up like them[13].

Not all cardinals are finite

Remember, all sets have cardinals, and not all sets are finite, so not all cardinals are finite. The finite cardinals are the natural numbers, $\mathbb{N}$, but we have to remember that there are other cardinals as well. Some sets are infinite ($\mathbb{N}$ is an infinite set) and some of these sets are of concern to you as computer scientists. You've never had to think about infinite sets before and all your reasoning about sets has relied on the tacit assumption that all sets are finite. If some of what follows in the coming sections seems too obvious to be worth stating, it may be that you are still making the tacit assumption that all sets are finite.

There are infinite cardinals: that was the surprise of this section. But there are no infinite reals, rationals or complexes, I can promise you that! (There are infinite ordinals, and they will matter, but we are not going to deal with them here)

[13] Do not, under **any** circumstances, google *The Cæsar Problem*. No, really.

The Order Relation on Cardinals

We can define a partial order on cardinals. We write '$|X| \leq |Y|$' to mean that there is an injection from a set of size $|X|$ (as it might be, X!) into a set of size $|Y|$ (as it might be, Y! It won't make any difference which sets of size $|X|$ or $|Y|$ you choose).

Blah congruence relation section 3.2.5

Did I say 'partial order'? Obvious that it's transitive and reflexive. (Look back at the demonstration on page 80 that the relation of "having the same cardinal" is an equivalence relation). How about antisymmetrical? You are tempted to say that if there is an injection from A into B and an injection from B into A then both injections must actually be surjections, so we are done. (Beware! Merely finding an injection $A \to B$ that is not a surjection does NOT show that $|A| < |B|$.) This argument certainly works if A and B are finite sets, and at this stage your intuition probably doesn't work freely with infinite sets so finite sets are likely to be the only kind you consider, so you think that it's obvious that $\leq$ is antisymmetrical. But actually it isn't obvious at all. Let me illustrate. Consider the two sets $\mathbb{N}$ (of natural numbers) and $\mathbb{Q}$ (of rational numbers). It is easy to describe *in*jections from $\mathbb{Q}$ into $\mathbb{N}$ and from $\mathbb{N}$ into $\mathbb{Q}$, as follows:

1. The function that sends the natural number n to the rational number n clearly is an injection from $\mathbb{N}$ into $\mathbb{Q}$.

2. Every rational can be expressed as a ratio of two naturals with no common factor, associated with a plus or minus sign. This means we can send the rational number $+(n/m)$ to the natural number $2 \cdot 3^n \cdot 5^m$, and we can send the rational number $-(n/m)$ to the natural number $3^n \cdot 5^m$. (Negative rationals go to even naturals and positive rationals go to odd naturals. And we can send the rational number 0 to the natural number 0). It's easy to check that this is an injection. (I know it looks fiddly: all injections from $\mathbb{Q}$ into $\mathbb{N}$ do!)

However it is not at all obvious that there is a bijection between $\mathbb{N}$ and $\mathbb{Q}$!

This is why we need the **Cantor-Bernstein** theorem, which tells us that $\leq$ on cardinals really is antusymmetrical as we expected. To put it another way: if we have injections $fX \hookrightarrow Y$ and $g : Y \hookrightarrow X$ then there really is a bijection between X and Y.

This rather odd feature exhibited by $\mathbb{N}$ and $\mathbb{Q}$, namely that there are injections going both ways neither of which is a bijection, could only happen because these two sets are infinite. If A and B are finite then if there is an injection $f : A \to B$ and an injection $g : B \to A$ then f (and g too for that matter) must be a surjection. More arrestingly, consider the map $\lambda n.2n : \mathbb{N} \hookrightarrow \mathbb{N}$. It's an injection from a set to itself that is not a surjection! Strange but true.... Actually we can exploit this strangeness to obtain a *definition* of infinite set:

An infinite set is one that is in bijection with a proper subset of itself.

In our present case, $\mathbb{N}$ qualifies as infinite because it is in bjiection with $\{2n : n \in \mathbb{N}\}$—the evens.

Nevertheless it is true: $\leq$ on cardinals really is antisymmetrical. The theorem that states this is the **Cantor-Bernstein** theorem. We won't prove it!

With a little bit of finagling (by means of an assumption called the axiom of choice which you will not need to know about) we can tidy cardinals up so that every cardinal is either infinite (in the above sense) or is a **natural number**.[14]

The name for the cardinal that the set of naturals and the set of rationals share is '$\aleph_0$'. ('$\aleph$' is the first letter of the Hebrew alphabet and Cantor was Jewish). It is the smallest infinite cardinal, in the sense that any set that is smaller than $\mathbb{N}$ is finite.

Exercise 44. *Look again at the box of isotopes on page 70.*
Why was I right to write the half-life of Pb^{208} as '∞' rather than '$\aleph_0$'?

You will need to know a bit about infinite cardinals, but not much. $\mathcal{P}(\mathbb{N})$ is of course of size $2^{\aleph_0}$ and there are $2^{\aleph_0}$ real numbers. (This is because every real number can be represented as a sequence of 0s and 1s, one for each natural number. So there are $\aleph_0$ independent choices from $\{0, 1\}$, making $2^{\aleph_0}$ possible outcomes. There is a certain amount of tidying up because some reals (those rational numbers whose denominators are powers of 2) have two representations as infinite binary numbers, but we won't go into that). Cantor's theorem (which we will see in section 3.3.4) tells you that infinite sets come in infinitely many different sizes: $\aleph_0 < 2^{\aleph_0} < 2^{2^{\aleph_0}} \ldots$ after all but despite this it's a fairly safe prediction that every infinite set you meet will be of size $\aleph_0$ or $2^{\aleph_0}$.

You may be wondering: is there an $\aleph_1$? There is—and an $\aleph_2$, $\aleph_3$ and so on—but you don't need to worry about them until further notice. The file `www.dpmms.cam.ac.uk/~tf/countability.pdf` tells you a bit more about some of this material (though not about $\aleph_1$) . It's designed for 1a maths students and so goes slightly beyond what you are going to need immediately. However it is probably quite digestible. local allusions

3.3.1 Stuff to fit in—message to self

Once you have introduced cardinals and have shown that equinumerosity is a congruence relation for disjoint union, product etc you exhibit a natural bijection between $A \to (B \to C)$ and $(A \times B) \to C$. What you do is exhibit two lambda terms and show that they are mutually inverse. Do this calculation explicitly.

3.3.2 Operations on Cardinals, and Curry-Howard

There are various natural operations on cardinals, and you encountered them long ago: multiplication, addition and exponentiation. These operations on cardinals correspond to operations on sets: multiplication corresponds to cartesian product and addition to disjoint union. You know about cartesian product but

[14] Q: "What about rationals and negative numbers? They're not infinite!"
A They're numbers all right, but they're not cardinals: see the digression above.

perhaps not disjoint union.. Refer back to page 21. 'Disjoint union' is an important construct, and to understand it we have to recall the inclusion-exclusion principle and the idea of multisets, if only to draw contrasts. How many things in $A \cup B$? Well, as we have seen, $|A| + |B| - |A \cap B|$, because we don't want to count things in $A \cap B$ twice. But what if we *do* want to count things twice? We might want a sort of union of A and B where we want to know, in this union, which elements came from A and which came from B. In this setting, if something appears in $A \cap B$ we want it to appear *twice* in the new union. This new kind of union is called the **disjoint union** of A and B and is written $A \sqcup B$ (sometimes also '$A + B$' using overloading of '+' because $|A \sqcup B| = |A| + |B|$.). You can think of $A \sqcup B$ as "Take everything in A, put a pink dot of paint on it, and take everything in B and put a blue dot of paint on it; the set of all painted things is the disjoint union $A \sqcup B$." Or, using slightly more words, but bringing to life the talk of copies from section 2.7, "For each a in A, make a copy of it and put a pink dot of paint on the copy; and for each b in B make a copy of it and put a blue dot of paint on the copy; the set of all things thus painted is the disjoint union $A \sqcup B$."

Notice that this set isn't the same as the multiset that is the union of A and B, because in the multiset union you can't tell which of A and B was the original home of any element of the new union multiset. You have two copies of things that were in $A \cap B$ but you can't tell them apart and you don't know which came from A and which from B. For example the union of the two multisets $\{2, 2, 3\}$ (the factors of 12) and $\{3, 5, 5\}$ (the factors of 75) is of course $\{2, 2, 3, 3, 5, 5\}$. But this is the same as the union of the two multisets $\{2, 2, 5\}$ and $\{3, 3, 5\}$: you can't tell which 3 came from the 12 and which came from the 75. This is because there is no way of distinguishing the two 3s in the factorisation of 900.

3.3.3 Natural bijections and Elementary Cardinal Arithmetic

Commutativity of Cardinal Multiplication

Some things are important and hard, and some things are important and easy. One important and easy observation is the fact that $A \times B$ is the same size as $B \times A$. You may be saying to yourself that it's obvious, because if $|A| = n$ and $|B| = m$ then $|A \times B| = n \cdot m$, but that's arguing back-to-front. The real reason is that there is a bijection between $A \times B$ and $B \times A$, and—if you think about it—it's pretty obvious what that bijection is.

A brief reality check before we go any further.

Exercise 45. *Write out a formal declaration of the obvious bijection between $A \times B$ and $B \times A$, using λ notation and* `fst` *and* `snd`*.*

The interesting thing about this bijection is that we don't need to know anything about A or B to specify it. This means that it's not merely a bijection, it's what we call a **natural** bijection.

What would a bijection be that wasn't natural? Well, there are bijections between the two sets $\{a, b, c\}$ and $\{1, 2, 3\}$ (Miniexercise: how many??) One particularly obvious one is the bijection $a \mapsto 1$, $b \mapsto 2$, $c \mapsto 3$. This is not "natural", because in order to specify it we need access to specific information about those two sets, in particular that the first set comes equipped with an alphabetical order and that members of the second are numbers and are ordered by magnitude. Another example of a non-natural bijection is the bijection between the set of all permutations of a set and the set of total orders of that set. See exercise 39. If I give you two three-membered sets and no further information beyond the fact that they are three-membered, then there is no obvious bijection for you to point to: no *natural bijection.*. For example, no natural bijection between the two sets {`&`, `%`, ,} and {`&~`, `@`, `$`,}. You might think, Dear Reader, that it would be *natural* to pair the symbols off using the order in which they appear in this paragraph, but that isn't natural in terms of the two *sets*; it natural only in terms of the (ordered!) representation of the two sets on the page.

Does your fault-tolerant pattern-matching make you think the tautology $A \wedge B \longleftrightarrow B \wedge A$ at this point ...? If it does, then for once it is not leading you astray, though it may be a little while before the connection beomes clear.

The point that I want to hang on this natural bijection is the point that the commutativity of cardinal multiplication (which is something you learned at Primary School) relies on this natural bijection.

Associativity of Cardinal Multiplication

Another fact you know about cardinal multiplication is that it is associative: $(\forall n, m, k)(n \cdot (m \cdot k) = (n \cdot m) \cdot k)$. (It's also true for multiplication of other kinds of numbers too, but those other multiplications mean something different). This fact too, relies on the existence of a natural bijection, this time on one between $A \times (B \times C)$ and $(A \times B) \times C$.

Exercise 46. *Write down a Lambda term (or, if you prefer, an* `ML` *program) for this bijection.*

It's a bit fiddly to write down, but you should be able to explain what it does.

Again, if you free associate from this to the fact that $A \wedge (B \wedge C)$ is logically equivalent to $(A \wedge B) \wedge C$ you will still be on the straight and narrow.

Associativity and Commutativity of Addition

Clearly $A \sqcup B$ is the same size as $B \sqcup A$ (The lambda term says: "Swap pink and blue spots"). And $A \sqcup (B \sqcup C)$ is the same size as $(A \sqcup B) \sqcup C$. It can do no harm to spell out the bijection. In $A \sqcup (B \sqcup C)$ there are things with only pink spots on them. (They were all in A). Then there are things that have both a pink spot and a blue spot (they started off in B) and finally there are things with two blue spots on them, and they had started off in C.

The bijection now does the following:

Anything with two blue spots has one blue spot removed;
Anything with just a pink spot is given an extra pink spot.

Notice that it doesn't seem to matter if this respotting is done simultaneously or in sequence, and if it is done in sequence it doesn't seem to matter which is done first. I don't know how significant this is. Check my working to make sure you understand what is going on, and that I haven't made a mistake. Respotting is *confluent* (see chapter 6).

Distributivity

There is a distributivity law for addition and multiplication of natural numbers, which you know:

$$(\forall abc)(a(b+c) = ab + ac) \tag{3.19}$$

This assertion boils down to the fact that there is a bijection between $A \times (B \sqcup C)$ and $(A \times B) \sqcup (A \times C)$. There is a lambda term and again it's a bit fiddly to give it but if you remember your pink dots and blue dots you should be able to describe its action in words.

This corresponds—as you are probably by now willing to predict—to the propositional tautology...

Exercise 47.
Well, which propositional tautology does equation 3.19 correspond to?

material needed here

Exponentiation and Currying

The time has now come to consider not only multiplication and addition, but also exponentiation. You may have seen the notation '$A \to B$' for the set of all functions from A to B. You may also know that there are $|B|^{|A|}$ functions from A to B. (Do you remember why?)[15]

You will remember some equations connecting exponentiation and multiplication. We are now going to check to see if they correspond to natural bijections. A good place to start is with

$$a^0 = 1 \tag{3.20}$$

0 is the cardinality of the empty set, so what equation 3.20 is trying to tell us is that there is precisely one function from a set A to the empty set. *And* that is true whatever A is! What is this function?

[15] Well, for each thing in A we have a choice of $|B|$ things to send it to, as we can send it to anything in B. These choices are independent—and we multiply independent choices (as we reminded ourselves in exercise 16 part 2)—so the answer is $|B|^{|A|}$.

That was a bit of a cheat, you may feel. (You shouldn't feel cheated if you took to heart my strictures about null objects in section 2.5.1). But this second one isn't. No doubt you remember:

$$(a^b)^c = a^{b \cdot c} \tag{3.21}$$

This means we should be looking for a bijection between $(B \times C) \to A$ and $C \to (B \to A)$.

If this is your first encounter with this bijection you might find it hard to describe, so I'll give you a lambda term, or rather two.

- If $f : B \times C \to A$ then $\lambda c.\lambda b.f(\langle b, c\rangle)$ is a map from C to $B \to A$ and so is a member of $C \to (B \to A)$. So $\lambda f.\lambda c.\lambda b.f(\langle b, c\rangle)$ is a member of $((B \times C) \to A) \to (C \to (B \to A))$. (Remember that most people will write this last term as $\lambda f c b.f(\langle b, c\rangle)$).
- If $g : C \to (B \to A)$ then $\lambda p.g(\mathtt{snd}(p))(\mathtt{fst}(p))$ is a map from $C \times B$ to A and so is in $(B \times C) \to A$. So $\lambda g.\lambda p.(g(\mathtt{snd}(p))(\mathtt{fst}(p))$ is in $(C \to (B \to A) \to ((B \times C) \to A)))$.

The second one is a bit hard to read. The lambda term $\lambda p.(g(\mathtt{snd}(p))(\mathtt{fst}(p)))$ indicates the function that, when you give it a pair p from $B \times C$, cracks it open to get the two components $\mathtt{snd}(p)$ and $\mathtt{fst}(p)$, then applies g to $\mathtt{snd}(p)$ to obtain a function from B to A, to which it then feeds $\mathtt{fst}(p)$. Some people put subscripts on the variables in contexts like this so you can tell where the arguments are coming from. So they would write

$$\lambda p_{B \times A}.(g(\mathtt{snd}(p))(\mathtt{fst}(p)))$$

for this, and

$$\lambda f_{(C \times B) \to A}.\lambda c_C.\lambda b_B.f(\langle b, c\rangle)$$

for the lambda term of the previous item.

If you find that a bit of a mouthful, try this special case. We can think of a binary relation $R \subseteq X \times Y$ as a matrix—as we saw earlier (section 3.2.3). This makes it quite easy to see such a binary relation as a function defined on members of X. Simply send each $x \in X$ to the set of things in Y to which it is related by R. (as it were, the set of places in its row where you find a 1 rather than a 0). A miniexercise:

Exercise 48. *Write down a lambda term for this function. (This is actually the same representation as the way the queue for the airplane loo is represented in section 3.2.6). You may wish to use the double apostrophe notation here ...*

exhumed material here

Curry-Howard

It's not only natural bijections that concern us in the long term, but natural maps that aren't necessarily bijections. For example, there is a natural map

from A to $B \to A$, namely $\lambda a_A.\lambda b_B.a$. After what you have been reading you will no doubt free-associate from this to the fact that $A \to (B \to A)$ is a truth-table tautology. So you can see that the connection that we have been looking at above—between tautologies and the existence of lambda terms—doesn't require the lambda terms in question to denote bijections.

This connection has subtleties that we cannot go into here. If you are intruiged by this, you might try to following exercise:

Exercise 49. *Find a lambda term for a function from $A \to (B \to C)$ to $(A \to B) \to (A \to C)$.*

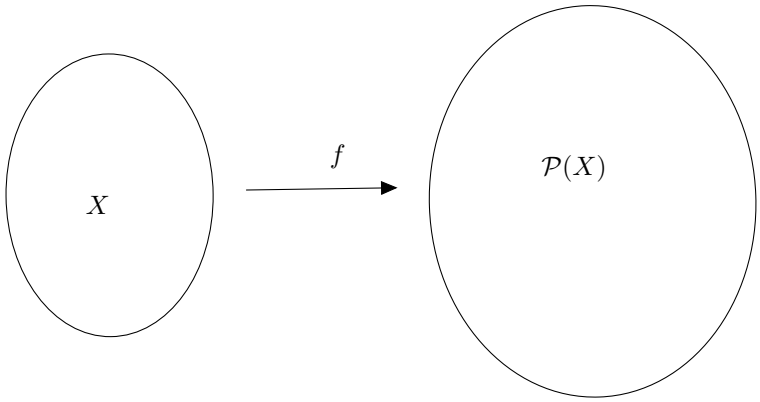
Yes, $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$ is a truth-table tautology!

This dual use of '$\to$' is no mere coincidence: it is a divine ambiguity, known as the *Curry-Howard correspondence*, on which a wealth of ink has been spilt. You will learn more about it when you study foundations of functional programming. Try, for example, [7] but not just yet!

Exercise 50. *You know that $|A \to B|$ is $|B|^{|A|}$. How many partial functions are there from A to B? (If you find you are heading towards a complicated answer, you are wrong: the answer is very simple, but most people find it hard to find)*

3.3.4 Cantor's Theorem

We've proved lots of equations, and they are all easy. There is one major theorem in the form of an *in*equation, and it is easy too. It is **Cantor's Theorem**.



Before we get stuck into the proof I want to identify a wee, *wee* assumption that we have to make. It is this: if there is a surjection from A onto B then there is an injection from B into A. This is another of those things (like the Cantor-Bernstein theorem) that is obvious when A and B are finite, but not obvious otherwise. (It's the axiom of choice again!)

This is unnneccessary

Cantor's theorem says that that $n < 2^n$. Now if $n = |X|$ then $2^n = |\mathcal{P}(X)|$. Clearly there is an injection $X \to \mathcal{P}(X)$: the singleton map $\lambda x \in X.\{x\}$ is one. So: to prove the inequality all we have to do is prove that there is no injection $\mathcal{P}(X) \to X$. In fact it's slightly easier to prove that there is no surjection $X \twoheadrightarrow \mathcal{P}(X)$ (which by assumption is the same thing) and that is what we will do. (I could have left out the bit about injections from A to B and surjections from B to A, and given instead a slightly more complicated proof that there is no injection from $\mathcal{P}(X)$ to X, but that proof is displeasingly messy. If you like, you can check and see how to do it for yourself. Determining which is easier is a delicate calculation)

The proof is now a doddle. Suppose f were a surjection from X onto $\mathcal{P}(X)$. Think about

$$\{x \in X : x \notin f(x)\}. \tag{3.22}$$

This is the set of those things in X that are not members of what f sends them to. Since f sends members of X to subsets of X, asking of a member x of X whether or not it is a member of what f sends it to is a perfectly sensible question, since x is a member of X and $f(x)$ is a subset of X.

If f is a surjection, this subset—3.22—of X must be f of something, x_0 say. Now (and I want you to work this out for yourselves) ask whether or not x_0 is a member of $\{x \in X : x \notin f(x)\}$. Think about this a bit before proceeding to the next paragraph.

If it is, it isn't, and if it isn't, it is. Clearly this is an impossible situation. How did we get into it? By assuming that f was a surjection; that ensured that 3.22 was f of something. Evidently it wasn't! ■

Time invested in understanding this proof is time well spent. The same argument is used to great effect in complexity theory, and in (for example) the proof of the unsolvability of the Halting problem for Turing machines, which you will see in a later course.

You Absolutely Must Understand This Proof.

Notice that nowhere in the proof of Cantor's theorem do we assume that X is finite. Indeed we don't even assume that it is nonempty!

3.3.5 Inclusion-Exclusion

If A and B are multisets then the number of things in $A \cup B$ is the number of things in A plus the number of things in B. Things are a bit more complicated with sets, since we don't want to count twice those things that appear in both A and B: we want to count everything only *once* even if it appears *twice*: once in A and once in B. We have the following equation:

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

Obvious, isn't it? To get the number of things in $A \cup B$ you have to subtract from $|A| + |B|$ the number of things in $A \cap B$ co's the members of $A \cap B$ are the things that get counted twice. In some sense equally obvious (but ever so slightly harder to compute) is:

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

(We subtract the number of things in $A \cap B$ because they get counted twice, and similarly $A \cap C$ and $B \cap C$. But then anything in $A \cap B \cap C$ has been counted three times and taken away three times, so it has to be put back!)

Think a bit about you might generalise this to the case where you are taking the union of several sets.

Here is a bald statement of the general principle that I found in the notes of one of my colleagues:

$$|\bigcup_{s \in S} A_s| \;=\; -\sum_{\emptyset \neq T \subseteq S} (-1)^{|T|} \cdot |\bigcap_{t \in T} A_t|. \tag{3.23}$$

This looks **extremely** scary, but it's actually nothing more than the obvious generalisation of the equations we have just seen. Let's decode this assertion carefully and without panicking.

The thing on the left hand side of 3.23 is the number of things that belong to the union of the A_s. The family of A_s is **indexed**: each A has a pointer pointing at it from an **index set**—which in this case is called 'S'. See page 44.

So, in English, 3.23 reads something like

> "The number of things in the union of the As is minus the sum—over nonempty subsets T of S—of minus-one-to-the-power-of-the-number-of-things-in-T times the number of things in the intersection of all those As whose subscripts are in T."

Or—plainer still—

> "For each nonempty $T \subseteq S$, take the intersection of all the As whose subscripts are in T (those As pointed to by elements of T) take its cardinality and take it negative or positive depending on whether T has an odd or even number of elements. Add them all together, for all such T, and make the answer positive."

The first thing to take note of is a bit of **overloading**. Primarily we write 'A_s' to denote one of the As, and the subscript is a member of the index set S. However we are now going to write 'A_T', where T is a *subset* of S not a member, and this expression will denote the intersection $\bigcap_{t \in T} A_t$ of all the As whose subscripts are in T. It's easy to detect which of these two usages are in play at any one time, because the indices themselves are *lower*-case Roman letters and the *sets* of indices are *upper*-case Roman letters. This is a common

use of the difference between upper and lower case Roman letters. Notice that $A_\emptyset$ is the whole universe of discourse—V.

Now recall indicator functions from section 3.2.8. Let I_B be the indicator function for B. Normally I_B is the function which (on being given $x \in V$) returns `true` or `false` depending on whether or not $x \in B$. However in this case we want to modify the indicator functions so that they return 0 and 1 instead of `false` and `true`. This piece of **casting** is in order that we can use *arithmetic* operations on the values of the indicator functions instead of *boolean* operations. It's universal practice in machine code. Hacky but clever. This ensures that

- $I_{\overline{B}}(x) = 1 - I_B(x)$; and
- $I_{B\cap C}(x) = I_A(x) \cdot I_B(x)$.

This second assertion generalises to

$$I_{A_1\cap A_2\ldots\cap A_n}(x) = I_{A_1}(x) \cdot I_{A_2}(x) \cdots I_{A_n}(x)$$

so, in particular (remember that $\overline{A_s}$ is the complement of A_s):

$$I_{\overline{A_1}\cap\overline{A_2}\ldots\overline{A_n}}(x) = I_{\overline{A_1}}(x) \cdot I_{\overline{A_2}}(x) \cdots I_{\overline{A_n}}(x) \tag{3.24}$$

Now $I_{\overline{A_1}}(x) = 1 - I_{A_1}(x)$ so 3.24 becomes

$$I_{\overline{A_1}\cap\overline{A_2}\ldots\overline{A_n}}(x) = (1 - I_{A_1}(x)) \cdot (1 - I_{A_2}(x)) \cdots (1 - I_{A_n}(x)) \tag{3.25}$$

For the next line consider what happens when you multiply out things like $(1-a)(1-b)(1-c)(1-d)$: you get $1-$ lots of things like $-abc$ and $+bd$ which are positive if the number of factors is odd and negative if the number of factors is even. "But shouldn't it start with a '$1-$' before the big Σ?" I hear you cry. It should indeed, but that '$1-$' is in fact included because one of the Ts you sum over is the empty set! Very cunning.

$$I_{\overline{A_1}\cap\overline{A_2}\ldots\overline{A_n}}(x) = \sum_{T\subseteq S}((-1)^{|T|} \cdot (\prod_{t\in T} I_{A_t}(x))) \tag{3.26}$$

Notice now that $I_s(x) \cdot I_t(x) = I_{\{s,t\}}(x)$, and in general $\prod_{t\in T} I_t(x) = I_T(x)$ giving

$$I_{\overline{A_1}\cap\overline{A_2}\ldots\cap\overline{A_n}}(x) = \sum_{T\subseteq S}((-1)^{|T|} \cdot (I_{A_T}(x))) \tag{3.27}$$

Now $\overline{A_1} \cap \overline{A_2} \ldots \cap \overline{A_n}$ is $\bigcap_{s\in S} \overline{A_s}$

$$I_{\bigcap_{s\in S}\overline{A_s}}(x) = \sum_{T\subseteq S}((-1)^{|T|} \cdot (I_{A_T}(x))) \tag{3.28}$$

Now, for any set X whatever, the number of things in X is simply the number of things x in V such that $I_X(x) = 1$; this number is just the sum of all $I_X(x)$, so the number of things in $\bigcap_{s\in S} \overline{A_s}$ is simply the number of $x \in V$ such that $I_{\bigcap_{s\in S} \overline{A_s}}(x) = 1$. This gives us

$$|\bigcap_{s\in S} \overline{A_s}| = \sum_{T\subseteq S} ((-1)^{|T|} \cdot |A_T|) \tag{3.29}$$

Finish this off

Applying a minus sign to both sides gets us back to equation 3.23.

3.4 Recursive Datatypes

We start this section with some revision of Mathematical Induction. You probably think you understand it already, but I want you to be Born Again!

3.4.1 Induction: revision

> *Induction can only be understood backwards, but it must be lived forwards.*
>
> Kierkegaard

Finite cardinals are called **Natural Numbers**, and the set of natural numbers is denoted with a special kind of boldface 'ℕ'. Natural numbers obey a wonderful principle called *Mathematical Induction* which you have certainly heard of. Mathematical induction is not a pleasant extra but a core skill, and one you must have; do not even think about skipping this section. Unfortunately it is also a well-known problem for beginners. There are several causes of this, and life becomes easier once they are teased apart and tackled separately.

I can think of three off the top of my head:

1. One is the old problem with fault-tolerant pattern-matching, which makes the average punter so imprecise in expressing their workings that they lose track of what they are doing. Unless you are extremely precise you won't have a hope.

2. Hypothetical reasoning is the process by which we prove $A \to B$ by assuming A and deducing B. In modern formal logic we call it "→-introduction". Lots of people find this hard. There is even a tradition in the battier parts of Western Philosophy that it cannot be done *at all* (or at least that argument by *reductio ad absurdum* is impossible). If you have done any Logic you will probably not have nightmares about this, but others may be spooked. Seek help if need be.

Explain 'antecedent'
Need to find something more to say about this

3. A lot of students are unnerved by having to think about a conditional whose antecedent is a conditional.

However, for the moment I'll address the issue about notation, because it often ensnares even students who are fairly happy about hypothetical reasoning.

Summary: we are trying to prove that every natural number has a property, F, say. We will succeed if we can do two things:

1. We establish that $F(0)$;
2. We establish that, whatever natural number n is, $F(n)$ implies $F(n+1)$.

Step 1 is the **base case**, and step 2 is the **induction step**. The base case doesn't always have to be 0; sometimes (and the example we are about to work through is a case in point) the zero case is exceptional and we don't worry about it. In this case we will start at 1.

A simple illustration

Let's take a simple example. It's simple in that the proposition we are trying to prove can be easily understood and looks fairly obvious, but the proof is difficult enough to exhibit all the standard problematic features.

Let us prove by induction that—for all n—the sum of the first n odd numbers is n^2. That is to say $F(n)$ is the assertion that the sum of the first n odd numbers is n^2.

Formula 3.30 says that the sum of the first n odd numbers is n^2. The nth odd number is of course $2n-1$.

$$\sum_{r=1}^{n}(2r-1) = n^2 \tag{3.30}$$

'n' is the **eigenvariable**: we are doing "induction **on** n".

Base case, $n = 1$ is easy.

We want to prove the induction step: if it holds for k it holds for $k+1$. What we are actually doing is a proof by Universal Generalisation (AKA $\forall$-introduction), with 'k' being the eigenvariable.

We are going to assume it true "for k" as we say, and hope to be able to deduce it for $k+1$. Notice that in this expression 'r' is bound and 'k' is free. (This terminology of free and bound variables wasn't chucked at you merely to annoy you: it's needed to properly understand stuff like this!). Do you know what I mean by this last remark? Make sure that you do before reading further!

The assumption that we have just made, that the assertion we are trying to prove does at least hold for k (and from which we intend to deduce that it holds for $k+1$) is called the **induction hypothesis**. rephrase this para

So you add the $k+1$th odd number to both sides, getting

$$(\sum_{r=1}^{k}(2r-1)) + 2k + 1 = k^2 + 2k + 1 \tag{3.31}$$

So far so good. You now have to do quite a lot of rearranging, and it may be that it helps if this is done in excruciating detail. Let's tackle the left-hand side first

$$(\sum_{r=1}^{k} 2r-1) + 2k + 1 \tag{3.32}$$

is just

$$\sum_{r=1}^{k+1} 2r - 1 \tag{3.33}$$

This is because they are both the sum of the first $k+1$ odd numbers. Formula 3.32 says "the sum-of-the-first-k-odd-numbers—with $2k+1$ added on" (and $2k+1$ just happens to be the $(k+1)$th odd number). Formula 3.33 says "the sum-of-the-first-$k+1$-odd-numbers".

So formula 3.31 has become

$$\sum_{r=1}^{k+1} 2r - 1 = k^2 + 2k + 1 \tag{3.34}$$

and now we can turn our attention to the RHS.

As any fule kno, the RHS of this is equal to $(k+1)^2$, so we get

$$\sum_{r=1}^{k+1} 2r - 1 = (k+1)^2 \tag{3.35}$$

But now notice that this formula is **exactly** the result of taking formula 3.30 and replacing 'k' (the eigenvariable) by '$k+1$' throughout. **Check this by hand so you understand it**.

And—just as formula 3.30 said that the sum of the first k odd numbers is k^2, formula 3.35 says that the sum of the first $k+1$ odd numbers is $(k+1)^2$. So we have taken an assertion about k, and deduced from it the corresponding assertion about $k+1$.

This concludes the proof of the induction step

This is something you will see in all the standard cases of proof-by-induction that the sum of the first k perfect squares, or cubes, or odd numbers, or triangular numbers, or whatever it is, is some expression in 'k'. In all these cases you will see a LHS that looks like ($\sum_{r=1}^{k}$ something-or-other) and an RHS that is some complex expression with 'k' free. Formula 3.30 is our example above. When proving the induction you infer from (as it were) formula 3.30 the result

'free'??!

of substituting '$k+1$' for 'k' in (as it were) formula 3.30. You add the kth term to both sides, which makes the LHS the sum of the first k terms plus the $k+1$th term—which is of course the sum of the first $k+1$ terms. And you add the $k+1$th term to the RHS as well and hope that you will be able to rearrange it into the result of substituting '$k+1$' for 'k' in the RHS.

A stylistic detail at this point. There is something rather special about the operation of adding 1 to a number (as opposed to the operation of adding 2, or 3, for example). This is because it is this operation-of-adding-1 that generates all the natural numbers, starting from 0. A natural number is either 0 or something one can obtain from 0 by adding 1 lots of times. For this reason we have a special notation for it: 'S', so we write '$S(x)$' instead of '$x+1$' [we don't think of S as a special case of addition but as something prior to addition] and (and this is the important part) express the inductive step in mathematical induction as "if it holds for n, it holds for $S(n)$". The point is that this special notation highlights the rôle of the operation of addition-of-1 in the genesis of the set of natural numbers.

Perhaps work this paragraph into the next section

Exercise 51. *Suppose f and g obey the declarations:*

$$f(0) := 1;\ (\forall n)(f(n+1) := (n+1)\cdot f(n))$$
$$g(0) := 1;\ (\forall n)(g(n+1) := (n+1)\cdot g(n))$$

Prove that $(\forall n \in \mathbb{N})(f(n) = g(n))$.

This shows we can use induction to prove the uniqueness of the function being defined.

3.4.2 Definition

'Recursive datatype' is the sexy, postmodern, techno-friendly way to talk about things that mathematicians used to call 'inductively defined sets'. I shall abbreviate these two words to the neologism 'rectype'.

The standard definition of the naturals is as the least (with respect to $\subseteq$) set containing zero and closed under successor, or, using some notation we have just acquired:

$$\mathbb{N} = \bigcap\{Y : 0 \in Y \wedge S\text{“}Y \subseteq Y\}.$$

$S\text{“}Y$ notation not defined yet!!

Of course $\mathbb{N}$ is merely the simplest example, but its definition exhibits the central features of a declaration of a rectype. In general, a rectype is a set defined as the smallest ($\subseteq$-least) set containing some **founders** [16] and closed under certain functions, commonly called **constructors**. (This *is* standard terminology.) $\mathbb{N}$ has only one founder, namely, 0, and only one constructor, namely, successor (often written 'S' or '`succ`': $S(x)$ is $x+1$). For the record, a founder is of course a nullary (0-place) constructor.

[16] This is not standard terminology, but I like it and will use it.

3.4.3 Structural Induction

This definition of $\mathbb{N}$ justifies induction over it. If $F(0)$ and $F(n) \to F(S(n))$ both hold, then $\{n : F(n)\}$ is one of these Y that contains 0 and is closed under S, and therefore it is a superset of $\mathbb{N}$, from which it follows that every natural number is F. It is a bit like original sin: if F is a property that holds of 0, and holds of $n+1$ whenever it holds of n, then each natural number is innoculated with it as it is born. As you are born, you arrive with a ready-minted certificate saying that you are F. Hence induction.

3.4.4 Generalise from $\mathbb{N}$

$\mathbb{N}$ is of course the simplest example of a rectype: it has only one founder and only one constructor, and that constructor is unary.

My first encounter with rectypes was when I was exposed to compound past tenses in Latin, when I was about eight. I pointed out to my Latin teacher that the construction that gives rise to the future perfect tense from the perfect could be applied to the pluperfect tense as well, and what was the resulting tense called, please? Maybe the reader has had similar experiences. In UK law, if it is a crime to do X, it is also a crime to attempt to do X or to conspire to do X. So presumably it is a crime to attempt to conspire to do X? Crimes and tenses form recursive datatypes.

The examples that will concern us here will be less bizarre. An X-**list** is either the empty object or the result of `cons`ing a member of X onto the front of an X-list. Thus a list can be thought of as a function from an initial segment of $\mathbb{N}$ to X. Thought of as a rectype, the family of X-lists has a founder (the empty list) and a single binary constructor: `cons`. In `ML` the notation '`h::t`' denotes the list obtained by `cons`ing the object `h` onto the front of the list `t`. `t` is the **tail** of `h::t`, and `h` is its **head**.

Exercise 52. *You can also think of the transitive closure $t(R)$ of a binary relation-in-extension as a rectype. What are the founders and the operations?*

We can develop analogues of mathematical induction for any recursive datatype, and I shall not spell out the details here, as we shall develop them in each case as we need them. This kind of induction over a rectype is nowadays called **structural induction**.[17]

This is an old example sheet question. You should definitely attempt it.

Exercise 53.
"We define the length of a Boolean proposition by recursion as follows:

[17] Historical note: Russell and Whitehead called it **ancestral induction** because they called the transitive closure of a relation the **ancestral** of the relation. (This is because of the canonical example: the transitive closure of the parent-of relation is the ancestor-of relation.) I used their terminology for years–and I still think it is superior–but the battle for it has been lost; readers should not expect the word 'ancestral' to be widely understood any longer, though they may see it in the older literature.

$|a| = 1$,
$|\top| = 1$,
$|\bot| = 1$,
$|A \wedge B| = |A| + |B| + 1$,
$|A \vee B| = |A| + |B| + 1$,
$|\neg A| = |A| + 1$.

We define a translation which eliminates disjunction from Boolean expressions by the following recursion:

$tr(a) = a$, $tr(\top) = \top$, $tr(\bot) = \bot$,
$tr(A \wedge B) = tr(A) \wedge tr(B)$,
$tr(A \vee B) = \neg(\neg tr(A) \wedge \neg tr(B))$,
$tr(\neg A) = \neg tr(A)$.

Prove by structural induction on Boolean propositions that

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

for all Boolean propositions A."

3.4.5 An Induction Exercise Concerning Evaluation

This section isn't actually difficult, but it relies on truth-tables, and it might look a bit scary, so don't feel guilty if you want to postpone it and come back to it later. Confident students should be fine.

I'm assuming that you are happy with truth-table definitions of the operations $\wedge$ and $\vee$ on booleans (perhaps you prefer the notations '`AND`' and '`OR`'). A **valuation** is a function from propositional letters to booleans. You can think of a valuation as a row of a truth-table, an assignment of boolean values to every propositional letter in sight. Given a complex formula A and a valuation v, one can *evaluate* A according to v and obtain a truth-value.

Clearly there is a function E: formulæ $\times$ valuations $\to$ booleans.

Exercise 54. *Write code for such a function E in your favourite functional programming language.*[18]

The recursion you have written does not contain any instruction as to the mechanics of calculating the answer. One can evaluate *lazily* or *strictly*. For example: suppose that at some point in the computation-of-the-truth-values-of-A-according-to-v you have determined, for some subformula A' of A, that A' is `false` according to v. Suppose $A' \wedge B$ is the next subformula for us to tackle. If we are evaluating "lazily" we can get clever and say "Ah! Since A' has evaluated to `false` we know already that $A' \wedge B$ must evaluate to `false`, so we don't need to compute the truth-value of B!" This (clever) strategy is

[18] In case you were wondering, your favourite functional programming language is `ML`!

called "*lazy* evaluation". In contrast, the strategy of *strict evaluation* requires us to compute the truth value of *all* subformulæ of an input before we try to compute the truth-value of the input. Strict is bottom-up.

Thus there are **two** functions E_s and E_l (strict evaluation and lazy evaluation ...) but these two functions have *three* arguments not two. Their three arguments are: a formula, a valuation, and a *time*; and the recursive declaration for E_l will have base clauses like

If A is atomic, then $(\forall n \in \mathbb{N})(E_s(A, v, n) = E_l(A, v, n) := v(A))$.
and

$$E_s(A, v, 0) = E_l(A, v, 0) := v(A),$$

and recursive clauses like (for example)

$$E_l(A \vee B, v, t+1) = \texttt{if } E_l(A, v, t) = \texttt{ true then true else}\ldots$$

Exercise 55. *Write code for E_l and E_s in your favourite functional programming language.*

It's probably obvious to you that E_s and E_l will give the same end result. Perhaps it even looks *so* obvious that you think it's not actually worth proving. However, this fact *is* worth proving, and for two reasons. One is that it is a useful exercise in induction, and the other is that if our valuations are allowed to be partial functions it ceases to be true!

Exercise 56. *(Easy)*
Find A and v to illustrate how $(\forall t > 0)(E_s(A, v, t) \neq E_l(A, v, t))$ can happen if v is not total.

However, things are better behaved if we assume that all our valuations are total functions. So, let's make this assumption *pro tem*. Then, as we observed above, E_s and E_l will give the same end result. It's worth thinking about how one would state this last fact properly. So shield the rest of this pdf from your eyes for the moment and give the matter some thought. The next exercise sets out my attempt at formulating this obvious fact in a way that one might prove.

Then we want to

Exercise 57. *Show that:*
For all valuations v and all formulæ A and all but finitely many t,

$$E_l(A, v, t) = E(A, v) \text{ and } E_s(A, v, t) = E(A, v).$$

This is pretty obvious, but it's not totally straightforward to prove. It's certainly obvious that you are going to have to do some induction... but an induction on what? On 't'? Or a structural induction on the subformula relation? This exercise is an object lesson in getting straight quite what it is you are proving by induction, stating the induction hypothesis carefully and being clear in your own mind what kind of induction you are doing.

3.4.6 Well-founded induction

Well-founded relations and induction

Suppose we have a set with a binary relation R on it, and we want to be able to infer

$$\forall x\ \psi(x)$$

from

$$(\forall x)[(\forall y)(R(y,x)\ \rightarrow \psi(y)) \rightarrow \psi(x)].$$

In words, we want to be able to infer that everything is ψ from the news that you are ψ as long as all your R-predecessors are ψ. y **is an R-predecessor of** x if $R(y,x)$. Notice that there is no "case $n = 0$" clause in this more general form of induction: the premiss we are going to use implies immediately that a thing with no R-predecessors must have ψ. The expression "$(\forall y)(R(y,x) \rightarrow \psi(y))$" is called the **induction hypothesis**. The first line says that if the induction hypothesis is satisfied, then x is ψ too. Finally, the inference we are trying to draw is this: **if** x has ψ whenever the induction hypothesis is satisfied, **then** everything has ψ. When can we do this? We must try to identify some condition on R that is equivalent to the assertion that this is a legitimate inference to draw in general (i.e., for any predicate ψ).

Why should anyone want to draw such an inference? The antecedent says "x is ψ as long as all the immediate R-predecessors of x are ψ", and there are plenty of situations where we wish to be able to argue in this way. Take $R(x,y)$ to be "x is a parent of y", and then the inference from "children of blue-eyed parents have blue eyes" to "everyone has blue eyes" is an instance of the rule schematised above. As it happens, this is a case where the relation R in question does *not* satisfy the necessary condition, for it is in fact the case that children of blue-eyed parents have blue eyes and yet not everyone is blue-eyed.

To find what the magic ingredient is, let us fix the relation R that we are interested in and suppose that the inference

$$\frac{(\forall y)(R(y,x) \rightarrow \psi(y)) \rightarrow \psi(x)}{(\forall x)(\psi(x))}$$

has failed for some choice ψ of predicate.[19] Then we will see what this tells us about R. To say that R has the magic ingredient all we have to do is stipulate that this failure (whatever it is) cannot happen for any choice of ψ.

Let ψ be some predicate for which the inference fails. Consider the set of all things that are *not* ψ. Let x be something with no R-predecessors. Then all R-predecessors of x are ψ (vacuously!) and therefore x is ψ too. This tells us that if y is something that is not ψ, *then there must be some y' such that $R(y',y)$ and y' is not ψ either*. If there were not, y would be ψ. This tells us that the collection of things that are not ψ "has no R-least member" in the sense that everything in that collection has an R-predecessor in that collection.

[19] This is a common way of representing arguments in logic: premisses above and conclusions below the line.

Thus we can see that if induction fails over R, then there is a subset X of the carrier set (to wit, the extension of the predicate for which induction fails) such that every member of X has an R-predecessor in X.

One might have expected that for the inference to be good one would have had to impose conditions on both R and ψ. It is very striking that there should be a condition on R alone that is enough by itself for this inference to be good for *all* ψ. All we have to do is exclude the possibility of the domain of R having any such pathological subsets and we will have justified induction over R. Accordingly, we will attach great importance to the following condition on R:

Definition 4. *R is* **well-founded** *iff every nonempty subset X of the domain of R has an element x such that all the R-predecessors of x lie outside X. (x is an "R-minimal" element of X.)*

This definition comes with a health warning: it is easy to misremember. The only reliable way to remember it correctly is to rerun in your mind the discussion we have gone through: well-foundedness is precisely what one needs a relation R to have if one is to be able to do induction over R. No more and no less. The definition is not memorable, but it is reconstructible.

A **well-ordering** is a well-founded strict total order. (No well-founded relation can be reflexive, so well-founded orders have to be of the strict flavour). Perhaps we should have some examples of well-orderings. Obviously any finite total order will be a well-order! What about infinite well-orderings? The only natural example of an infinite well-ordering is one we have already seen– $\langle \mathbb{N}, <_{\mathbb{N}} \rangle$. Notice that the real line $\langle \mathbb{R}, <_{\mathbb{R}} \rangle$ is not a well-ordering, for it is a simple matter to find sets of real numbers with no least element, for example, the set of all real numbers strictly greater than 0. This set has a lower bound all right, namely 0, but this lower bound is not a member of the set and so cannot be the least member of it.[20]

Exercise 58. *One can define well-orderings as relations that are trichotomous and well-founded.*

Exercise 59.
A pointwise product of two well-founded (strict) partial orders is a well-founded (strict) partial order.
A lexicographic product of two well-founded (strict) partial orders is a well-founded (strict) partial order.

It is not hard to see that for a finite binary structure to be well-founded it is neccessary and sufficient for it to have no loops.

It's clearly necessary that there should be no loops, since a loop is manifestly a subset with no least element! Sufficiency is slightly harder, but you should

[20] It is important not to get confused (as many people do) by the fact that every set of reals has a *greatest lower bound*. For example, $\{x \in \mathbb{R} : x > 0\}$ has no least member, but it does have a greatest lower bound, which is of course 0. Notice that $0 \notin \{x \in \mathbb{R} : x > 0\}$!!

have no difficulty persuading yourself that if you have a subset with no least element, then you can use it to build a loop.

With infinite structures, absence of loops remains a necessary condition of course, but it is no longer sufficient: the negative integers with the relation $\{\langle n, n-1\rangle : n \in \mathbb{Z}^-\}$ has no loops, but it is still not well-founded. With the help of an apparently minor assumption we can show that this is the only badness that can happen in infinite ill-founded structures.

This means that one can safely think of a wellfounded relation R as a relation that "**has no infinite descending chains**". That is to say, there is no sequence $x_1, x_2, x_3, \ldots$ where, for all $n \in \mathbb{N}$, $R(x_{n+1}, x_n)$.

The official definition of well-foundedness is a lot more unwieldy than the definition in terms of descending sequences. In consequence, it is very easy to misremember it. A common mistake is to think that a relation is well-founded as long as its domain has a minimal element, and to forget that *every nonempty subset* must have a minimal element. The only context in which this definition makes any sense at all is induction, and the only way to understand the definition or to reconstruct it is to remember that **it is cooked up precisely to justify induction; it serves no other purpose**.

THEOREM 2. *R is a well-founded relation iff we can do well-founded induction over the domain of R.*

Proof: The left-to-right inference is immediate: the right-to-left inference is rather more interesting.

What we have to do is use R-induction to prove that every subset of the domain of R has an R-minimal element. But how can we do this by R-induction? The trick is to prove by R-induction ("on x") that every subset of the domain of R to which x belongs contains an R-minimal element. Let us abbreviate this to "x is R-**regular**".

Now let x_0 be such that every R-predecessor of it is R-regular, but such that it itself is not R-regular. We will derive a contradiction. Then there is some $X \subseteq dom(R)$ such that $x_0 \in X$ and X has no R-minimal element. In particular, x_0 is not an R-minimal element of X. So there must be x_1 s.t. $R(x_1, x_0)$ and $x_1 \in X$. But then x_1 is likewise not R-regular. But by hypothesis everything R-related to x_0 was R-regular, which is a contradiction.

Therefore everything in $dom(R)$ is R-regular. Now to show that any subset X of $dom(R)$ is either empty or has an R-minimal element. If X is empty, we are all right. If it is not, it has a member x. Now we have just shown by R-induction that x is R-regular, so X has an R-minimal element as desired. ■

Well-foundedness is a very important concept throughout Mathematics, but it is usually spelled out only by logicians. (That is why you read it here first.) Although the rhetoric of Mathematics usually presents Mathematics as a static edifice, mathematicians do in fact think dynamically, and this becomes apparent in mathematical slang. Mathematicians often speak of *constructions* underlying proofs, and typically for a proof to succeed it is necessary for the construction

in question to terminate. This need is most obvious in computer science, where one routinely has the task of showing that a program is well-behaved in the sense that every run of it halts. Typically a program has a main loop that it goes through a number (which one hopes will be finite!) of times. The way to prove that it eventually halts is to find a parameter changed by passage through the loop. There are various sorts of parameters that can play this rôle:

- The simplest illustration is the `count` variable to be found in many programs. A `count` variable is not affected by any of the code within the loop other than the decrement command that decrements it at the start (or on the end) of each pass.
- Sometimes the rôle is played by a program variable that is not explicitly decremented at the start of each pass in the way a `count` variable, but is decremented as a side-effect of what happens on each pass.
- In general we look for a parameter that need not be a program variable at all, but merely some construct put together from program variables.

In all cases we want the parameter of interest to take values in a set X with a binary relation R on it such that

1. at each pass through the loop the value of the parameter changes from its old value v to a new value v' such that $\langle v, v' \rangle \in R$ and
2. any sequence v_0, v_1 ... where for all n, $\langle v_n, v_{n+1} \rangle \in R$, is finite.

(If you were expecting this sentence to end "is eventually constant", look ahead to section 3.4.7, p. 106.)

If we can do this, then we know that we can only make finitely many passes through the loop, so the program will halt. Condition (2) is of course the descending-sequence version of well-foundedness.

Exercise 60. *Go back and look at exercise 9 again. This time do the following:*

1. *Show that the colour of the ball that remains is determined by b and w alone and hence the algorithm determines a function of b and w.*
2. ** How can you be sure that the algorithm always terminates whatever you pluck out of the bag at each stage? hint: think about the lexicographic order of $\mathbb{N}^2$.*

As we saw in section 3.2.3, we can think of binary relations as digraphs, where there is a vertex for each element of the domain and an edge from a to b if a is related to b. This is a very natural thing to do in the present context, since we can also think of the arrows as representing a possible step taken by the program in question. It also gives us a convenient way of thinking about composition and transitive closures. a is related to b by R^n if there is a path of length n from a to b in the digraph picture of R, and a is related to b by the

transitive closure of R if there is a path from a to b at all. It also makes it very easy to see that the transitive closure of a symmetric relation is symmetric, and makes it obvious that every subset of a well-founded relation is well-founded. This makes it easy to explain why pointwise products of well-founded relations are well-founded.

Recursion on a well-founded relation

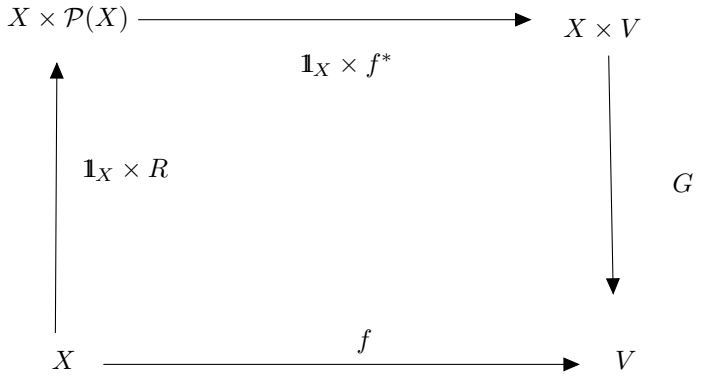
THEOREM 3. *Let $\langle X, R\rangle$ be a well-founded structure and $g : X \times V \to V$ be an arbitrary (total) function. Then there is a unique total function $f : X \to V$ satisfying $(\forall x \in X)(f(x) = g(x, f\text{“}\{y : R(y, x)\}))$*

Here V is the universe, so that when we say "$g : X \times V \to V$" we mean only that we are not putting any constraints on what the values of g (or its second inputs) are to be.

Let us have a brief cogitate about what this says, before we start trying to prove it. It says that if R is wellfounded, then if we try to define a function f by saying "take the set of all the values of f for arguments R-related to x, and do g to that set and x; call the result $f(x)$", then we succeed in defining f uniquely.

Proof: The idea is very simple. We prove by R-induction that for every $x \in X$ there is a unique function f_x satisfying $(\forall y)({}^*R(y, x) \to f_x(y) = g(y, f_x\text{“}\{z : R(z, y)\}))$. We then argue that, if we take the union of the f_x, the result will be a function, and this function is the function we want. ∎

The following commutative diagram might help.



The function corresponding to the top left-right arrow, written "$\mathbb{1}_X \times f^*$ is $\lambda a.\langle \texttt{fst}\ a, f\text{“}(\texttt{snd}\ a)\rangle$" means "leave the first component alone and translate the second under f". It may be that you won't understand until you have finished digesting section 3.3.2 how R can be thought of as a map. The map R is not the map from X into $\mathcal{P}(X)$ corresponding to R (we saw in section 3.3.2 how every subset of $X \times X$ corresponds to a map $X \to \mathcal{P}(X)$) but instead the map

that sends a pair $\langle x, y\rangle$ to $\langle x, \{z : R(z, y)\}\rangle$. ($V$ contains everything: not just junk but sets of junk as well, so you don't have to worry about whether values of g are sets or junk.)

The reason this crops up here is that all rectypes–since they are generated by functions–will have a sort of **engendering relation**[21] that is related to the functions that generate the recursive datatype rather in the way that $<_{\mathbb{N}}$ is related to the successor function. The engendering relation is that binary relation that holds between an object x in the rectype and those objects "earlier" in the rectype out of which x was built. Thus it holds between a formula and its subformulæ, between a natural number and its predecessors and so on. Put formally, the (graph of the) engendering relation is the transitive closure of the union of the (graphs of the) constructors.

The (graph of, extension of) the engendering relation is itself a rectype. For example, $<_{\mathbb{N}}$ is the smallest set of ordered pairs containing all pairs $\langle 0, n\rangle$ with $n > 0$ and closed under the function that applies S to both elements of a pair (i.e., $\lambda p.\langle S(\texttt{fst}\ p), S(\texttt{snd}\ p)\rangle$).

The following triviality is important.

THEOREM 4. *The engendering relation of a rectype is well-founded.*

Proof: Let X be a subset of the rectype that has no minimal element in the sense of $<$, the engendering relation. We then prove by structural induction ("on x") that $(\forall y)(y < x \rightarrow y \notin X)$. ■

Theorem 4 means that we can always do well-founded induction over the engendering relation. In this simplest case, $\mathbb{N}$, this well-founded induction is often called *strong* induction or sometimes *course of values* induction. Quite often arguments by well-founded induction are presented in contrapositive form. We first establish that, if there is a counterexample to what we are trying to prove, then there is an earlier counterexample. So the set of counterexamples has no least element and so by well-foundedness must be empty. The standard example of this style of proof is due to Fermat, who proved that $x^4 + y^4 = z^2$ has no nontrivial solutions in $\mathbb{N}$. It uses the fact that all pythagorean triples are of the form $a^2 - b^2$, $2ab$, $a^2 + b^2$ to show that for any solution to $x^4 + y^4 = z^2$ there is one with smaller z. This gives us a proof by well-founded induction on $<_{\mathbb{N}}$ that there are no solutions at all. The details are fiddly, which is why it is not an exercise. The examples which follow are more straightforward, and the example with which we start is the most natural use of this technique known to me.

define contrapositive

EXERCISE 61. *A square can be dissected into finitely many squares all of different sizes (see Gardner, [3]). Why can a cube not be dissected into finitely many cubes all of different sizes?*

EXERCISE 62. *Show that the relation $\trianglelefteq$ on $\mathbb{N}$ defined by $n \trianglelefteq m$ iff $n < m \leq 100$ is well-founded.*

[21] This is not standard terminology.

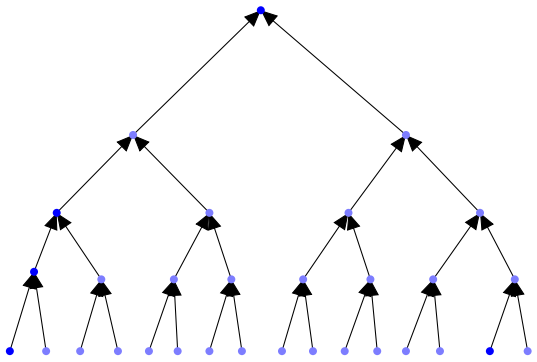
Consider the two functions f and g defined thus:
$f(x) = \texttt{if } x > 100 \texttt{ then } (x - 10) \texttt{ else } f(f(x + 11))$*;*
$g(x) = \texttt{if } x > 100 \texttt{ then } (x - 10) \texttt{ else } 91$*.*
Prove that f and g are the same functions-in-extension.

You might like to google 'McCarthy's 91 function'.

3.4.7 An Illustration from Game Theory

Before we leave recursion on a well-founded relation it might be helpful to have an illustration. Discrete games in which all plays are finite and no draws are allowed always have a winning strategy for one player or another. Let us prove this.

First we'd better say what a discrete game is. Let X be an arbitrary set. (X is intended to be the set of *moves*.) $\texttt{lists}(X)$ is the set of (finite) lists of members of X. $\texttt{lists}(X)$ has a natural tree structure, and indeed it is usually thought of as a tree. Let G be a subset of $\texttt{lists}(X)$ that is **closed under shortening** (i.e., initial segments [sometimes called *prefixes*] of lists in G are also in G). Naturally 'G' is intended to suggest 'Game'. G is a subset of $\texttt{lists}(X)$ rather than the whole of it because some moves might not always be legal: P-K4 isn't legal if there is a piece on K3! There is a map v defined on the **endpoints** of G (sequences in G with no proper end-extensions in G) taking values in the set $\{\texttt{I}, \texttt{II}\}$.



Players `I` and `II` play a game by picking elements of X alternately, with `I` playing first, with their choices constrained so that at each finite stage they have built a finite sequence in G.

The game ends when they reach an endpoint of G, at which point v tells them who has won. For the purposes of illustration we will assume that all plays in G are finite.

Exercise 63. *How many (binary) games of length n are there? (Easy)*

Show that, for every $n \in \mathbb{N}$, and for every game of length n, one of the two players `I` *and* `II` *must have a winning strategy.*

Let $\texttt{II}_n$ *be the proportion of these games for which player* `II` *has a winning strategy: what is the limit of* $\texttt{II}_n$ *as n gets large?*

perhaps rearrange a bit

The connection with well-foundedness is that this condition is captured by saying that the relation "$s \in G \wedge t \in G$ and s is an end-extension of t" is well-founded. (If you cannot work out which way round to read this, just note: one way round it is *obviously* well-founded: what we mean is that it is well-founded the other way round too.)

Next we need the notion of **even** and **odd** positions. A sequence from X of even length is a position when it is `I`'s turn to move; a sequence from X of odd length is a position when it is `II`'s turn to move. Clearly, if s is an even position and even one of its children (positions to which `I` can move at his next move) is labelled '`I`', then we can label s '`I`' too, since `I` can win from there. Similarly, if s is an *odd* position and *all* its children (positions to which `II` can move at his next move) are labelled '`I`', then s can be labelled '`I`' too. This ratcheting up the upside-down tree of lists that comprise G is a recursive definition of a labelling extending v that—because of well-foundedness—is defined on the whole of G. Thus the empty sequence ends up being labelled, and the lucky owner of the label has a winning strategy.

It is very important that no assumption has been made that X is finite, nor that there is a finite bound on the length of lists in G. Notice also that these games are nothing to do with the games of chapter 7.

Other definitions of well-foundedness

It is clearly an immediate consequence of our definition of well-foundedness that any well-founded relation must be irreflexive. Nevertheless, one could define a relation $R \subseteq X \times X$ to be well-founded if $(\forall X' \subseteq X)(\exists x \in X')(\forall x' \in X')(R(x', x) \to x = x')$. This definition of well-foundedness has a "descending chain" version too: "every R-chain is eventually constant". This definition is more appealing to some tastes. It has the added advantage over the other definition that it distinguishes between a well-ordering of the empty set (which will be the empty relation) and a well-ordering of the singleton $\{x\}$, which will be the relation $\{\langle x, x\rangle\}$. In contrast, according to the other definition, the empty relation is not only a well-ordering of the empty set, but it is also a well-ordering of the singleton $\{x\}$!

It is a miniexercise to verify that each concept of well-foundedness is definable in terms of the other. The situation is rather like that with regard to strict and nonstrict partial orders.

Structural induction again

We know that structural induction holds for rectypes, but we could deduce it from the well-foundedness of the engendering relation if we wished. Take the example of $\mathbb{N}$. Suppose we know that 0 has property F, and that whenever n

has property F so does $S(n)$. Then the set of naturals that are *not* F (if there are any) will have no least member and therefore, by well-foundedness of $<_{\mathbb{N}}$, will be empty.

This holds in general: we can deduce structural induction from the well-foundedness of the engendering relation. For example, if we can prove $(\forall n)(\Phi(n))$ by a well-founded induction over $<_{\mathbb{N}}$, then we can prove $(\forall n)(\forall m <_{\mathbb{N}} n)(\Phi(m))$ by structural induction.

Other uses of well-foundedness

Intuitions of well-foundedness and failure of well-foundedness are deeply rooted in common understandings of impossibilities. For example: it is probably not unduly fanciful to claim that the song "There's a hole in my bucket, dear Liza" captures the important triviality that a process that eventually calls itself with its original parameters will never terminate. The attraction of tricks like the ship-in-a-bottle seems to depend on the illusion that two processes, each of which (apparently) cannot run until it has successfully called the other, have nevertheless been successfully run. A similar intuition is at work in the argument sometimes used by radical feminists to argue that they can have no (nonsexist) surnames, because if they try to take their *mother's* surname instead of their fathers, then they are merely taking their *grandfather's* surname, and so on. Similarly one hears it argued that, since one cannot blame the person from whom one catches a cold for being the agent of infection (for if one could, they in turn would be able to pass the blame on to whoever infected them, and the process would be ill-founded[22]), so one cannot blame anyone at all. This argument is used by staff in STD clinics to help their patients overcome guilt feelings about their afflictions.

The reader is invited to consider and discuss the following examples from the philosophical literature.

1. "In every judgement, which we can form concerning probability, as well as concerning knowledge, we ought always to correct the first judgement, deriv'd from the nature of the object, by another judgement, deriv'd from the nature of the understanding. 'Tis certain a man of solid sense and long experience ought to have, and usually has, a greater assurance in his opinions, than one who is foolish and ignorant, and that our sentiments have different degrees of authority, even with ourselves, in proportion to the degrees of our reason and experience. In the man of the best sense and longest experience, this authority is never entire; since even such-a-one must be conscious of many errors in the past, and must still dread the like for the future. Here then arises a new species of probability to correct and regulate the first, and fix its just standard and proportion. As demonstration is subject to the control of probability, so is probability liable to a new correction by a reflex act of the mind, wherein the nature

[22] Unless one can blame Eve!

of our understanding, and our reasoning from the first probability become our subjects.

"Having thus found in every probability, beside the original uncertainty inherent in the subject, a new uncertainty deriv'd from the weakness of that faculty, which judges, and having adjusted these two together, we are oblig'd by our reason to add a new doubt deriv'd from the possibility of error in the estimation we make of the truth and fidelity of our faculties. This is a doubt, which immediately occurs to us, and of which, if we wou'd closely pursue our reason, we cannot avoid giving a decision. But this decision, though it shou'd be favourable to our preceding judgement, being founded only on probability, must weaken still further our first evidence, and must itself be weaken'd by a fourth doubt of the same kind and so *ad infinitum*; till at last there remain nothing of the original probability, however great we may suppose it to have been, and however small the diminution by every new uncertainty. No finite object can subsist under a decrease repeated *in infinitum*; and even the vastest quantity, which can enter into human imagination, must in this manner be reduc'd to nothing."

Hume (1739) Book I Part IV, Section 1, pp 5–6.

2. "Volitions we postulated to be that which makes actions voluntary, resolute [etc.]. But . . . a thinker may ratiocinate resolutely, or imagine wickedly . . . . Some mental processes then can, according to the theory, issue from volitions. So what of the volitions themselves? Are they voluntary or involuntary acts of mind? Clearly either answer leads to absurdities. If I cannot help willing to pull the trigger, it would be absurd to describe my pulling it as voluntary. But if my volition to pull the trigger is voluntary, in the sense assumed by the theory, then it must issue from a prior volition and from that another *ad infinitum*."

Ryle (1983) pp. 65–6.

EXERCISE 64. *Hume seems to be saying that if we multiply together infinitely many numbers all between* 0 *and* 1 *then the product must be zero, but this is incorrect. Prove Hume wrong by considering the product*

$$\prod_{1<i<j\in\mathbb{N}} (i^2-1)/i^2.$$

(of 3/4, 8/9, 15/16 . . . *).*

Chapter 4

Some Elementary Number Theory

Number Theory is a relatively recent development in Discrete Mathematics courses. It became important because it's the mathematics that underlies cryptography, and cryptography became a Hot Topic for Computer Science really only with the advent of the internet and the consequent urgent need for secure secret communication between computers.

The usual ambition for a number theory slot in a level one Discrete Mathematics course is coverage of the RSA algorithm for public-key cryptography. That is what we shall aim for here!

But we'll start with something a bit more basic and familiar which will launch us in the right direction.

4.1 Different bases

Q: What goes "Pieces of nine! Pieces of nine!"?

A: A Parroty error.

Binary, octal, decimal and hexadecimal. They are all **positional** notations (polynomial notations). The only difference between them is the **base**. "Positional"? The meaning of a symbol depends very sensitively on where it appears in the formula. Does '1' mean *one* or *ten* or *one hundred*? It depends where it is. We take the positional nature of our decimal system for granted but we shouldn't. Roman numerals are not positional, or at least not in the same way (the 'I' in 'IX' doesn't mean the same as the 'I' in 'XI').

Familiarity with bases other than decimal and binary is not as important as it used to be, because familiarity with octal and hexadecimal is useful primarily to assembly language programmers, and the proportion of computer users who need skill in writing assembly languages is shrinking all the time. (One of my

students said to me "*Assembly language programmers are the proletariat of the information age*".)

Exercise 65. *There are tests for divisibility by* 3*, by* 9*, and by* 11 *in base* 10*; tests for divisibility by* 7 *and by* 9 *in base* 8*, and tests for divisibility by* 15 *and by* 17 *in hexadecimal. Do you know these tests? If you do know them, can you explain why they work?*

Euclid's Proof that there are Infinitely many Primes

We start with an old chestnut. A prime number is a natural number with no factors other than itself and 1. Euclid proved that there are infinitely many primes. His proof is simplicity itself.

Suppose there are only finitely many primes, so that P, the set of all primes, is a finite set. Then we can multiply them all together to get ΠP, which will be a natural number. Add 1 to obtain $(\Pi P) + 1$. This is a natural number too. Is it prime? It might be, but even if it isn't we know that none of its prime factors can belong to P. (After all, no number can divide into both n and $n + 1$, can it!). Either way we know we have a prime that is not in P. This contradiction proves that P wasn't finite. ■

A historical subtlety: Euclid would not have described this line of reasoning as showing that there are infinitely many primes; Greek mathematics did not see infinite quantities in the same way we did. The Greeks would have taken this as a proof that there is no largest prime.

4.2 Euclid's Algorithm

The idea is to find the highest common factor of two natural numbers x and y. The key fact is that anything that divides x and y divides $x - y$. This tells us that the HCF of x and y is the same as the HCF of x and $x - y$ (assuming $x > y$—otherwise it's y and $y - x$.) So, if I want to find HCF(x, y), I should start with two natural numbers x and y and then, at each stage, subtract the smaller of the two numbers that I have from the larger and replace the larger number with the result of that subtraction. For example, the HCF of 39 and 231 is the same as the HCF of 39 and $231 - 39 = 192$. So if I start with 39 and 231, at the next stage I have 39 and 192. The HCF of 39 and 192 is the same as the HCF of 39 and $192 - 39 = 153$. And so on.

The HCF of the pair-of-numbers-in-hand is a loop invariant, and when the process stops with the two elements of the pair equal then we have found the HCF.

For example if we start with the pair $(12, 18)$ we obtain $(12, 6)$ and then $(6, 6)$. If we start with the pair $(7, 25)$ we obtain $(7, 18)$, then $(7, 11)$, then $(7, 4)$, $(3, 4)$, $(1, 3)$ and finally $(1, 1)$.

If the bigger number is *much* bigger than the smaller one then we could end up subtracting the smaller one many times, and we would be able to save

ourselves time by conflating lots of these subtractions together by dividing the bigger number by the smaller and keeping only the remainder. For example, that way—to take our $(7, 25)$ example, we would have missed out $(7, 18)$ and $(7, 11)$, and gone straight to $(7, 4)$.

If we keep track of what we are doing when we run Euclid's algorithm on two natural numbers a and b (by keeping an eye on the remainders at each division, among other things) we can not only find HCF(a, b) (hereafter (a, b) as promised on page 50) but we can even find two integers x and y such that

$$ax - by = (a, b) \tag{4.1}$$

I shall not explain how this can be done (since we don't need it for our narrow task of climbing Mount RSA) but I can give you material on this if you wish. rephrase this

4.3 Modular Arithmetic

I mentioned earlier that there is another kind of number: integers mod p. It's easy to check that, for any natural number n, the equivalence relation "x and y have the same remainder on division by n" is a congruence relation for $+$ and $\times$.

That is to say since (if we care only about the remainder mod n of the answer) the $+$ and $\times$ operation don't notice if we replace an argument by some thing with the same remainder mod n, we can think of $+$ and $\times$ as taking for their arguments the equivalence classes under this relation, rather than the numbers themselves.

This gives us the integers mod n. How is it best to think of these numbers? Let's illustrate with integers mod 5. The equivalence classes are $\{0, 5, 10, \ldots\}$, $\{1, 6, 11, \ldots\}$, $\{2, 7, 12, \ldots\}$, $\{3, 8, 13, \ldots\}$ and $\{4, 9, 14, \ldots\}$. Usually it's easier to identify these equivalence classes with their smallest members, so that—for example—the integers mod 5 is the set $\{0, 1, 2, 3, 4\}$, equipped with the multiplication and addition tables

$\times$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

What sort of arithmetic do these numbers obey? It's easy to check that addition and multiplication are commutative as before, and that addition distributes over multiplication as usual. There is an additive unit, which is of course (the equivalence class of) 0. Of course (the equivalence class of) 1 is a multiplicative unit. What equation 4.1 tells us now is that if n is a prime, then the integers mod n have multiplicative inverses. Consider equation 4.1 again, this time where a is a prime p

$$bx - py = 1 \tag{4.2}$$

(The RHS is 1, because—since p is a prime—$(p, b) = 1$.)
But then we have

$$bx = py + 1 \tag{4.3}$$

This says that bx is one more than a multiple of p. But this says precisely that (the equivalence class of) x is a multiplicative inverse of (the equivalence class of) b mod p.

Actually we didn't need p to be prime: all we really needed was that the right-hand side of equation 4.3, the HCF, should be 1. So we can say:

Euclid's algorithm tells us that a has a multiplicative inverse mod n as long as n and a are coprime.

"these"...?

These are called **integers mod** n. One disconcerting difference between them and all the other kinds of number you know is that they have no natural order to them: no sense of magnitude. No "greater than". However they do have a *circular order*, like the numbers of a clock face: 1 comes after 12. (recall exercise 31)

4.3.1 Euler's theorem

Euler's totient function: $\phi(n)$ is the size of the set

$$\{m < n : m \text{ and } n \text{ have no factors in common}\}.$$

Slightly more formally (remembering that we were warned on page 50 that (x, y) would sometimes mean "the highest common factor of m and n"):

$$\phi(n) := |\{m < n : (m, n) = 1\}| \tag{4.4}$$

This set is sometimes called U_n, and its members are sometimes called *units*. The important point for us at the moment is that members of U_n are precisely the numbers that have multiplicative inverses mod n.

Euler's theorem says that ...

Theorem 5. *If $(a, n) = 1$ (which is to say that a and n are co-prime) then $a^{\phi(n)} = 1$ (mod n)*

So what happens if I multiply a by a member u of U_n? u and a are both prime to n so their product $a \cdot n$ is also prime to n and will be in U_n (or at least its remainder mod n will be). So multiplying members of U_n by a simply moves them around. Indeed we can say more that that. If u and v are two distinct members of U_n then au and av are also members of U_n (we've seen this already) and are distinct. Let's prove this.

Suppose

$$au = av \tag{4.5}$$

a and n are coprime so a has a multiplicative inverse mod n, which we will write a^{-1}. Multiply both sides of equation 4.5 by a^{-1} to obtain

Suppose

$$a^{-1}au = a^{-1}av \tag{4.6}$$

which gives $u = v$.

So multiplication-by-a is just a permutation of U_n. So

$$\prod U_n = \prod_{i \in U_n} a \cdot i \tag{4.7}$$

because the two sets over which we are taking the products are one and the same set!

Now

$$\prod_{i \in U_n} a \cdot i = (\prod U_n) \cdot a^{\phi(n)} \tag{4.8}$$

We get this by collecting all the a's together, and noting that there are $\phi(n)$ of them.

But what do we get if we multiply together all the units in U_n? They all have multiplicative inverses, and the multiplicative inverses are also in U_n, so they all cancel, giving 1. More detail needed here?

Modular exponentiation is easy

There is one consequence of this that we may as well minute now. Euler's theorem means that modular exponentiation is easy to calculate. What is $3^{1,000,000,000}$ mod 257? Well, $\phi(257) = 256$ so Euler's theorem tells us that $3^{256} = 1$ mod 257. So any power of 3^{256} is likewise equivalent to 1 mod 257. But $1,000,000,000$ is a multiple of 256 so $3^{1,000,000,000}$ is a power of 3^{256} and therefore $3^{1,000,000,000}$ mod 257 must be 1!

That looks like a special case, because 256 divides into $1,000,000,000$.

But, had we wanted $3^{1,000,000,007}$ mod 257 instead, we would only have had to calculate 3^7 mod 257.

So, in general, how do we compute a^b mod n when $(a, n) = 1$? Well, as the above illustration shows, all we need to worry about is the remainder of b on division by $\phi(n)$. So, no matter how huge b is, we never have to calculate a to the power of anything bigger than $\phi(n)$.

This tells us that modular exponentiation is no more difficult than division. Use inclusion-exclusion to show ϕ is multiplicative.

Exercise 66. *The game of Sylver Coinage was invented by Conway, Berlekamp and Guy. See [5]. It is played by two players,* `I` *and* `II`*, who move alternately, with* `I` *starting. They choose natural numbers greater than 1 and at each stage the player whose turn it is to play must play a number that is not a sum of multiples of any of the numbers chosen so far. The last player loses.*

Notice that by 'sum of multiples' we mean 'sum of positive multiples'. The give-away is in the name: 'Sylver Coinage'. What the players are doing is trying at each stage to invent a new denomination of coin, one that is of a value that cannot be represented by assembling coins of the denominations invented so far. (There is a significance to the spelling of 'silver', but I do not think we need to concern ourselves with that.)

Prove that no play of this game can go on forever.

The way to do this is to identify a parameter which is altered somehow by each move. The set of values that this parameter can take is to have a well-founded relation defined on it, and each move changes the value of the parameter to a new value related to the old by the well-founded relation. The question for you is, what is this parameter? and what is the well-founded relation?

(You should give a much more rigorous proof of this than of your answer to exercise 61 below: it is quite easy to persuade oneself that all plays are indeed finite as claimed, but rather harder to present this intuition as reasoning about a well-founded relation.)

4.4 The RSA algorithm

Let p and q be two primes. Let m be $p \cdot q$. $\phi(m)$ will be $(p-1)(q-1)$.

Alice (for some reason she is always called 'Alice') wishes to arrange matters so that people can send her messages that only she and the other party can read. She arms herself with p, q and m as above, and calculates $\phi(m)$. Now comes the clever bit. Alice chooses a number e (the 'e' is intended to suggest **encryption exponent**). This encryption exponent must be prime to $\phi(m)$; this is to ensure that it has a multiplicative inverse mod $\phi(m)$. This multiplicative inverse mod $\phi(m)$ is the **decryption exponent** and is written 'd'.

Alice announces m and e to the world. (She does *not* divulge p or q or $\phi(m)$!). Anyone who tries to calculate p or q or $\phi(m)$ apparently has only one way in: to factorise m. This seems to be very hard.

Now, if you wish to send Alice a message, you do the following

1. You code up your message as a natural number somehow, using ASCII perhaps. Let a be this number. You want a to be less than m, so you might have to chop up your message into blocks.

2. You then calculate $a^e \pmod m$, (this is why you want $a < m$!) and you send it to Alice *in clear* as the spies say: in an open way that everyone can see.

(It's worth remembering that (2) can be done quite easily: we established in section 4.3.1 that modular exponentiation can be done quickly.)

Alice receives the message. She decrypts it as follows. Let t (for Thomas) be the number she receives. She can calculate $t^d \pmod m$, since modular exponentiation is easy.

What is $t^d \pmod m$? Well, t is $a^e \pmod m$ so

$$t^d \pmod m \tag{4.9}$$

is

$$(a^e)^d \pmod m \tag{4.10}$$

which is

$$a^{ed} \pmod m. \tag{4.11}$$

Now $a^{\phi(m)}$ is 1 (mod m), by Euler's theorem. We know that d and e are multiplicative inverses mod $\phi(m)$ so we can think of $de \pmod{\phi(m)}$ as $c \cdot \phi(m)+1$ for some number c. This makes $a^{ed} \pmod m$ the same as

$$a^{c\cdot\phi(m)+1} \pmod m \tag{4.12}$$

which is

$$a^{c\cdot\phi(m)} \cdot a \pmod m \tag{4.13}$$

$$(a^{\phi(m)})^c \cdot a \pmod m \tag{4.14}$$

which simplifies, since $a^{\phi(m)}$ is 1 mod m, to

$$1^c \cdot a \pmod m \tag{4.15}$$

which is of course

$$a \pmod m \tag{4.16}$$

which is the ASCII code for my message

Do we want to use Inclusion-exclusion to show that Euler's totient function is multiplicative...?

Chapter 5

Graph theory

The word 'graph' has many uses: there is the graph of a relation or function (a relation-in-extension or function-in-extension) which is a set. There is also the graph of a function, which is a picture (the graph of $\lambda x.x^2$ is a parabola drawn in the plane). If you stop and think about this, you will see that these are really the same thing: the second is merely a depiction—a *visualisatioin*—of the first. However, here we are going to use the word in a different way.

A graph is a set of points (known as "vertices") with lines joining them. Each pair of vertices in the graph either has a line joining its two members together or it doesn't. The line between a and b (that might or might not be there) is called an *edge*. Normally we don't think of there being an edge joining a vertex to itelf. So a graph is a set of vertices with a set of edges, each edge joining *distinct* vertices. Or it can be thought of as a set of vertices with a symmetrical irreflexive relation. People in graph theory usually think of a graph as a pair $\langle V, E\rangle$ of a set V of vertices and a set E of edges.

Draw some pictures

Need some exercises here

Graphs and digraphs are very useful data structures: lots of things can be represented using them. For example, binary relations can be represented by digraphs. (Well, digraphs with loops). **Decorated graphs** are very useful not only for displaying information but also for reasoning about it. For example we can represent a network of depôts with pipelines between them by a decorated graph. Vertices represent depôts and directed edges represent the pipelines. We can decorate the vertices with numbers indicating the amount of stuff they can store, and decorate the edges with numbers indicating the rate at which they can deliver stuff. Of course if the depôts store more than one commodity (and the pipelines correspondingly transmit more than one commodity) then the vertices and edges will be decorated by more than one number. Or we might prefer to have multisets of edges between vertices (one edge for each commodity). Graphs with multiple edges in this manner are called **multigraphs**.

A **complete** graph on a set V of vertices is the graph containing all the possible edges. The **complement** $\overline{G}$ of a graph G is the graph containing precisely the edges missing from G. A graph is **connected** if for any two vertices in the graph there is a path between them. If it's not connected it is

disconnected

EXERCISE 67. *Prove that a graph and its complement cannot both be disconnected.*

If you have already done some Logic and are happy with the method of *resolution* you may wish to try proving this using resolution. *Hint: If G and $\overline{G}$ are both disconnected then there are vertices a and b that are disconnected in G and vertices c and d that are disconnected in $\overline{G}$. Invent propositional letters ac, bd and so on, which say that there is an edge between a and c, between b and d and so on.*

A **decoration** of a graph is a function from edges (or vertices) of the graph to things.

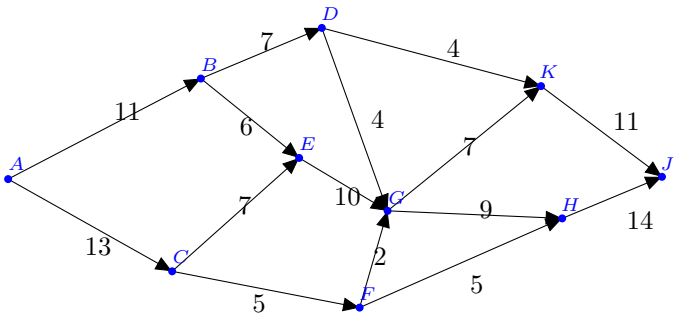
We have only one section on Graph theory, so we haven't got time or space do anything in depth. The two results we do cover give a flavour of the kind of thing that we prove and a taste of the methods of proof.

5.1 Menger's theorem

We illustrate this kind of application of graph theory by exhibiting Menger's "Min cut max flow" theorem.

Menger's theorem applies to directed graphs with a **source** and a **sink**, and which have their edges decorated with whole numbers ("capacities"). These things are also called **networks**.

In the picture below A is the source and J is the sink



(Don't worry about the numbers on the edges for the moment).

A **cut** in a digraph $G = \langle V, E\rangle$ is a set of edges which disconnects the graph. ("No path from the source to the sink") For example, the set $\{(K, J), (H, J)\}$ containing the two edges $K \to J$ and $H \to J$ is a cut[1].

[1] Yet ANOTHER use of round brackets! But this one isn't standard.

Alternatively a cut is a partition of the set of vertices into two pieces, one of which contains the source and the other contains the sink. These two ways of thinking about cuts are related: If $\{A, B\}$ is a partition of the set of vertices into two pieces, then the edges of G that join things in A to things in B form a cut (in the other sense). Similarly if we have a set of edges that disconnect the graph we can define a partition of the set of vertices into two pieces. One piece contains those vertices you can reach from the source without traversing any of the edges in the cut, and the other is the set of those vertices from which you can reach the sink without traversing any of the edges in the cut.

A graph might have its edges decorated with quantities... that could in principle be anything. (That is why the datatype of graphs is so useful). The digraph above has had its edges decorated by natural numbers, and it's the kind of picture one would dream up if one were trying to represent a network of oil pipelines: each edge is a pipeline from one pumping station to the next, and the decoration of each edge (pipeline) tells you the capacity of that edge.

Now a **flow** is an allocation of numbers to edges, where the number allocated to an edge is no more than its decoration (can't pump more oil than the pipeline will carry) and the sum of the allocations to the edges going *into* a node equals the sum of the allocations to the edges *leaving* the node. (oil doesn't get lost or created at nodes). There is an obvious notion of the **value** of a flow, namely the sum of the decorations on the edges leaving the source (or, equivalently) the sum of the decorations on the edges entering the sink.

The **value** of a cut in a decorated graph is the sum of the numbers in its decorations.

Menger's "Max flow min cut" theorem says that: in any network, the largest value that a flow can have is the same as the smallest value of a cut. It's obvious that any flow must be less than the value of any cut. (Every flow must go through every cut). It's not at all obvious that the maximum flow you can propel through a network is the same as the cheapest cut.

Let's see what we can do. Suppose we are given a flow.

We are going to colour some vertices blue. ("Blue vertices are those you can increase the flow to".) We rule that the source is blue. Thereafter if x is blue and there is an edge $x \to y$ used to less than its capacity then y is blue. It's obvious what this condition is doing, but there is a second clause which will require a bit of thought. If x is blue and there is an edge $y \to x$ which is used more than 0 then y is blue.

Ask: is the sink blue?

If yes, then there is a path source $\to$ sink on which you can increase the flow. If the only way in which we made a vertex blue was by finding that it was downstream from a blue vertex along an underused edge it would be obvious that we would improve the flow: just pump more along the trail of blue vertices joining the source to the sink. But what is the second clause doing? What is the significance of the "backward" edges?

Suppose we have—in addition to the source and the sink—two vertices a, b both blue. a is blue because there is an underused edge from the source $\to a$; b

is blue because there is an edge $b \to a$ which is used (it doesn't matter whether partially or to full capacity), and the sink is blue because there is an underused edge to it from b. How does this help? Easy! we can take some of the flow currently going from b to a and divert it so that it goes to the sink instead. That way we have increased the flow. This illustrates what we are supposed to do at each "backwards" edge. ■

If the sink is blue, we can increase the flow. So if we can't increase the flow, the sink is not blue. Let us consider this case. Think about the two-piece partition of the set of vertices into $\{\{v : \text{blue}(v)\}, \{v : \neg\text{blue}(v)\}\}$. The piece containing all the blue vertices contains the source and the piece containing the non-blue vertices contains the sink, so this partition is a cut within the meaning of the act. It is now quite easy to see that the capacity of this cut is precisely the capacity of the flow we started with. Every molecule of oil starts at a blue vertex (namely, the source) and visits blue vertices until it reaches a non-blue vertex, and once having reached a non-blue vertex it never looks back: it never sees a blue vertex again. (say more about this?) So at some point it traverses an edge from a blue vertex to a non-blue vertex—which is to say, it crosses one of the edges of the cut.

5.2 Euler's Theorem on graphs

An *Eulerian circuit* in a graph is a tour round the graph (a walk that takes you back to where you started) that visits every vertex (possibly several times) but traverses each edge precisely once. This is in contrast to a *Hamiltonian circuit* which visits each *vertex* once. You will need to know about Hamiltonian circuits later when you do complexity theory, but we will not go into any detail about them now, as they are quite hard.

Need some pictures

In contrast, there is a rather nice theorem about Eulerian circuits which we will prove. Actually, before we jump in let's decide to restrict ourselves to (finite) connected graphs only, just to keep things simple. Something similar is true for arbitrary graphs but we don't care.

We first need the concept of the *degree* of a vertex. **The degree of a vertex v is the number of edges that meet at v.**

Theorem 6. *A (finite) graph has an Eulerian circuit if and only if every vertex has even degree.*

One direction is fairly easy. Suppose there is such a circuit. Follow it round the graph. If you are walking round the graph, and are not to get trapped at any vertex v, then there must be a way out of v as well as a way in. (So you can, unlike Omar Khayyam, always go out from a different door from that through which you came in[2].) Now—since you can only use each edge once—the edges

[2] "Myself when young did eagerly frequent
doctor and saint, and heard great argument
about it and about: but evermore
came out by the same door as in I went". Khayyam tr. Fitzgerald.

at each vertex come in pairs, from the point of view of your circuit. Each time you come in on an edge, you must come out on a different edge, and this pairs up the edges that meet at v. Since you were travelling on an Eulerian circuit, you have used up all the vertices, so every vertex is of even degree.

The other direction is harder.

We are going to do a proof by wellfounded induction, and the wellfounded relation we exploit is the *subgraph* relation. We say $G_1 \prec G_2$ if G_1 is obtained from G_2 by deleting edges and/or deleting disconnected vertices. We claim that $\prec$ is wellfounded.

We will prove by induction on $\prec$ that every graph all of whose vertices have even degree has an Eulerian circuit.

So let G be a finite graph all of whose vertices have even degree, and assume that, for all $G' \prec G$, if all vertices of G' have even degree, then G' has an Eulerian circuit. We will establish that G has an Eulerian circuit too.

We will try to build a circuit. Start at any vertex, and walk around it as we did in the other direction of the proof, in no particular order. (*Caminante, no hay camino. El camino se hace al andar* said Antonio Machado). The only place at which you can ever get stuck is the vertex where you started, and that is, indeed, where you will eventually get stuck. You might of course be lucky and have visited all the vertices, in which case we have an Eulerian circuit as desired. But suppose we don't.

You do at least have a circuit, C. It might not visit all vertices. Now we delete from G all the edges that belong to C. The graph G' that remains is a proper subgraph G all of whose vertices have even degree. G' might actually be a union of disconnected subgraphs (the **components** of G') rather than one single subgraph, but this doesn't make much difference. By induction hypothesis this subgraph (or these subgraphs) have Eulerian circuits.

What happens next is a bit hand-wavy, and you will have to draw some pictures. Equip G' (or each of its components) with an Eulerian circuit. Then you join up these Eulerian circuits with the circuit C that we found in the previous paragraph to obtain an Eulerian circuit for G. Hint: think about the vertices that lie both on the Eulerian circuits for G' (or its components) and on the circuit C. We start off by trying to think of C as an Eulerian circuit for G, but we find that each of these vertices is an invitation to take a *détour* round G or one of its components.

Chapter 6

Confluence

A computer scientist's life is well-supplied with situations in which there is an object that has to be processed into a particular form, or turned into something else. The processing involves performing certain actions, probably several different actions, and each of them potentially more than once. For example when trying to convert a propositional formula into disjunctive normal form one exploits the identity

$$A \wedge (B \vee C) \longleftrightarrow ((A \wedge B) \vee (A \wedge C))$$

every time one wants to "push" a '$\vee$' "inside" a '$\wedge$'.

For example, suppose one is trying to obtain a disjunctive normal form for

$$p \wedge (q \vee (r \wedge (s \vee t))) \tag{6.1}$$

This formula has two '$\wedge$'s in it and they both have to be processed. One can attack either of them first and it's not going to make any difference to the formula one finally obtains which way round one does it. If you attack the top (leftmost) '$\wedge$' first you get

$$(p \wedge q) \vee ((p \wedge r) \wedge (s \vee t))).$$

after which you attack the third '$\wedge$' to get

$$(p \wedge q) \vee ((p \wedge r \wedge s) \vee (p \wedge r \wedge t)) \tag{6.2}$$

Alternatively you could first attack the second '$\wedge$' in 6.1 and get

$$p \wedge (q \vee (r \wedge s) \vee (r \wedge t)).$$

Now you can distribute the first '$\wedge$' over the two '$\vee$'s that follow it and end up with 6.2 as before.

A reduction or simplification process that sometimes allows you a choice of ways to reduce the object-in-hand but which nevertheless guarantees that, when

the music stops and you have no more reductions to perform, then you will have the same thing in your hand whichever choices you made *en route*, is said to be **confluent**. Clearly, when you are given a bundle of operations which you use for processing an input, it does matter greatly whether or not these operations are confluent. Not all processes are confluent and even when they are it can be hard to prove.

Let's have some examples.

• There is another confluent process one can do to propositional formulæ like 6.1. You might have a valuation in mind—a row of the truth-table—and have to decide whether 6.1 comes out true or false under this valuation. Suppose we consider the row of the truth-table in which p comes out false and all the others come out true. One might notice that p being false makes the whole thing false, and that it makes no difference in what order one computes the truth-values of other subformulæ. This is *lazy evaluation* of which you will hear more. As long as your valuations are total functions then your process of evaluating [the truth-value of] a complex formula will be confluent.

• Another example with which you will have recently (page 62) become acquainted is the process of adding ordered pairs to a relation R to obtain $t(R)$ the transitive closure of R. You roll up your sleeves, and stick your hands into R and pull out two ordered pairs p and p'. You put them both back of course, but if p happens to be $\langle x, y\rangle$ and p' happens to be $\langle y, z\rangle$ you also add the pair $\langle x, z\rangle$—unless it's already there of course. Clearly the order in which you pull the various pairs out of R and put them back makes no difference to the end result, the stage at which there are no further pairs to add co's they're all there. This process of adding pairs to R to obtain $t(R)$ is *confluent*.

• Notice that the ball-extraction process in Exercises 9 and 60 is confluent.

• Here is another pertinant example that I found on an example sheet. It's actually an exercise on Rule Induction but it serves to make other points as well (as good example sheet questions often do!)

Consider the set X of strings inductively defined over the alphabet $\{a, b\}$ as the least set which contain the string ab and

> (i) whenever X contains a string au (u a string) also contains auu (so you can double the number of bs that follow the a); and
>
> (ii) whenever X contains a string $abbbu$ (u a string) it also contains au (so you can subtract 3 from the number of bs that follow the a).

It's not hard to persuade yourself that the set of strings obtainable in this manner consists precisely of those strings that are of the form a followed by n bs where n is of the form $2^k - 3m$. Indeed that is what the exercise invites you to do. For example if you want $abbbbb$ you start with ab, double to obtain abb and again (and again) to get $abbbbbbb$ and then subtract 3 to get $abbbbb$. But if i'd doubled a *fourth* time to get ab^{16}—thereby *overshooting* you might think—i could still get to $abbbbb$ in the end. How? I subtract 3 twice to get

ab^{10}, double again to get ab^{20} then subtract 3 five times to get *abbbbb*. *The process of obtaining a* $2^k - 3m$ *string of* b*s by doubling the number of* b*s and subtracting* 3 *from the number of* b*s is confluent.* You might like to try, Dear Reader, to show that

Exercise 68. *One can obtain any number of the form* $2^k - 3m$ *from any other number of that form by repeatedly doubling and subtracting 3.*

It might feed usefully into your learning about Number Theory from earlier chapters.

Among other examples of confluent phenomena that you will encounter later are:

- the sequent rules for classical propositional logic are confluent, and
- β-reduction in λ-calculus is confluent.

But those are for later.

I would love to fit some of that "for later" material in here (as you can see, the judgement was that confluence in worth a chapter all to itself) but I haven't found a way of making it into first-year material.

But we could end with a *non*-example. The process of compiling a guest list for a party is not reliably confluent. At any stage in the compilation you may add to the growing list anyone who is on speaking terms with everyone on the list so far. But Arthur and Bertha might both be on speaking terms with everyone on the list-so-far but not with each other! So you can invite either, but once you've invited one you cannot then invite the other as well. No confluence.

This non-confluence of the process that adds invitees to obtain a party is related to the fact that there is no notion of *party closure* the way there is of *symmetric closure* and *transitive closure*. You can obtain the transitive closure of a relation by adding ordered pairs to it according to an obvious rule, and this process is confluent. See section 3.2.6.

Chapter 7

A Bit of Game Theory

Game theory is a huge ramshackle area of Mathematics. There are discrete games (e.g. chess), continuous games (lion *vs* antelope) among others. The only kind of game we are going to be concerned with here is the kind represented by a matrix. There are two players, called `I` (who picks rows) and `II` (who picks columns)—neither knowing the other's choice at the time they make their own.

In the game corresponding to the matrix

$$\begin{matrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{matrix}$$

when `I` has picked the ith row and `II` the jth column, `II` must pay `I` the sum of $£a_{i,j}$. ($a_{i,j}$ is of course the entry in the ith row and the jth column).

Games represented like this are **zero-sum** games. That is to say that the sum of the payoffs to the two players is 0.

[This jargon "zero-sum" tells us at least that the payoffs take values in a structure with an additive inverse. In what follows it's clear that the values are ordered, and I think that we can take them to be an ordered abelian group, or perhaps a module over an ordered field. Don't worry if you don't yet know the meanings of these expressions.]

(*Non*zero sum games are represented by matrices where each entry is a *pair* of numbers, being the payoffs to the two players. We will see those in section 7.1.)

We start with zero-sum games. There is a theorem due to von Neumann and Morgenstern that says that there is an optimal strategy for both players. This theorem is known as the Von Neumann and Morgenstern *minimax theorem*; see [].

What do we mean by an optimal strategy?

First let's get the concept of dominance out of the way. (The following example is from [8])

Consider the following matrix.

$$\begin{matrix} 0 & 1 & 4 & 2 \\ 2 & 0 & 1 & 4 \\ 1 & 2 & 5 & 3 \\ 4 & 1 & 3 & 2 \end{matrix}$$

Compare the first and third rows. Notice that whatever `II` does, `I` is always better off playing the third row than playing the first: in each column the entry in the third row is bigger than the entry in the first row. We say the third row **dominates** the first row. If `I` is playing sensibly he will never play the first row, so we can delete it. So we can assume they are playing this next matrix.

$$\begin{matrix} 2 & 0 & 1 & 4 \\ 1 & 2 & 5 & 3 \\ 4 & 1 & 3 & 2 \end{matrix}$$

The second column—from `II`'s point of view—dominates the fourth, so we delete the fourth, since `II` will never play it.

$$\begin{matrix} 2 & 0 & 1 \\ 1 & 2 & 5 \\ 4 & 1 & 3 \end{matrix}$$

Then—from `I`'s point of view—the third row of the result dominates the first row, so we can delete the first row, since `I` will never play it.

$$\begin{matrix} 1 & 2 & 5 \\ 4 & 1 & 3 \end{matrix}$$

Finally the third column—from `II`'s point of view—is dominated by the second. Thus `I` is never going to play either of the first two rows of the originL display, and `II` is never going to play either column 3 or column 4, in effect leaving us with

$$\begin{matrix} 1 & 2 \\ 4 & 1 \end{matrix}$$

Something to think about.... In the sequence we ran through above, the commentator (us) acting on assumptions about the rationality of the two players, deleted rows and columns *alternately*. That's just the way it happened in this case; there could have been a stage at which we could have deleted *two* (or more) rows (or columns). Suppose now we are at a stage at which there is both a row and a column that can be deleted ... might it make any difference which one we delete first? This brings up the question of of *confluence* from chapter 6. How confident can you be that we will always end up with the same

matrix once we have run out of rows and columns to delete? In general proving confluence can be hard, but in this case it's easy ... *so* easy, in fact, that I am going to leave it to you to deal with.

Exercise 69.
Prove that the process of deleting dominated rows and columns is confluent.

You should be prepared to spend some time on this exercise, and really get it to come out. (It took me a while!)

By this process of weeding we end up with a matrix in which no row dominates any other row and no column dominates any other column.

What is the sensible thing to do? The *maximin* strategy is to choose that course of action which give you the best result if things go well. For player `I` this strategy tells him to play that row whose greatest element is the greatest among all the rows available. That is to say: he is considering, for each row r, the best case b_r that can happen if he plays r. (b_r is the biggest number in row r.) He then chooses r so as to optimize (= maximise) b_r. The trouble is, `II` might not oblige by picking the column in which that number appears! In contrast the **minimax** strategy is the strategy of minimising the disasters that can befall you: acting so as to ensure that the worst-case scenario in the course of action you have embarked on is less dire than the worst-case scenarios that would have awaited you down other paths.

For player `I` this strategy tells him to play that row whose least element is the greatest among all the rows available. That is to say: he is considering, for each row r, the worst case w_r that can happen if he plays r. (w_r is the smallest number in row r.) He then chooses r so as to optimize (= maximise) w_r. That way he can be sure of compelling `II` to pay him at least the largest number that is a w_r for some row r. Let us call this value $\texttt{I}^*$. `II` has a corresponding ("dual") strategy which is to consider, for each column c, the largest number g_c in that column. She then chooses c so as to optimise (= minimise) g_c. That way she can be sure that she doesn't have to pay out more than the smallest number that is a g_c for some column c. Let us call this quantity $\texttt{II}^*$.

We will minute the following fundamental fact:

$$\texttt{I}^* \leq \texttt{II}^*$$

Proof:

Let A be the set $\{x$: for some row r, x is the smallest thing in $r\}$ and let B be the set $\{y$: for some column c, y is the largest thing in $c\}$. We want to show that everything in A is less than everything in B. Let x be an arbitrary member of A and y an arbitrary member of B. x belongs to row r, for some r, and y belongs to some column c. Clearly at the point where r and c meet we will find a number that is as big as the smallest thing in r (namely x) but no bigger than the biggest thing in c (namely y). So $x \leq y$. But x and y were arbitrary, so this tells us that everything in A is less than or equal to everything in B. So the biggest thing in A (which is $\texttt{I}^*$) can be no bigger than the smallest thing in B (namely $\texttt{II}^*$). ■

So "The sup of the infs is less than or equal to the inf of the sups." The point is often made that this inequality is the same fact as the implication $(\exists x)(\forall y)(F(x,y)) \to (\forall y)(\exists x)(F(x,y))$. What is going on? If you think of the truth-values `true` and `false` as ordered `false` $<$ `true` then you will think that the truth-value of $(\exists x)(F(x,y))$ is the sup of the truth values of $F(x,y)$ for all x, and the truth-value of $(\forall x)(F(x,y))$ is the inf of the truth values of $F(x,y)$ for all x. So the truth-value of $(\exists x)(\forall y)(F(x,y))$ is the sup of the infs, and the truth-value of $(\forall y)(\exists x)(F(x,y))$ is the inf of the sups. You may recall the same ideas cropping up in connection with Menger's theorem.

The cases where $\texttt{I}^* = \texttt{II}^*$ are games with *saddle points*. This is because such a game has an element which is the largest in its column and the smallest in its row, and we say of such an element that it is a *saddle point*.

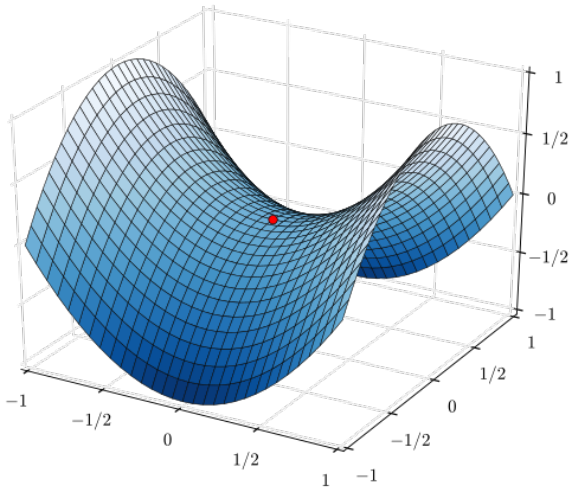


Figure 7.1: A saddle

However there are lots of games—even 2×2 games—without saddle points, like

$$\begin{array}{cc} 4 & 2 \\ 1 & 3 \end{array}$$

In this game `I` will play safe and choose the first row (the row with the largest minimum), giving $\texttt{I}^* = 2$ and `II` will pick column 2 (with the smallest maximum) giving $\texttt{II}^* = 3$.

Games lacking saddle points merit further analysis. It's not clear what is the best thing to do. We can make progress in understanding this situation if we ask instead a different question. Instead of asking "What is the best thing to do in (a single play of) this game?", one asks: "If one is going to play a lot of plays of this game, what is the best way to maximise your aggregate payoff?"

This leads us to the notion of a **mixed strategy**. A mixed strategy gives you an assignmnent of probabilities to rows (if you are `I`) or columns (if you are `II`) and you toss a suitably biased coin or roll a suitably biased die to decide which to choose at each play of the game.

It is in this sense of *best* that the Minimax theorem tells us that both players have a best strategy. For each player there is an optimal mixed strategy.

One way of thinking of this is to fill out the four points in the matrix above into a surface, as in the picture. This surface is saddle-shaped, and has a saddle point. It is this saddle point that will be the solution to the game—in a sense which we must now make clear.

Theorem 7. *In any $n \times m$ matrix game there is a (?unique?) mixed strategy for* `I` *(and* `II`*) which is optimal in the sense that no other strategy guarantees as good an average payoff.*

For simplicity's sake let us restrict ourselves to the case where the matrix is a two-by-two matrix. We are also going to assume that neither row dominates the other and neither column dominates the other. This not only restricts our analysis to the cases we have not yet covered, but also coincidentally excludes from application of this analysis cases that would cause it to do stupid things like divide by 0. So we are looking at

$$\begin{array}{cc} a & b \\ c & d \end{array}$$

and let us assume without loss of generality that both a and d are bigger than both c and b.

`I` and `II` use mixed strategies, so that `I` picks row 1 with probability x and `II` picks column 1 with probability y. We can represent this by decorating the matrix thus:

	y	$1-y$
x	a	b
$(1-x)$	c	d

Let P be the expected payoff for the pair of mixed strategies. That is to say, P is the number of £ that `II` will be paying `I` per game *on average.* (Remember P may be negative!) P is the average of the matrix entries weighted in the proportion of the time that `I` and `II` choose each entry. To be precise, P is

$$axy + b(1-y)x + cy(1-x) + d(1-y)(1-x) \qquad (P)$$

Thus on average, if `I` picks row 1 a proportion x of the time and `II` picks column 1 a proportion y of the time, on average `II` will pay `I` £P. We can rearrange this expression to

$$xy(a-b-c+d) + y(c-d) + x(b-d) + d. \qquad (7.1)$$

Now once we have fixed on constant values of a, b, c and d the formula P above gives us a function of two variables x and y. What we want is to find a value of x that makes the dependence of P on y disappear. The coefficient of y in P is

$$x((a+d)-(c+b))+(c-d) \tag{7.2}$$

For what value of x does equation 7.2 take the value 0? Clearly we must have

$$x((a+d)-(c+b))=(d-c) \tag{7.3}$$

So if x takes the value

$$x=\frac{d-c}{(a+d)-(b+c)} \tag{x^*}$$

then P takes a value from which y has disappeared. This means that if `I` and `II` play repeatedly, with x playing row one with probability x^*, it makes no difference what player `II` does.

This probably looks horrible, but it makes good sense. We are in a situation where the two rows slope in different ways: row 1 slopes down (going from left to right) and row 2 slopes upward. $(d-c)$ is a measure of the slope of row 2. If it's very nearly zero then—if you want a strategy that produces the same result (on average) whatever `II` is doing—then you want to be playing row 2 most of the time. So you want x to be very small. And it's simple to check that the top line of x^* is positive and smaller than the bottom line (which is also positive) so x^* is between 0 and 1.

Similarly we want to find a value for y that will make P's dependency on x disappear. You might like to try this by hand, pursuing calculations analogous to equations 7.2 and 7.3. But we don't actually need to. Substituting x^* into P we get

$$d-\frac{(d-c)(d-b)}{(a+d)-(b+c)} \tag{7.4}$$

which simplifies (try it!) to

$$\frac{(ad-cb)}{(a+d)-(b+c)} \tag{v}$$

Which we call the **value** of the game.

Finally we need to check that

Exercise 70.

$$\mathtt{I}^* \leq v \leq \mathtt{II}^*$$

(Remember that $\mathtt{I}^*$ *is the larger of* b *and* c*, and that* $\mathtt{II}^*$ *is the smaller of* a *and* d*).*

This resolves the unsatisfactory situation where there was a gap between the best `I` could get for himself and the worst that `II` had to endure. Using the mixed strategy `II` can ensure that nothing worse than v happens, and `I` can ensure that he does at least do as well as v. v is, in some sense, a **solution** to the game.

It is possible to find an optimal mixed strategy even if there are more than two rows or columns, but we have to use some slightly trickier mathematics to do it, and we have no space for that here. Instead we close with a brief glimpse of some very mysterious and complex generalisations, which are susceptible of wide application.

7.1 Bimatrix games

In these games the second components are the payoffs to the player picking the columns (player `II`); the first components are the payoffs to the player picking the rows (player `I`).

These games are deep and important objects, and there is a huge literature on them. The most famous game of this kind is:

Figure 7.2: The Prisoners' Dilemma

cooperate	defect	
$(3,3)$	$(1,4)$	cooperate
$(4,1)$	$(2,2)^*$	defect

The Prisoners' Dilemma is a game played by two players, both of them prisoners at the mercy of the Evil Gaoler. You have to choose between shopping your accomplice (the other player) and staying solid. The game is symmetrical: the options open to the two players are the same. If you shop your accomplice you get a new identity and a case of whisky (that's 4) and your accomplice is fed to crocodiles (that's 1). If you both shop each other, your gaolers treat you with contempt and merely shoot you (that's 2). If you both stand firm and loyal you escape with your lives and your freedom (that's 3).

It is almost impossible to overestimate the importance of this discovery of the Prisoners' Dilemma. The number of real-life situations of which it is a plausible formalisation is astonishing. This makes it not merely intruiging but important. However, since the fundamental concepts of game theory are not entirely clear, our analysis of the Prisoners' Dilemma game is not as applicable as one would like. An old problem in Ethics (I think the Greeks wrote about it) is the problem of inferring individual obligation from collective obligation. Clearly all nations are under a collective obligation to disarm themselves of their nuclear weapons. But—equally clearly—one does not infer from this that each and every nation is obliged to disarm unilaterally. In contrast, in the logically very similar situation of the problem posed by global warming and the need to cut carbon emissions,

there is a much stronger inclination to argue that individual countries are under individual obligations to cut their emissions. This discrepancy arises because it is not entirely clear how to draw correctly the parallel between the real-life situation and the formal version.

A subtly different game is **Chicken**:

Figure 7.3: Chicken

	blink	don't blink	
	$(3,3)$	$(2,4)^*$	blink
	$(4,2)^*$	$(1,1)$	don't blink

Chicken is played by two car drivers approaching each other at high speed on a single carriageway. The one who veers off the road to avoid getting killed loses a lot of cred and probably writes off the car (that's 2). Still, it beats getting killed (that's 1). If you both veer off the road you still write off the car but you don't lose as much cred (that's 3). If you stay on the road and the other player veers off, then you gain a great deal of cred—and you keep the car (that's 4).

Can one say anything sensible about these games? Well, there are always these things called **Nash equilibria**. (The Nash equilibria are the starred entries above). What is a Nash equilibrium? It's a pair of a row R and a column C such that `I` cannot do better than R—given that `II` is going to play C—and `II` cannot do better than C—given that `I` is going to choose R.

On the face of it there may be lots of Nash equilibria. We can use a fixed point theorem to show that—if we allow mixed strategies—there must be at least one. I'm not planning to prove that all bimatrix games have Nash equilibria.

And yes, it is the Nash of the Beautiful Mind, who died in a car crash on the fourth day of the ENG v NZ Lord's Test 2015.

7.1.1 Symmetrical Bimatrix Games

Consider bimatrix games where the ordered pair at $a_{i,j}$ is the flip of the ordered pair at $a_{j,i}$ and the ordered pairs on the main diagonal have two identical omponents. What's this 'main diagonal'?—doesn't make sense! After all, we could have written the columns in any order and we could have written the rows in any order, and that would jumble things up terribly. Well, it does if there is a way of identifying rows with columns: and this will happen if, for example, the options available to `I` are the same as those available to `II`. This is the case in the prisoners' dilemma for example—where we can name one row (column) 'cooperate' and the other 'defect' and in chicken, where one row (column) is 'blink' and the other is 'don't blink'.

In these circumstances one can once again represent the game by a matrix (not a bimatrix, two superimposed matrices)—even though the game is not zerosum—but in this case the entry in $a_{i,j}$ represents that payoff to strategy i

Spell all this out properly

played against strategy j and the payoff to strategy j played against strategy i is to be found in $a_{j,i}$.

It turns out that this special case has a biological motivation. (I think this is right) an **evolutionarily stable strategy** is a (possibly mixed) strategy σ such that $\langle \sigma, \sigma \rangle$ is an equilibrium pair.

They write $E(p, q)$ for the payoff to someone playing p against someone playing q.

The Bishop-Cannings theorem states that if I is a mixed strategy then $E(p, I) = E(I, I)$ for all p in the *support* of I.

Chapter 8

More Exercises

Starred exercises have model answers. The following relevant tripos questions also have model answers in the chapter of answers.

Maths tripos questions: 1995:5:4X,

Exercise 71. *An old examination question*

Let R be a relation on a set X. Define the reflexive, symmetric and transitive closures $r(R)$, $s(R)$ and $t(R)$ of R. Prove that

1. $R \circ \mathbb{1} = R$
2. $(R \cup \mathbb{1})^n = \mathbb{1} \cup (\bigcup_{i \leq n} R^i)$ for $n \geq 1$
3. $tr(R) = rt(R)$.

Show also that $st(R) \subseteq ts(R)$.

If $X = \mathbb{N}$ and $R = \mathbb{1} \cup \{\langle x, y\rangle : y = px$ for some prime $p\}$ describe $st(R)$ and $ts(R)$.

Comp Sci tripos questions: 1990:1:9, 1990:1:11, 1993:11:11, 1994:10:11, 1996:1:8, Local info: delete

8.1 Exercises on (binary) relations

1. (Do not do more than a sample of the bits of this question: if you are making any mistakes they will always be the same mistakes, and there is no point in making the discovery more than once!)

   (a) Given the operations of composition and union, express the following relations in terms of brother-of, sister-of, father-of, mother-of, son-of, daughter-of. (You may use your answers to earlier questions in answering later questions.)

parent-of
uncle-of
aunt-of
nephew-of
niece-of
grandmother-of
grandfather-of
first-cousin-of

You can also express some of the relations in the original list in terms of others by means of composition and union. Do so.

(b) Do the same to include all the in-law and step relations, by adding spouse-of to the original list. This time you may use intersection and complement as well.

(c) If the formalisation of "x is a parent of y" is

$$\text{"father-of}(x, y) \vee \text{mother-of}(x, y)\text{"}$$

(i.e., use logical connectives not $\cup$ and $\cap$. . . you will also need to use quantifiers) what are the formalisations of the other relations in the preceding list? And for a bonus point, formalise "x is the double cousin of y".[1] Hint: might need new variables!

(d) Using the above gadgetry, plus set inclusion ("$\subseteq$") formalise

Every mother is a parent;
The enemy of [my] enemy is [my] friend;
The enemy of my friend is my enemy;
The friend of my enemy is my enemy;
No friend is an enemy.

probably need to incorporate this into Exercise 43

2. * Let R be a relation on A. Recall that $r(R)$, $s(R)$ and $t(R)$ are the reflexive, symmetric and transitive closure operations respectively.

(a) Prove that $rs(R) = sr(R)$;
(b) Does R transitive imply $s(R)$ transitive?
(c) Prove that $rt(R) = tr(R)$ and $st(R) \subseteq ts(R)$;
(d) If R is symmetrical must the transitive closure of R be symmetrical? Prove or give a counterexample.

3. Think of a binary relation R, and of its graph, which will be a directed graph $\langle V, E\rangle$. On any directed graph we can define a relation "I can get from vertex x to vertex y by following directed edges" which is certainly transitive, and we can pretend it is reflexive because after all we can get from a vertex to itself by just doing nothing at all. Do this to our graph $\langle V, E\rangle$, and call the resulting relation S. How do we describe S in terms of R?

[1] Fred and Bert are double cousins if they are first-cousins in two different ways.

4. * Show that—at least if $(\forall x)(\exists y)(\langle x, y\rangle \in R)$—$R \circ R^{-1}$ is a fuzzy. What about $R \cap R^{-1}$? What about $R \cup R^{-1}$?

5. * Given any relation R there is a least $T \supseteq R$ such that T is transitive, and a least $S \supseteq R$ such that S is symmetrical, namely the transitive and symmetric closures of R. Must there also be a unique maximal (aka **maximum**) $S \subseteq R$ such that S is transitive? And must there be a unique maximal (maximum) $S \subseteq R$ such that S is symmetrical? The answer to one of these last two questions is 'yes': find a cute formulation.

6. What are the transitive closures of the following relations on $\mathbb{N}$?

   (a) $\{\langle 0, 1\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle, \ldots\}$: i.e., $\{\langle n, n+1\rangle : n \in \mathbb{N}\}$,

   (b) $\{\langle n, 2n\rangle : n \in \mathbb{N}\}$.

7. What is an antichain? Let D_n be the poset whose elements are the divisors of n, with $x \leq y$ if $x|y$. Find a maximum antichain in D_{216}.

8. Consider the set $[1, n] \subseteq \mathbb{N}$ of natural numbers from 1 up to n inclusive.

   How many partial orders $\leq$ are there on this set with the property that

   $$(\forall xyz)(z \leq x \not\leq y \not\leq x \to z \leq y)?$$

9. * Show that $R \subseteq S$ implies $R^{-1} \subseteq S^{-1}$

'monotone' re-used

10. (a) The purpose of this question was to make a point about lexicographic orders: in this case, about the order on $\mathbb{N} \times \mathbb{N}$. Check that you have really understood what is going on by rewriting the question for the scenario in which the balls come in three colours ... k colours.

    (b) (abstruse: not for a first pass) Extend the product order of $\mathbb{N} \times \mathbb{N}$ by stipulating that $\langle x, y\rangle < \langle y, S(x)\rangle$ and taking the reflexive transitive closure. Write the result $\leq_{\mathcal{B}}$. Is $\leq_{\mathcal{B}}$ a total order? Define $\leq$ between finite subsets of $\mathbb{N} \times \mathbb{N}$ by $X \leq Y$ iff $(\forall x \in X)(\exists y \in Y)(x \leq_{\mathcal{B}} y)$. Is $\leq$ wellfounded?

There's a question missing here

11. Let $K = \lambda x.(\lambda y.x)$. Evaluate $K8$, $K(K8)$ and $(KK)8$.

12. What is a wellordering? What is an initial segment of an ordering? (If you don't know what a **chain** in a poset is you probably won't know what an initial segment in a total ordering is either.) If $\langle X, \leq\rangle$ is a total order, then a *suborder* of it is a subset $X' \subseteq X$ ordered by the obvious restriction of $\leq$. Prove that $\langle X, \leq\rangle$ is a wellordering if every suborder of it is isomorphic to an initial segment of it. (The converse is also true but involves more work.)

13. * Show that $\bigcup_{n \in \mathbb{N}} R^n$ is the smallest transitive relation extending R.

14. $t(R)$ is the transitive closure of R.[2]

    (a) * Give an example of a relation R on a set of size n for which $t(R) \neq R^1 \cup R^2 \cup \ldots \cup R^{n-1}$.

    (b) Give an example of a set and a relation on that set for which $t(R) \neq R^1 \cup R^2 \cup \ldots \cup R^n$ for any finite n.

    (c) If R is reflexive then $t(R)$ is clearly the reflexive transitive closure of R (often called just the transitive closure): if you are not happy about this, attempt to write out a proof.

    (d) Find an example of an *irreflexive* relation R on a set such that $t(R)$ is indeed the reflexive transitive closure of R.

15. Think about $\mathbb{N}$ and S (the successor function on $\mathbb{N}$).

    What is the transitive closure of S?

    For $n, m \in \mathbb{N}$ when do we have $(S^n)^* \subseteq (S^m)^*$?

    When do we have $(S^n \cup (S^n)^{-1} \cup S^m \cup (S^m)^{-1})^* = (S \cup S^{-1})^*$?

16. * Show that the smallest equivalence relation containing the two equivalence relations R and S is $t(R \cup S)$.

17. If $R \subseteq X \times X$ is a fuzzy on X, is there a largest equivalence relation on X that $\subseteq R$? Is there a smallest equivalence relation on X that $\supseteq R$?

18. (a) Suppose that for each $n \in \mathbb{N}$, R_n is a transitive relation on a (presumably infinite) set X. Suppose further that for all n, $R_n \subseteq R_{n+1}$. Let R_∞ be $\bigcup_{n \in \mathbb{N}} R_n$, the union of all the R_n.
    Prove that R_∞ is also transitive.

    (b) Give an example to show that the union of two transitive relations is not always transitive.

19. For all the following choices of allegations, prove the strongest of the correct options; explain why the other correct options are not best possible and find counterexample to the incorrect ones. If you find you are doing them with consummate ease, break off and do something else instead.

    (a) An intersection of a fuzzy and an equivalence relation is (i) an equivalence relation (ii) a fuzzy (ii) neither

    (b) A union of a fuzzy and an equivalence relation is (i) an equivalence relation (ii) a fuzzy (iii) neither

    (c) An intersection of two fuzzies is (i) an equivalence relation (ii) a fuzzy (iii) neither

[2] Misleadingly people often use the expression "transitive closure of R" to mean the transitive reflexive closure of R.

(d) An intersection of the complement of a fuzzy and an equivalence relation is (i) an equivalence relation (ii) a fuzzy (iii) neither

(e) An intersection of a fuzzy and the complement of an equivalence relation is (i) an equivalence relation (ii) a fuzzy (iii) neither

(f) A union of a fuzzy and the complement of an equivalence relation is (i) an equivalence relation (ii) a fuzzy (iii) neither

(g) An intersection of a fuzzy and the complement of a fuzzy is (i) an equivalence relation (ii) a fuzzy (iii) neither

(h) An intersection of the complement of a fuzzy and the complement of an equivalence relation is (i) an equivalence relation (ii) a fuzzy (iii) neither

(i) A union of two fuzzies is (i) an equivalence relation (ii) a fuzzy (iii) neither.

20. A PER ('*P*artial *E*quivalence *R*elation') is a binary relation that is symmetrical and transitive. Is the complement of a PER a fuzzy? Is the complement of a fuzzy a PER? In each case, if it is false, find sensible conditions to put on the antecedents that would make it true.

21. Let $<$ be a transitive relation on a set X. Consider the two relations

(i) $\{\langle x, y\rangle : (x \in X) \wedge (y \in X) \wedge (x < y) \wedge (y < x)\}$ and

(ii) $\{\langle x, y\rangle : (x \in X) \wedge (y \in X) \wedge (x \not< y) \wedge (y \not< x)\}$.

(a) Are either of these fuzzies, or equivalence relations?

(b) If one of these isn't a fuzzy, but "ought to be", what was the correct definition?

(c) If the relation in (i) was an equivalence relation, what sort of relation does $<$ induce on the equivalence classes? Why is the result a mess? What extra condition or conditions should i have put on $<$ to start with to prevent this mess occurring?

(d) If (the correct definition of) relation (ii) is an equivalence relation, what can we say about the quotient?

22. Explain how to find the two greatest numbers from a set of n numbers by making at most $n + \lfloor \log_2 n \rfloor - 2$ comparisons. Can it be done with fewer? How about the 3 biggest numbers? The k biggest numbers, for other values of k? What happens to your answer as k gets bigger and bigger and approaches n?

23. * Show that the largest and smallest elements of a totally ordered set with n elements can be found with $[3n/2] - 1$ comparisons if n is odd, and $3n/2 - 2$ comparisons if n is even.

24. Construct *natural* bijections between the following pairs of sets. (For the purposes of this exercise a natural map is (expressed by) a closed λ-term; a natural bijection is (expressed by) a closed λ term (L, say) with an inverse L'. That is to say, both `compose`(L, L') and `compose`(L', L) simplify to $\lambda x.x$. Alternatively, a natural function is one you can write an `ML` program for. If you want to think more about what a natural bijection is, look at your earlier answers to the questions: If A is a set with n members, how many symmetrical relations are there on A, and how many antisymmetrical trichotomous relations are there on A? The answers to these two questions are the same, but there doesn't seem to be any 'obvious' or 'natural' bijection between the set of symmetrical relations on A and the set of antisymmetrical trichotomous relations on A. You will need to assume the existence of primitive pairing and unpairing functions which you might want to write as '`fst`', '`snd`' and $\langle x, y\rangle$.)

    $A \to (B \to C)$ and $B \to (A \to C)$;
    $A \times B$ and $B \times A$;
    $A \to (B \times C)$ and $(A \to B) \times (A \to C)$;
    $(A \times B) \to C$ and $A \to (B \to C)$;

    You may wish to try the following pairs too, but only once you have done the `ML` machinery for disjoint unions of types:

    $(A \to C) \times (B \to C)$ and $(A + B) \to C$;
    $A + (B + C)$ and $(A + B) + C$;
    $A \times (B + C)$ and $(A \times B) + (A \times C)$.

    Let Z be a set with only one element. Find a natural bijection between $(Y + Z)^X$ and the set of partial functions from X to Y.

    Find natural functions[3]

    (i) from A into $B \to A$;
    (ii) from A into $(A \to B) \to B$;
    (iii) from $A \to (B \to C)$ into $(A \to B) \to (A \to C)$;
    (iv) from $(((A \to B) \to B) \to B)$ into $A \to B$. (This one is hard: you will need your answer to (ii))
    (v) from $(A \to B) \to A$ into $(A \to B) \to B$.

    (it might help to think of these as invitations to write `ML` code of types `'a -> 'b -> 'a`, `'a -> ('a -> 'b) -> 'a` etc.)

25. What is a fixed point? What is a fixpoint combinator? Let T be your answer to the last bit of the preceding question. (So T is a natural function from $(A \to B) \to A$ into $(A \to B) \to B$.) Show that something is a fixpoint combinator iff it is a fixed point for T.

[3] These do not have to be either injective or surjective. They only have to be (total) functions.

26. Let $P = \lambda G.(\lambda g.G(gg))(\lambda g.G(gg))$. Show that P is a fixpoint combinator. Why is it not typed? After all, T was typed!

27. Give `ML` code for a higher-order function `metafact` such that any fixed point for `metafact` will turn out to be good old `fact`. Do the same for something tedious like `fibo`. Delight your class tutor by finding, for other recursively defined functions, higher-order functions for which they are fixed points.

28. Prove that $2^n - 1$ moves are sufficient to solve the Towers of Hanoi problem.

29. The fellows of Porterhouse ring each other up every sunday to catch up on the last week's gossip. Each fellow passes on (in all subsequent calls that morning) all the gossip (s)he has picked up, so there is no need for each fellow to ring every other fellow directly. How many calls are needed for every fellow to have acquired every other fellow's gossip?

30. A *triomino* is an L-shaped pattern made from three square tiles. A $2^k \times 2^k$ chessboard, whose squares are the same size as the tiles, has one of its squares painted puce. Show that the chessboard can be covered with triominoes so that only the puce square is exposed.

31. Is it possible to tile a standard (8×8) chessboard with thirty-one 2×1 rectangles (dominoes) to leave two diagonally opposite corner squares uncovered?

32. * Let $k \in \mathbb{N}$ and let $\mathcal{F}$ be a family of finite sets closed under symmetric difference, such that each set in $\mathcal{F}$ has at most k elements. How big is $\bigcup \mathcal{F}$? How big is $\mathcal{F}$?

33. Fix a set X. If π_1 and π_2 are partitions of it, we say π_1 *refines* π_2 if every piece of π_1 is a subset of a piece of π_2. What properties from the usual catalogue (transitivity, symmetry, etc.) does this relation between partitions of X have?

34. Let X be a set, and R the refinement relation on partitions of X. Let $\Pi(X)$ be the set of partitions. Why is it obvious that in general the structure $\langle \Pi(X), R \rangle$ is not a boolean algebra?

Boolean Algebra

1. Write down the truth tables for the 16 functions $\{T, \bot\}^2 \to \{T, \bot\}$, and give them sensible names (such as $\wedge, \vee, \to$, `NOR`, `NAND`). Which of these functions `splat` that you have identified have the feature that if p `splat` q and p both hold, then so does q? Why are we interested in only one of them?

2. (a) Show that `NAND` and `NOR` cannot be constructed by using $\wedge$ and $\vee$ and $\to$ alone

(b) Show that none of `NAND`, `NOR`, $\rightarrow$, $\wedge$, $\vee$ can be constructed by using `XOR` alone. (hard)

(c) Show that `XOR` and $\longleftrightarrow$ and $\neg$ cannot be defined from $\vee$ and $\wedge$ alone.

(d) *(for enthusiasts only)* Can $\wedge$ and $\vee$ be defined in terms of $\longleftrightarrow$ and $\rightarrow$?

(e) *(for enthusiasts only)* Show that all connectives can be defined in terms of `XOR` and $\rightarrow$.

(f) A *monotone* propositional function is one that will output 1 if all its inputs are 1. Show that no nonmonotone function can be defined in terms of any number of monotone functions. (easy)

3. What is a boolean algebra? Find a natural partial order on the set of functions from question 9.1 that makes them into a boolean algebra.

4. How many truth-functions of three propositional letters are there? Of four? Of n?

5. Prove that $\mathcal{P}([0,2])$ and $\{T,\perp\}^3$ are isomorphic posets.

Generating functions etc.

1. Let u_n be the number of strings in $\{0,1,2\}^n$ with no two consecutive 1's. Show $u_n = 2u_{n-1} + 2u_{n-2}$, and deduce $u_n = \frac{1}{4\sqrt{3}}[(1+\sqrt{3})^{n+2} - (1-\sqrt{3})^{n+2}]$.

2. Let m_n be the number of ways to obtain the product of n numbers by bracketing. (For example, $((ab)c)d$, $(ab)(cd)$, $(a(bc))d$, $a((bc)d)$ and $a(b(cd))$ show $m_4 = 5$.) Prove $m_n = \frac{1}{n}\binom{2n-2}{n-1}$.

3. Prove that $\mathbb{N} \times \mathbb{N}$, with the lexicographical order, is well-ordered, and that $\mathbb{N} \times \mathbb{N}$ with the product order has no infinite antichain.

4. Say $n \in m$ (where $n, m \in \mathbb{N}$) if the nth bit of m is 1. $n \subseteq m$ is defined in terms of this in the obvious way. Apparently $n \subseteq m$ iff $\binom{m}{n}$ is odd, but i have forgotten how to prove it!

5. Let p_n be the number of ways to add $n-3$ non-crossing diagonals to a polygon with n sides, thus splitting it into $n-2$ triangles. So $p_3 = 1$, $p_4 = 2$, $p_5 = 5$, and we define $p_2 = 1$. Show that

$$p_n = p_2 p_{n-1} + p_3 p_{n-2} + \ldots + p_{n-1} p_2 \quad \text{for} \quad n \geq 3\,,$$

6. and hence evaluate p_n.

7. A question on generating functions which will keep you out of mischief for an entire afternoon![4] Let A_n be the number of ways of ordering the

[4] This is problem 16 on p 64. of [6].

numbers 1 to n such that each number is either bigger than (or smaller than) *both* its neighbours. ("zigzag permutations"). Find a recurrence relation for $(A_n/2)$. (*Hint*: Think about how many zigzag permutations of $[1, n]$ there are where n appears in the rth place.) Further hints: you will have to divide the nth term by $n!$ and solve a (fairly simple) differential equation.

8. What can you say about

$$q_0 := 1; \quad q_{n+1} := 1 - e^{-q_n}?$$

Truth-definitions

An `ML` question which will prepare you for the 1b courses entitled "Logic and Proof" and "Semantics". You should make a serious attempt at—at the very least—the first part of this question. The fourth part is the hardest part and provides a serious work-out to prepare you for the semantics course. Parts 2 and 3 are less central, but are educational. *If you are a 1b student treating this as revision you should be able to do all these questions.* Cambridge references

Propositional Logic	Predicate (first-order) Logic
A recursive datatype of formulæ	A recursive datatype of formulæ
	An interpretation $\mathcal{I}$ is a domain $\mathcal{D}$ with: for each n-place predicate letter F a subset $\mathcal{I}$`F` of $\mathcal{D}^n$; for each n-ary function letter f a function $\mathcal{I}$`f` from $\mathcal{D}^n \to \mathcal{D}$. (Also constants).
`states`: `literals` → `bool`. A (recursively defined) satisfaction relation `SAT`:`states`×`fmla`→`bool`	(Fix $\mathcal{I}$ then) `states`: $vbls \to \mathcal{D}$; a recursively defined satisfaction function: $sat_{\mathcal{I}}$: formulæ × `states` → `bool`
A formula ϕ is **valid** iff for all `states` v, `SAT`(v, ϕ) =`true`.	ϕ is **true** in an interpretation $\mathcal{I}$ iff for all states v, $sat_{\mathcal{I}}(\phi, v)$ =`true`. ϕ is **valid** iff it is true in all interpretations.

1. Write `ML` code to implement the left-hand column. If you are completely happy with your answer to this you should skip the next two questions of this section.

2. (*For enthusiasts*). Expand the propositional language by adding a new unary connective, written '□'. The recursive definition of `SAT` for the language with this extra constructor has the following additional clause:

   > if s is a formula of the extended language and v is a state then `SAT`$(v, \Box s) = 1$ iff for all states v' we have `SAT`$(v', s) = 1$

   Then redo the first question with this added complication.

3. (*For enthusiasts*). Complicate further the construction of the preceding question by altering the recursive step for □ as follows. Accept as a new input a (binary) relation R between states (presumably presented as a list

of pairs, tho' there may be prettier ways of doing it). The new clause is then:

> if t is a formula of the form $\Box s$ and v is a state then $\texttt{SAT}(v, t) = 1$ iff for all states v' such that $v'\ R\ v$ we have $\texttt{SAT}(v', s) = 1$

4. Declare a recursive datatype which is the language of partial order. That is to say you have a set of variables, quantifiers, connectives etc., and two predicate letters '$\leq$ and '='. Fix an interpretation of it, possibly the `ML` type `int`. Implement as much as you can of the apparatus of states, truth etc.

5. Declare a recursive datatype which is the language of fields. That is to say you have a set of variables, quantifiers, connectives etc.; two constants '0' and '1'; a binary predicate letter '=' and two function symbols, '+' and '$\times$'. Fix an interpretation of it, for example the natural numbers below 17. Implement as much as you can of the apparatus of states, truth etc. You should be able to write code that will accept as input a formula in the language of fields and evaluate to `true` or `false` depending on what happens in the naturals mod 17.[5]

In the last two questions you could make life easier for yourself (but less natural) by assuming that the language has only finitely many individual variables. This would enable you, for example (by somehow generating all the possible states, since there are now only finitely many of them) to verify that the naturals as an ordered set are a model for the theory of total order, and that the naturals mod 17 are a model for the theory of fields.

When you have done this ask the system minders or any member of the hvg group about how to run HOL on the machines available to you. In HOL is a dialect of ML in which all the needed datatypes are predefined.

Other logic: for 1b revision, mainly

1B?!

1. π and e are transcendental. By considering the equation

$$x^2 - (\pi + e)x + \pi e = 0$$

prove a trivial but amusing fact. (If you cannot see what to do, read the footnote for a HINT).[6] What have you proved? Is your proof constructive? If not, does this give rise to a constructive proof of something else?

2. The uniqueness quantifier '$\exists!x$' is read as "There is precisely one x such that ...". Show how to express the uniqueness quantifier in terms of the old quantifiers $\exists$ and $\forall$ (and $=$).

   (a) Find an example to show that $(\exists!x\exists!y)\phi(x, y)$ is not always the same as $(\exists!y\exists!x)\phi(x, y)$

[5] It won't run very fast!

[6] At least one of $\pi + e$ and πe must be transcendental.

(b) Is the conjunction of $\exists!x\phi(x)$ and $\exists!y\psi(y)$ equivalent to something of the form $\exists!x\exists!y\ldots$?

Horn clauses

They haven't been told what Horn clauses are

1. What is a Horn clause? What is an intersection-closed property of relations?[7] Let $\phi(\vec{x})$ be a Horn clause (in which 'R' appears and the $\vec{x}$ range over the domain of R). Show that the property $\forall\vec{x}(\phi(\vec{x}))$ is intersection-closed. (The converse is also true but do not attempt to prove it!)

2. Let I be an index set, and for each $i \in I$, P_i is a person, with an associated set of beliefs, B_i. We assume (unrealistically) that each B_i is deductively closed and consistent. Show that $\bigcap_{i\in I} B_i$ is deductively closed and consistent. What about the set of all propositions p such that p is believed by a majority of people? (You may assume I is finite in this case, otherwise it doesn't make sense). What about the set of things believed by all but finitely many of the P_i? (You may assume I is *in*finite in this case, otherwise it doesn't make sense).[8]

3. We are given a set $\mathcal{L}$ of literals. We are also given a subset $K_0 \subseteq \mathcal{L}$. ('$K$' for '$K$nown'.) Also a set C_0 ('C' for 'Conditionals) of formulaæ of the kind

$$(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \to q$$

If we are given two such sets, of literals and of conditionals, we can get a new set of Known literals by adding to K_0 any q that is the consequent of a conditional all of whose antecedents are in K_0. Of course we can then throw away that conditional.

(a) Turn this into a precise algorithm that will tell us, given K_0, C_0 and a candidate literal q, whether or not q can be deduced from K_0 and C_0. By coding this algorithm in `ML`, or by otherwise concentrating the mind, determine how efficient it is.

(b) What difference does it make to the implementation of your algorithm if the conditionals are of the form

$$p_1 \to (p_2 \to (p_3 \to \ldots q)\ldots)?$$

(c) What happens to your algorithm if Conditionals are allowed to be of the (more complicated) form:

$$(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \to (q_1 \vee q_2)?$$

Can anything be saved?

[7] A horn clause is a formula of the kind $\bigwedge_{i\in I} \psi_i \to \phi$ where ϕ and all the ψ_i are atomic.
[8] What about the set of propositions believed by an even number of people?

(d) Define a quasi-order (remember what a quasi-order is?[9]) on $\mathcal{L}$ by setting $p \; R \; q$ if there is a conditional in C_0 which has q as its consequent and p as one of its antecedents, and letting $<$ be the transitive closure of R. Is $<$ reflexive? Irreflexive? Antisymmetrical? What happens if $p < p$? What happens if $(p < q) \wedge (q < p)$?

[9] And don't lose sleep over the reflexivity condition: we can add lots of silly clauses like $p \rightarrow p$ at no cost!

Chapter 9

Answers to Exercises

I am trying to adhere to a policy of reprinting the questions in *italic* and the answers/discussions in ordinary type.

Exercise 11

Hilary and Jocelyn are married. One evening they invite Alex and Chris (also married) to dinner, and there is a certain amount of handshaking, tho' naturally nobody shakes hands with themselves or their spouse. Later, Jocelyn asks the other three how many hands they have shaken and gets three different answers.

How many hands has Hilary shaken? How many hands has Jocelyn shaken?

The next day Hilary and Jocelyn invite Chris and Alex again. This time they also invite Nicki and Kim (also married). Again Jocelyn asks everyone how many hands they have shaken and again they all give different answers.

How many hands has Hilary shaken this time? How many has Jocelyn shaken?

Answer

In the general case Jocelyn asks $2n + 1$ people and gets $2n + 1$ different answers. Since the largest possible answer is $2n$ and the smallest is 0, there are in fact precisely $2n + 1$ possible answers and that means Jocelyn has got every possible answer from 0 up to $2n$ inclusive.

Think about the person who shook $2n$ hands. This person shook hands with everyone that they possibly could shake hands with : that is to say everyone except their spouse. So everybody except their spouse shook at least one hand. So their spouse shook no hands at all. Thus the person who shook $2n$ hands and the person who shook 0 hands are married. Henceforth disregard these two people and their handshakes and run the same argument to show that the person who shook $2n - 1$ hands and the person who shook 1 hands are married. And so on.

Where does this get us? It tells us, after n iterations, that the person who shook $n + 1$ hands and the person who shook $n - 1$ hands are married. So

what about the person who shook n hands, the odd man out? Well, the only person of whom Jocelyn asks this question who isn't married to another person of whom Jocelyn asks this question is Jocelyn's spouse.

Let's name people (other than Jocelyn) with the number of hands they shook. (This is ok since they all shook different numbers of hands.) $2n$ didn't shake hands with its spouse, or itself, and there are only $2n$ people left, so it must have shaken hands with all of them, in particular with Jocelyn. Correspondingly 0 didn't shake hands with anyone at all, so it certainly didn't shake hands with Jocelyn. We continue reasoning in this way, about $2n-1$ and 1. $2n-1$ didn't shake hands with itself or its spouse or with 0, and that leaves only $2n-1$ people for it to shake hands with and since it shook $2n-1$ hands it must have shaken all of them, so in particular it must have shaken hands with Jocelyn. Did 1 shake hands with Jocelyn? No, because 1 shook only one hand, and that must have been $2n-1$'s. And so on. The people who shook Jocelyn's hand were $2n$, $2n-1$, $2n-2 \ldots n+1$ and the people who didn't were 1, 2, 3, $\ldots n-1$. And of course, Jocelyn's other half. So Jocelyn shook n hands.

or do i mean 'n'?

Exercise 19

$\{n^2 : n \in \mathbb{N}\}$ and $\{n : (\exists m)(m \in \mathbb{N} \ \wedge n = m^2)\}$

Exercise 71

Let R be a relation on a set X. Define the reflexive, symmetric and transitive closures $r(R)$, $s(R)$ and $t(R)$ of R. Prove that

1. $R \circ 1 = R$
2. $(R \cup 1)^n = 1 \cup (\bigcup_{i \leq n} R^i)$ for $n \geq 1$
3. $tr(R) = rt(R)$.

Show also that $st(R) \subseteq ts(R)$.

If $X = \mathbb{N}$ and $R = 1 \cup \{\langle x, y\rangle : y = px$ for some prime $p\}$ describe $st(R)$ and $ts(R)$.

Answer:

The reflexive (symmetric, transitive) closure of R is the intersection of all reflexive (symmetric, transitive) relations of which R is a subset.

1. $R \circ 1$ is R composed with the identity relation. x is related to y by R-composed-with-S if there is z such that x is related to z by R, and z is related to y by S. Thus $R \circ 1 = R$.

2. It is probably easiest to do this by induction on n. Clearly this is true for $n = 1$, since the two sides are identical in that case. Suppose it is true for $n = k$.
$$(R \cup 1)^k = 1 \cup (\bigcup_{1 \leq i \leq k} R^i)$$

$$(R \cup 1)^{k+1} = (R \cup 1)^k \circ (R \cup 1).$$

By induction hypothesis this is

$$(1 \cup (\bigcup_{1 \le i \le k} R^i)) \circ (R \cup 1)$$

Now $(A \cup B) \circ (C \cup D)$ is clearly $(A \circ C) \cup (A \circ D) \cup (B \circ C) \cup (B \circ D)$ and applying this here we get

$$(1 \circ R) \ \cup \ (1 \circ 1) \ \cup \ ((\bigcup_{1 \le i \le k} R^i) \circ R) \ \cup \ ((\bigcup_{1 \le i \le k} R^i) \circ 1)$$

Now $1 \circ R$ is R; $1 \circ 1$ is 1;

$$(\bigcup_{1 \le i \le k} R^i) \circ 1 \text{ is } \bigcup_{1 \le i \le k} R^i \text{ and}$$

$$(\bigcup_{1 \le i \le k} R^i) \circ R \text{ is } (\bigcup_{1 < i \le k+1} R^i)$$

so we get

$$R \cup 1 \cup (\bigcup_{1 < i \le k+1} R^i) \cup (\bigcup_{i \le k} R^i)$$

which is

$$1 \cup \bigcup_{1 \le i \le k+1} R^i$$

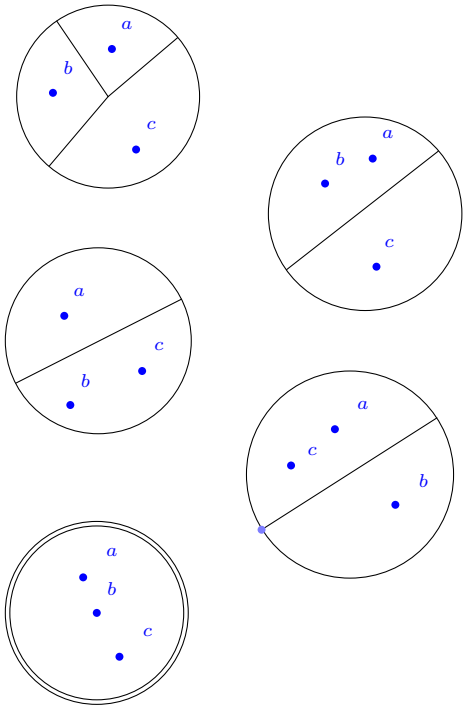
3. The transitive closure of the reflexive closure of R is the transitive closure of $R \cup 1$ which is $\bigcup_{n \in \mathbb{N}} (R \cup 1)^n$ which (as we have—more-or-less—just proved) is $1 \cup (\bigcup_{i \in \mathbb{N}} R^i)$ which is the reflexive closure of the transitive closure of R.

s is *increasing* so $R \subseteq s(R)$. t is *monotone*, so $t(R) \subseteq t(s(R))$. But the transitive closure of a symmetrical relation is symmetrical so $t(R) \subseteq t(s(R))$ implies $s(t(R)) \subseteq t(s(R))$ as desired.

Finally if $X = \mathbb{N}$ and $R = 1 \cup \{\langle x, y \rangle : y = px \text{ for some prime } p\}$ then $st(R)$ is the relation that holds between two numbers when they are identical or one is a multiple of the other, and $ts(R)$ is the universal relation $\mathbb{N} \times \mathbb{N}$.

Exercise 25

List all partitions of $\{a, b, c\}$. (You might find it helpful to draw a picture of each—a kind of Venn Diagram.)



Why the two concentric circles on the bottom left?

A: That partition has only one piece. The partition is a singleton!

Exercise 29

Look up 'monophyletic'. Using only the auxilliary relation "is descended from" give a definition in first-order logic of what is is for a monadic predicate of lifeforms to be monophyletic.

F is monophyletic iff both

$$(\forall xy)((F(x) \wedge F(y)) \rightarrow (\exists z)(F(z) \wedge D(z,x) \wedge D(z,y)))$$

and

$$(\forall x)(\forall y)(F(x) \rightarrow (D(x,y) \rightarrow F(y)))$$

hold.

On finite domains this is equivalent to $(\exists x)(\forall y)(D(x,y) \longleftrightarrow F(y))$ which we should probably accept.

You may many years later need the concept of a *directed subset*. X is a directed subset of a poset $\langle P, \leq \rangle$ if $(\forall x, y \in X)(\exists z \in X)(x \leq z \wedge y \leq z)$.

Exercise 31

The graph of the three-place order relation on the four positions on the face is
$\{\langle$XII, III, VI$\rangle$, $\langle$III, VI, IX$\rangle$, $\langle$VI, IX, XII$\rangle$, $\langle$IX, XII, III$\rangle$,
$\langle$XII, VI, IX$\rangle$, $\langle$III, IX, XII$\rangle$, $\langle$VI, XII, III$\rangle$, $\langle$IX, III, VI$\rangle$,
$\langle$XII, III, IX$\rangle$, $\langle$III, VI, XII$\rangle$, $\langle$VI, IX, III$\rangle$, $\langle$IX, XII, VI$\rangle\}$.

Exercise 35

They are equivalent. I shall deduce the second from the first. (The implication in the other direction is analogous.) Assume

label 'first' has been used twice

$$(\forall xyz)(z < x \not\leq y \not\leq x \rightarrow z < y) \tag{9.1}$$

and let a, b and c be such that $a > b \not\leq c \not\leq b$. We seek to infer $a > c$.

Suppose not. Then we have $a \not> c$, which is $\neg(a \geq c \wedge a \neq c)$ which is $a = c \vee a \not\geq c$. $c = a$ contradicts our assumption that $a > b$ but b incomparable with c, so we must have $a \not\geq c$. We can't have $c \geq a$ beco's $a > b$ and the conjunction would give us $c > b$ contradicting the assumption that $\{b, c\}$ are an incomparable pair. So $\{a, c\}$ are an incomparable pair. But now b is strictly below one member of an incomparable pair and so, by 9.1, must be below the other member of the pair—namely c. But $\{b, c\}$ are an incomparable pair. So our assumption must have been wrong, and we must have $a > c$ after all.

Exercise 39

(a) Is $R \setminus R^{-1}$ antisymmetric? Asymmetric?
(b) Is R `XOR` R^{-1} symmetrical? Antisymmetric? Asymmetric?
(c) Is the composition of two symmetrical relations symmetrical?
(d) Is the composition of two transitive relations transitive?
(e) Is the converse of a symmetrical relation symmetrical?
(f) Is the converse of a transitive relation transitive?

Answer

(b) is the only one that requires much thought. $(R$ `XOR` $S)^{-1}$ must be R^{-1} `XOR` S^{-1} so the converse of R `XOR` R^{-1} must be R^{-1} `XOR` R which is the same thing. So it's symmetrical.

(a) Asymmetric, so both; (c) Yes; (d) No; (e) Yes; (f) Yes.

Exercise 40

Can there be a function $f : X \to X$ whose graph is
(i) a reflexive relation? or
(ii) a transitive relation? or
(iii) a symmetrical relation?

Answers

(i) Obviously $\mathbb{1}_X$, the identity relation on X. And it's the only one!

(ii) This is precisely the condition for f to be idempotent, as on p. 72.

(iii) If it's symmetrical then it must be surjective. And it must be injective, since if it sends a and b both to c, then by symmetry it must send c to both a and b. So $a = b$. So it's a permutation of X, and a permutation of order 2. Such permutations are called *involutions*.

Exercise 41

This exercise is partly about brute calculation of certain quantities, but also an excuse to think a bit about natural bijections.

How many binary relations are there on a set of size n?

This is not a difficult question at all, but 99% of beginners get it wrong simply because they expect to be able to wing it, and they won't think it through.

The answer—of course—is 2^{n^2}.

How many of them are . . . ?	*Answer*
(a) reflexive?	2^{n^2-n}
(b) fuzzies?	$2^{\binom{n}{2}}$
(c) symmetrical?	$2^{\binom{n+1}{2}}$
(d) antisymmetric?	$2^n \cdot 3^{\binom{n}{2}}$
(e) total orders?	$n!$
(f) trichotomous?	$2^n \cdot 3^{\binom{n}{2}}$
(g) antisymmetric and trichotomous?	$2^{\binom{n+1}{2}}$
(h) extensional?	$\binom{2^n}{n}$
(i) partial orders	*Do not answer this question*
(j) strict partial orders	*Do not answer this question*
(k) permutations?	$n!$
(l) circular orders?	$(n-1)!$

Without actually calculating the answers to (c), (d), (f), (g), (j) or (i) . . .
(m) *Explain why are the answers to (d) and (f) are the same*
(n) Explain why are the answers to (g) and (c) are the same;
(o) 41 Explain why are the answers to (j) and (i) are the same.

Answer:

For (m) To get a bijection you need a choice function on the set of pairs of elements of the underlying set: take the matrix for a symmetrical relation, and flip all the bits in the upper right triangle. This is just `XOR`-ing with the graph of

a total order. Thus every total order naturally gives rise to a bijection between the set of symmetrical relations and the set of antisymmetrical trichotomous relations.

For (n). For both symmetrical relations and antisymmetrical trichotomous relations you have n independent choice of whether or not to put in the "diagonal" orderedd pairs $\langle x, x\rangle$. Now consider the pair $\langle x, y\rangle$ with $x \neq y$ and its mate $\langle y, x\rangle$. A symmetrical relation either has both these pairs or neither (two possibilities); an antisymmetrical trichotomous relation has precisely one of them (again, two possibilities). So, either way you have $\binom{n}{2}$ choices of two outcomes. So the two answers are the same. How are we to find a bijection between the set of symmetrical relatiions and gthe set of antisymmetrical trichotomous relations? It's pretty clear that there is no natural way of doing it. However, if we arm ourselves with a total order $<$ of our set we can do something. If R is antisymmetrical and trichotomous we want to obtain a symmetrical relation R' from it. Put into R' all (and only) the "diagonal" pairs in R. Then if R contains $\langle x, y\rangle$ where $x < y$, then put into R' *both* the pairs $\langle x, y\rangle$ and $\langle y, x\rangle$. If instead it contains $\langle y, x\rangle$ then put neither pair into R'. It should now be clear how to go in the opposite direction.

The answers to (k) and (e) are of course the same, as we all know. There doesn't seem to be any natural bijection between the set of all permutations of a set A (also known as the *symmetric group on* a set and notated $\Sigma(A)$) and the set of total orders of it. However, if we fix a total order of a set A, any other total order of A can be thought of as the application of a permutation of A to that fixed total order. So there is a natural bijection between the set of total orders of A and the set of bijections between $\Sigma(A)$ and the set of total orderings of A. I think the bijection induced by the permutation π is precisely the operation "conjugate with π".

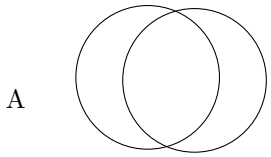
Expand on this; draw some pictures.

Ad (h); if R is an extensional relation on a set A then the function $a \mapsto \{a' : \langle a', a\rangle \in R\}$ is extensional, so the set of extensional relations on A is naturally bijected with the set of injective functions $A \hookrightarrow \mathcal{P}(A)$ and that is obviously of size $\binom{2^n}{n}$.

Edit this answer properly

Exercise 42

Let X be our set of pairs. Consider $\mathrm{dom}(X \cap \mathbb{1})$ (in full: $\{x : \langle x, x\rangle \in X\}$). If X is of the form $(A \times B) \cup (B \times A)$ then $\mathrm{dom}(X \cap \mathbb{1})$ is clearly going to be $A \cap B$. We can get $A \cup B$ because it is $\{x : (\exists y)(\langle x, y\rangle \in X)\}$. Then A `XOR` B can be obtained as $(A \cup B) \setminus (A \cap B)$. Then $A \setminus B$ and $B \setminus A$ are the two crescents in the following picture. How do we characterise them? This is the hard part. The two crescents are the two equivalence classes of an equivalence relation of index 2.



This is not obvious! When do x_1 and x_2 belong to the same crescent? When they both belong to A (but not to B) or both belong to B (but not to A). Notice that in those circumstances the pair $\langle x_1, x_2\rangle \notin X$! The relation $x_1 \sim x_2$ defined by $\langle x_1, x_2\rangle \notin X$ is an equivalence relation on A XOR B!!

Exercise 43

The enemy of my enemy is my friend
The friend of my enemy is my enemy
The enemy of my friend is my enemy.
The friend of my friend is my friend.

You might like to express these observations in first-order logic, using binary relation symbols like $F(\ ,\ )$ and $E(\ ,\ )$.

Answer:

$(\forall xyz)(E(x,y) \wedge E(y,z) \rightarrow F(x,z))$
$(\forall xyz)(E(x,y) \wedge F(y,z) \rightarrow E(x,z))$
$(\forall xyz)(F(x,y) \wedge E(y,z) \rightarrow E(x,z))$
$(\forall xyz)(F(x,y) \wedge F(y,z) \rightarrow F(x,z))$

1. *If you have an enemy must you have a friend? If you have a friend are you friends with yourself?*

   We aren't actually told that $E(\ ,\ )$ and $F(\ ,\ )$ are commutative but most readers will probably assume they are. If they are, then if i have (even one) enemy then there is an enemy of that enemy who is my friend. I am one such friend. The second half is similar.

2. *Can you infer from the foregoing that two things cannot be simultaneously friends and enemies? Prove or find a countermodel.*

   Not really. That picture is quite consistent with there being lots of people all of whom are simultaneously friends and enemies of all the others—and even themselves. We are missing the two assumptions of commutativity and the irreflexivity of $E(\ ,\ )$

3. *Explain "congruence relation for ...." Assume 'friend-of' to be reflexive, so it is an equivalence relation. Think about the equivalence-classes-under-friendship. Let's also assume that—the expression "he's his own worst enemy" notwithstanding—'enemy-of' is irreflexive.*

(i) How does 'enemy-of' "lift" to these equivalence classes? Is 'friend-of' a congruence relation for 'enemy-of'?

'Congruence relation for' is bookwork. And yes, if $F(\ ,\ )$ is an equivalence relation then it is certainly a congruence relation for $E(\ ,\ )$.

(ii) *How many equivalence classes can an equivalence class be hostile to?*

Only one. If you are at war with two alliances then, because the enemy of your enemy is your friend, the two alliances would coalesce into one.

(iii) Explain how your answer to (ii) partitions the domain. Need an answer to (iii)

4. *Clearly 'enemy-of' is not transitive, but it does have a property that is rather like transitivity. Can you describe this feature exactly, and state it for a binary relation R in the style in which you know how to state that R is transitive?*

   Answer:

   $R^3 \subseteq R$. There is no word standardly used to describe this property.

5. *Does the feature (analogous to transitivity) from the previous part admit a notion of closure analogous to transitive closure, symmetric closure etc.? Give a proof or a counterexample.*

   Answer

   Yes. Whenever you find the pairs $\langle x, y\rangle$, $\langle y, z\rangle$ and $\langle z, w\rangle$ in R, add the pair $\langle x, w\rangle$. Keep doing this until you can add no more pairs. The thingie-closure of R is $\bigcap\{S \supseteq R : S^3 \subseteq S\}$ by analogy with definition of transitive closure etc.

Exercise 44

Look again at the box of isotopes on page 70.

Why was I right to write the half-life of Pb^{208} as '∞' rather than '$\aleph_0$'?"

Answer

There is no number α such that after α years half the atoms in a sample of Pb^{208} have decayed. Indeed the function *half-life-of* is defined only for radioactive species, and Pb^{208} is not radioactive: the half life of Pb^{208} is not defined. In particular it isn't $\aleph_0$. What symbol do we use for undefined quantities that seem to be infinite, such as '1/0'? Yes, '∞'. Thus you sometimes see people writing '$t < \infty$' where 't' is a term (sometimes undefined) of type `real` or `natural number` to mean that t is defined.

Exercise 53

[*statement of exercise omitted to save space*]

Discussion

This is a beautiful question, co's it touches several important points. It tests your understanding of structural induction; it tests your ability to do the fiddly manipulation necessary to perform the inductive step; it underlines the importance of having a sufficiently strong induction hypothesis, and finally it makes a point about dereferencing.

So: we have a propositional language—a recursive datatype of formulæ—which starts off with three propositional letters ("literals") 'a', '$\top$' and '$\bot$'. We then build up compound formulæ by means of the constructors '$\wedge$', '$\vee$' and '$\neg$'. We have a *length* function defined on objects in the datatype of formulæ, written with two vertical bars as in the question, which is roughly what you think it is—so that the length of a literal is 1, and the length of a conjunction (or a disjunction) of two formulæ is one plus the sum of their lengths, and the length of the negation of a formula is one plus the length of the formula. Evidently the question-designer thought that the length of a '(' or a ')' is zero!

One tends naturally to write the second half of the preceding paragraph with expressions like

$$|A \wedge B| = |A| + |B| + 1.$$

This looks fair enough, and in some sense it is, but we need to be clear about the conventions we are using. The letter 'A' by itself is a single symbol, so a pedant might insist that $|A| = 1$. This is wrong of course: the letter 'A' is not a formula, but a variable ranging over formulæ... when looking for the length $|A|$ of A we have to *see through*[1] the variable all the way to the value it takes—and that value is a formula. All this is well and good, but it can cause some confusion when we start thinking about expressions like: $|A \vee B|$. The constructor '$\vee$' is something we put between two formulæ to make a new formula; we don't put it between two *names* of formulæ or between two *pointers* to formulæ! Until we have a convention to make our practice OK, writing things like '$|A \vee B|$' should generate a `syntax error` warning. If you look back to page 96 where this exercise first appears you will find that I wrote

> "...length of a literal is 1, and the length of a conjunction (or a disjunction) of two formulæ is one plus the sum of their lengths..."

...and this is syntactically correct. When we wrote '$|A \wedge B|$' we should really have written '| the conjunction of A and B|'.

There are two ways of dealing with this. One is to have explicit names for the constructors, as it might be 'conjunction of ...' and 'disjunction of ...' and 'negation of ...' This makes huge demands on our supply of alphanumerics. The other solution is to have a kind of `environment` command that creates an environment within which [deep breath]

[1] I have italicised this word because the metaphor is a good one: google *referential transparency*.

Contructors applied to **pointers***-to-objects*
construct
pointers *to the objects thereby constructed.*

Inside such a context things like '$|A \vee B|$' have the meaning we intend here. There is a culture within which this environment is created by the '$\ulcorner$' symbol (LaTeX: `\ulcorner`) and closed by the '$\urcorner$' symbol (LaTeX: `\urcorner`). In practice people tend to leave these things out. The fact that this is—apparently—a safe strategy tells us quite a lot about the skills of our language module: it's very good at dereferencing (among other things)

Thus we should/should-have posed the question as:

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

$|a| = 1$,
$|\top| = 1$,
$|\bot| = 1$,
$|\ulcorner A \wedge B| = |A| + |B|\urcorner + 1$,
$|\ulcorner A \vee B| = |A| + |B|\urcorner + 1$,
$|\ulcorner \neg A| = |A|\urcorner + 1$."

[or *something* like that, with the corners placed correctly!]

And the remainder of the question:

"Define a translation which eliminates disjunction from Boolean expressions by the following recursion:

$tr(a) = a$, $tr(\top) = \top$, $tr(\bot) = \bot$,
$\ulcorner tr(A \wedge B) = tr(A) \wedge tr(B)$,
$tr(A \vee B) = \neg(\neg tr(A) \wedge \neg tr(B))$,
$tr(\neg A) = \neg tr(A)\urcorner$.

Prove by structural induction on Boolean propositions that

$$|\ulcorner tr(A)| \leq 3|A| - 1\urcorner,$$

for all Boolean propositions A."

The above use of corner quotes illustrates how there is no restriction that says that the scope of the corner quotes has to live entirely inside a single formula. I use corner quotes in what follows, but (although—i *think*—i have put them in correctly) they can be inserted correctly in more than one way.

The Proof by Structural Induction

We aspire to prove by structural induction on the recursive datatype of formulæ that

$$(\forall A)(|tr(A)| \leq 3 \cdot |A| - 1)$$

The base case we verify easily. The induction step has three cases

$\neg$ If $|tr(A)| \leq 3 \cdot |A|$ what is $|\ulcorner tr(\neg A)\urcorner|$? $\ulcorner tr(\neg A) = \neg tr(A)\urcorner$ so $\ulcorner |tr(\neg A)| = |\neg tr(A)|\urcorner$, and $|\ulcorner \neg tr(A)\urcorner|$ is $|tr(A)| + 1$ which is certainly $\leq 3 \cdot |\ulcorner \neg A\urcorner|$.

$\wedge$ If $|tr(A)| \leq 3 \cdot |A|$ and $|tr(B)| \leq 3 \cdot |B|$ what is $|\ulcorner tr(A \wedge B)\urcorner|$?

$\ulcorner tr(A \wedge B)\urcorner$ is $\ulcorner tr(A) \wedge tr(B)\urcorner$. By induction hypothesis $|tr(A)| \leq 3 \cdot |A| - 1$ and $|tr(B)| \leq 3 \cdot |B| - 1$ so $\ulcorner |tr(A) \wedge tr(B)|\urcorner \leq (3 \cdot |A| - 1) + (3 \cdot |B| - 1) + 1$. The final '+1' is for the '$\wedge$'. This rearranges to

$|\ulcorner tr(A) \wedge tr(B)\urcorner| \leq 3 \cdot (|A| + |B|) - 1$

but $|A| + |B| \leq |\ulcorner A \wedge B\urcorner|$ whence

$\ulcorner |tr(A) \wedge tr(B)| \leq 3 \cdot (|A \wedge B|) - 1\urcorner$ and finally

$\ulcorner |tr(A \wedge B)| \leq 3 \cdot (|A \wedge B|) - 1\urcorner$.

$\vee$ If $|tr(A)| \leq 3 \cdot |A|$ and $|tr(B)| \leq 3 \cdot |B|$ what is $|tr(A \vee B)|$? $\ulcorner tr(A \vee B)\urcorner$ is $\ulcorner \neg(\neg tr(A) \wedge \neg(tr(B)))\urcorner$. What is the length of this last expression? Clearly it's going to be $|tr(A)| + |tr(B)|$ + one for the outermost '$\neg$' + one for the '$\neg$' attached to $tr(A)$ + one for the '$\neg$' attached to $tr(B)$ + one for the '$\wedge$' ... giving $|tr(A)| + |tr(B)| + 4$. By induction hypothesis $|tr(A)| \leq 3 \cdot |A| - 1$ and $|tr(B)| \leq 3 \cdot |B| - 1$ so we have

$\ulcorner |tr(A \vee B)| \leq (3 \cdot |A| - 1) + (3 \cdot |B| - 1) + 4\urcorner$. We can rearrange this to

$\ulcorner |tr(A \vee B)| \leq 3 \cdot (|A| + |B|) - 1 - 1) + 4\urcorner$ and further to

$\ulcorner |tr(A \vee B)| \leq 3 \cdot (|A| + |B|) + 2\urcorner$.

Now $|A| + |B| = \ulcorner |A \vee B|\urcorner - 1$ so we can substitute getting

$\ulcorner |tr(A \vee B)| \leq 3 \cdot (|A \vee B| - 1)) + 2\urcorner$ and rearrange again to get

$\ulcorner |tr(A \vee B)| \leq 3 \cdot |A \vee B| - 1\urcorner$ as desired.

A final thought ... I wouldn't mind betting that quite a lot of thought went into this question. We've proved $|tr(A)| \leq 3 \cdot |A| - 1$ so we've certainly also proved the weaker claim $|tr(A)| \leq 3 \cdot |A|$. However wouldn't stake my life on our ability to prove the weaker claim by induction. You might like to try ... i'm not going to!

Exercise 57

Show that:

For all valuations v and all formulæ A and all but finitely many t,

$$E_l(A, v, t) = E(A, v) \text{ and } E_s(A, v, t) = E(A, v).$$

In the classical setting one takes a propositional valuation—that is to say, a [total!] function atoms → `:bool`—and "extends" it ("by abuse of notation") to a function defined on complex expressions. This is all right because—if the valuation is total—one can safely conceal the evaluation process. *If the valuation is total then the process of evaluation is confluent.* If the functions are partial then one needs to make explicit the recursion that takes a time t and returns the estimate-at-time-t of the truth-value.

Let v be a valuation, a function with values in `:bool` defined only on atomics. We define

$E_s(A, v, 0) = E_l(A, v, 0) = v(A)$

and thereafter

$$\begin{aligned} E_s((A \wedge B), v, t+1) &= E_s((A), t) \wedge E_s(B, v, t) \\ E_s((A \vee B), v, t+1) &= E_s((A), v, t) \vee E_s(B, v, t) \\ E_s((\neg A), v, t+1) &= \neg E_s(A, t) \end{aligned}$$

where the connectives in the definiens are interpreted strictly, more explicitly:

$$\begin{aligned} E_s((A \wedge B), v, t+1) = {} & \texttt{if } E_s(A, v, t) = \texttt{false then if } E_s(B, v, t) = \texttt{false then false else} \\ & \texttt{if } E_s(A, v, t) = \texttt{false then if } E_s(B, v, t) = \texttt{true then false else} \\ & \texttt{if } E_s(A, v, t) = \texttt{true then if } E_s(B, v, t) = \texttt{false then false else} \\ & \texttt{if } E_s(A, v, t) = \texttt{true then if } E_s(B, v, t) = \texttt{true then true else undefined} \end{aligned}$$

$$\begin{aligned} E_s((A \vee B), v, t+1) = {} & \texttt{if } E_s(A, v, t) = \texttt{true then if } E_s(B, v, t) = \texttt{false then true else} \\ & \texttt{if } E_s(A, v, t) = \texttt{true then if } E_s(B, v, t) = \texttt{true then true else} \\ & \texttt{if } E_s(A, v, t) = \texttt{false then if } E_s(B, v, t) = \texttt{false then false else} \\ & \texttt{if } E_s(A, v, t) = \texttt{false then if } E_s(B, v, t) = \texttt{true then true else undefined} \end{aligned}$$

In contrast, when the functions are partial and we evaluate lazily

$$\begin{aligned} E_l((A \wedge B), v, t+1) = {} & \texttt{if } E_l(A, v, t) = \texttt{false then false} \quad \texttt{else} \\ & \texttt{if } E_l(B, v, t) = \texttt{false then false} \quad \texttt{else} \\ & \texttt{if } E_l(A, v, t) = \texttt{true then } E_l(B, v, t) \texttt{ else} \\ & \texttt{if } E_l(B, v, t) = \texttt{true then } E_l(A, v, t) \texttt{ else} \\ & E_l((A \wedge B), v, t) \end{aligned}$$

$$\begin{aligned} E_l((A \vee B), v, t+1) = {} & \texttt{if } E_l(A, v, t) = \texttt{true then true} \quad \texttt{else} \\ & \texttt{if } E_l(B, v, t) = \texttt{true then true} \quad \texttt{else} \\ & \texttt{if } E_l(A, v, t) = \texttt{false then } E_l(B, v, t) \texttt{ else} \\ & \texttt{if } E_l(B, v, t) = \texttt{false then } E_l(A, v, t) \texttt{ else} \\ & E_l((A \vee B), v, t) \end{aligned}$$

$$\begin{aligned} E_l((\neg A), v, t+1) &= \texttt{if } E_l(A, v, t) = \texttt{true then false} \quad \texttt{else} \\ &= \texttt{if } E_l(A, v, t) = \texttt{false then true} \quad \texttt{else} \\ & \quad E_l((\neg A), v, t) \end{aligned}$$

Exercise 58

A wellordering of a set of X is a wellfounded relation, whatever else it is. In fact its (graph is) a $\subseteq$-maximal wellfounded relation on X. Is that a sufficient condition? Probably need transitivity too.

We aspire to prove that any $\subseteq$-maximal wellfounded relation is a wellordering. We prove two factoids:

(i) a $\subseteq$-maximal wellfounded relation will be transitive.

(ii) a $\subseteq$-maximal wellfounded relation will be trichotomous.

For (i) recall that the transitive closure of a wellfounded relation is wellfounded, so a $\subseteq$-maximal wellfounded relation must be equal to its transitive closure.

For (ii) suppose R is maximal wellfounded, with $\langle a, b\rangle \notin R$ (where $a \neq b$) so that $R \cup \{\langle a, b\rangle\}$ is not wellfounded. That means there is some subset $X' \subseteq X$ s.t. X' has an R-minimal element but no minimal element according to $R \cup \{\langle a, b\rangle\}$. This must mean that b was that R-minimal element and that $a \in X'$. But that means that we could add the pair $\langle b, a\rangle$ to R without violating wellfoundedness and *that* means—by maximality of R—that $\langle b, a\rangle$ was already in R.

This means that the following are equivalent, for $R \subseteq X \times X$:

(i) R is a wellordering of X
(ii) R is maximal wellfounded
(iii) R is wellfounded and trichotomous.

Exercise 60

Part 2.

The lexicographic order on $\mathbb{N}^2$ is wellfounded, so we can do wellfounded induction on it. This means that if we can prove that, if every ordered pair below p has some property ϕ then the pair p has property ϕ as well, then every ordered pair in $\mathbb{N}^2$ has that property.

Now let $\phi(\langle x, y\rangle)$ say that if the bag is started with x black balls and y white balls in it the process will eventually halt with only one ball in the bag. Suppose $\phi(\langle x', y'\rangle)$ holds for every $\langle x', y'\rangle$ below $\langle x, y\rangle$ in the lexicographic product $\mathbb{N}^2$. We want to be sure that if the bag is started with x black balls and y white balls in it the process will eventually halt with only one ball in the bag. The first thing that happens is that we pick two balls out of the bag and the result is that at the next stage we have either $x - 2$ black balls and an unknown number of white balls, or we have x black balls and $y - 1$ white balls. But both these situations are described by ordered pairs below $\langle x, y\rangle$ in the lexicographic product $\mathbb{N}^2$, so by induction hypothesis we infer that if the bag is started with x black balls and y white balls in it the process will eventually halt with only one ball in the bag, as desired.

Exercise 68

Suppose we are given n_1 and n_2, both of them numbers of the form $2^k - 3m$. (This is simply to say that neither n_1 nor n_2 is divisible by 3). We wish to obtain n_2 from n_1 by repeatedly doubling and subtracting 3.

If the two numbers have the same residue mod 3 and $n_1 > n_2$ then subtract 3 from n_1 the appropriate number of times. If $n_1 < n_2$ or they have different residues mod 3 then procede along the sequence $2 \cdot n_1$, $4 \cdot n_1$, $8 \cdot n_1$ until you reach a multiple which is both bigger than n_2 and congruent to n_2 mod 3. Then subtract 3 the correct number of times.

Exercise 69

We (the match referee) start with a matrix, and delete rows or columns when dominance allows us to, and we continue until no more deletions are possible. It might be that at some stage(s) there is more than one row (or column) that can be deleted, or perhaps a stage at which both a row and a column are delete-able. Is the outcome affected by the order in which we perform these deletions?

One column dominates a second column iff at every row the entry in the first column is at least as big as the entry in the second column. (And of course the same goes for two rows *mutatis mutandis* ...). This means that if one column dominates another it will continue to do so even if some rows are deleted. This means that if a chance ever arises to delete a particular column then *that chance remains on the table whatever else we do.* So we can postpone any deletion for as long as we like. (This is helpful because of course there are occasions where a chance to delete a particular column cannot arise until a particular row has been deleted.) This means that, should we ever reach a stage where no more deletions can be performed, this can only be because all deletions that ever became possible have been performed. And that uniquely characterises our destination.[2]

It is the fact that deletion-possibilities remain permanently open that ensures confluence. Your sequence-of-deletion strategies might diverge, but the permanent-possibility feature ensure that all such strategies can rejoin.

But wait! How can we be sure that a deletion possibility that arises down one path will also arise down any other? This isn't a problem: think about the possibilities in front of your eyes at the stage where the two paths part company. [should probably say a bit more about this]

What happens if your matrix has infinitely many rows and columns? In those circumstances you have the possibility that your dependency relation between deletions ("I can't delete column c until i have deleted row r; I can't delete row r until I have deleted column c'; I can't delete column c' until I have deleted row r'; I can't delete row r' until I have deleted column c'' ...") might not be

[2]But we should probably Say something about why any given deletion always has the same effect (whatever that means) irrespective of when we do it. Can we find a situation where this doesn't hold? It would be useful

wellfounded. Can you cook up an infinite matrix where—for all i—you can't delete row i until you have deleted column i and can't delete column i until you have deleted row $i+1$?

Question 8.2

Let R be a relation on A. Recall that $r(R)$, $s(R)$ and $t(R)$ are the reflexive, symmetric and transitive closure operations respectively.

(a) Prove that $rs(R) = sr(R)$;
(b) Does R transitive imply $s(R)$ transitive?
(c) Prove that $rt(R) = tr(R)$ and $st(R) \subseteq ts(R)$;
(d) If R is symmetrical must the transitive closure of R be symmetrical? Prove or give a counterexample.

Think of a binary relation R, and of its graph, which will be a directed graph $\langle V, E\rangle$. On any directed graph we can define a relation "I can get from vertex x to vertex y by following directed edges" which is certainly transitive, and we can pretend it is reflexive because after all we can get from a vertex to itself by just doing nothing at all. Do this to our graph $\langle V, E\rangle$, and call the resulting relation S. How do we describe S in terms of R?

Answer

(a) Prove that $rs(R) = sr(R)$:

$$\begin{aligned} r(s(R)) &= s(R) \cup 1 \\ &= (R \cup R^{-1}) \cup 1 \\ &= (R \cup I) \cup (R^{-1} \cup 1) \\ &= (R \cup I) \cup (R^{-1} \cup 1^{-1}) \\ &= (R \cup 1) \cup (R \cup I)^{-1} \\ &= s(r(R)) \end{aligned}$$

(b) The symmetric closure of a transitive relation is not automatically transitive: take R to be set inclusion on a power set.

(c) Prove that $rt(R) = tr(R)$:

$$\begin{aligned} r(t(R)) &= t(R) \cup I \\ &= R \cup R^2 \cup \ldots R^n \ldots \cup I \\ &= (R \cup I) \cup (R^2 \cup I) \cup (R^n \cup I) \ldots \end{aligned}$$

At this point it would be nice to be able to say $(R^n \cup I) = (R \cup I)^n$ but that isn't true. $(R \cup I)^n$ is actually $R \cup R^2 \ldots R^n \cup I$. But it is enough to rewrite the last line as

$$(R \cup I) \cup (R \cup I)^2 \cup (R \cup I)^3 \ldots$$

which is of course $t(r(R))$ as desired.

(d) The transitive closure of a symmetrical relation is also symmetrical. First we show by induction on n that R^n is symmetrical as long as R is. Easy

when $n = 1$. Suppose R^n is symmetrical. $R^{n+1} = R \circ R^n$. The inverse of this is $(R^{-1})^n \circ R^{-1}$ which by induction hypothesis is $R^n \circ R$ which is of course R^{n+1}. Then the union of a lot of symmetrical relations is symmetrical, so the transitive closure (which is the union of all the (symmetrical) iterates of R) is likewise symetrical.

Question 8.4

> *Show that—at least if* $(\forall x)(\exists y)(\langle x, y\rangle \in R)$*—*$R \circ R^{-1}$ *is a fuzzy. What about* $R \cap R^{-1}$*? What about* $R \cup R^{-1}$*?*

If $\langle x, y\rangle \in R$ then $\langle y, x\rangle \in R^{-1}$ so $\langle x, x\rangle \in R \circ R^{-1}$.

That takes care of reflexivity. Suppose $\langle x, z\rangle \in R \circ R^{-1}$. Then there is a y such that $\langle x, y\rangle \in R$ and $\langle y, z\rangle \in R^{-1}$. But then $\langle z, y\rangle \in R$. So $\langle x, z\rangle \in R \circ R^{-1}$ is the same as $(\exists y)(\langle x, y\rangle \in R \ \wedge \ \langle z, y\rangle \in R)$. But this is clearly symmetric in x and z, so we can rearrange it to get $(\exists y)((\langle y, x\rangle \in R^{-1}) \wedge (\langle z, y\rangle \in R))$ which is $\langle z, x\rangle \in R \circ R^{-1}$ as desired.

$R \cup R^{-1}$ is the symmetric closure of R and is of course symmetric, but there is no reason to expect it to be reflexive: it'll be reflexive iff R is reflexive.

Question 8.5

> *"Given any relation* R *there is a least* $T \supseteq R$ *such that* T *is transitive, and a least* $S \supseteq R$ *such that* S *is symmetrical, namely the transitive and symmetric closures of* R*. Must there also be a maximal* $S \subseteq R$ *such that* S *is transitive? And must there be a maximal* $S \subseteq R$ *such that* S *is symmetrical? The answer to one of these last two questions is 'yes': find a cute formulation."*

$R \cap R^{-1}$ is the largest symmetrical relation included in R. The unwary sometimes think that it is *this* that is the symmetric closure of R. The point is that altho' being-the-complement-of-a-transitive-relation is not an intersection-closed property, nevertheless being-the-complement-of-a-symmetric-relation **is** intersection-closed, since it is the same as being symmetric. $R \cap R^{-1}$ is the complement of the symmetric closure of the complement of R. Do not confuse complements with converses!!

Question 8.9

Show that $R \subseteq S$ implies $R^{-1} \subseteq S^{-1}$

The way to do this is to assume that $R \subseteq S$ and let $\langle x, y\rangle$ be an arbitrary ordered pair in R^{-1}. We then want to infer that $\langle x, y\rangle$ is in S^{-1}.

If $\langle x, y\rangle$ is in R^{-1} then $\langle y, x\rangle$ is in R, because R^{-1} is precisely the set of ordered pairs $\langle x, y\rangle$ such that $\langle y, x\rangle$ is in R. (We would write this formally as: $R^{-1} = \{\langle x, y\rangle : \langle y, x\rangle \in R\}$.) But $R \subseteq S$, so $\langle y, x\rangle$ is in S, and so (flip things round again) $\langle x, y\rangle$ is in S^{-1}.

Question 8.13

Show that $\bigcup_{n\in\mathbb{N}} R^n$ is the smallest transitive relation extending R.

To do this it will be sufficient to show

1. $\bigcup_{n\in\mathbb{N}} R^n$ is transitive;
2. If S is a transitive relation $\supset R$ then $\bigcup_{n\in\mathbb{N}} R^n \subseteq S$.

(1)
We need to show that if $\langle x, y\rangle$ and $\langle y, z\rangle$ are both in $\bigcup_{n\in\mathbb{N}} R^n$ then $\langle x, z\rangle \in \bigcup_{n\in\mathbb{N}} R^n$.

If $\langle x, y\rangle \in \bigcup_{n\in\mathbb{N}} R^n$ then $\langle x, y\rangle \in R^k$ for some k and if $\langle y, z\rangle \in \bigcup_{n\in\mathbb{N}} R^n$ then $\langle y, z\rangle \in R^j$ for some j.
Then $\langle x, z\rangle \in R^{j+k} \subseteq \bigcup_{n\in\mathbb{N}} R^n$.

For (2) Let $S \supset R$ be a transitive relation. So $R \subseteq S$. We prove by induction on $\mathbb{N}$ that for all $n \in \mathbb{N}$, $R^n \subseteq S$. Suppose $R^n \subseteq S$. Then

$$R^{n+1} = R^n \circ R \ \subseteq^{(a)} \ S \circ R \ \subseteq^{(b)} \ S \circ S \ \subseteq^{(c)} \ S.$$

(a) and (b) hold because $\circ$ is *monotone*: if $X \subseteq Y$ then $X \circ Z \subseteq Y \circ Z$.
(c) holds because S is transitive.)

Question 8.16

Let R and S be equivalence relations. We seek the smallest equivalence relation that is a superset of $R \cup S$. We'd better note first that this really is well defined, and it is, because being-an-equivalence-relation is the conjunction of three properties all of them intersection-closed, so it is itself intersection-closed.

This least equivalence relation extending $R \cup S$ is at least transitive, so it must be a superset of $t(R \cup S)$, the transitive closure of $R \cup S$. Wouldn't it be nice if it actually *were* $t(R \cup S)$? In fact it is, and to show this it will be sufficient to show that $t(R \cup S)$ is an equivalence relation. Must check: transitivity, reflexivity and symmetry. Naturally $t(R \cup S)$ is transitive by construction. R and S are reflexive so $R \cup S$ is reflexive. In constructing the transitive closure we add new ordered pairs but we never add ordered pairs with components we haven't seen before. This means that we never have to add any ordered pairs $\langle x, x\rangle$ because they're all already there. Therefore $t(R \cup S)$ is reflexive as long as R and S are. Finally we need to check that $t(R \cup S)$ is symmetrical. The transitive closure of a symmetrical relation is also symmetrical. First we show by induction on n that R^n is symmetrical as long as R is. Easy when $n = 1$. Suppose R^n is symmetrical: i.e., $R^n = (R^n)^{-1}$. $R^{n+1} = R \circ R^n$ anyway. The inverse of this

is $(R^{-1})^n \circ R^{-1}$. $(R^{-1})^n$ is of course R^{-n}, so $(R^{-1})^n \circ R^{-1}$ is $(R^{-n}) \circ R^{-1}$. $R^{-n} = R^n$ by induction hypothesis so $(R^{-1})^n \circ R^{-1}$ is $R^n \circ R$ which is of course R^{n+1}. Then the union of a lot of symmetrical relations is symmetrical, so the transitive closure (which is the union of all the (symmetrical) iterates of R) is likewise symmetrical.

Actually we can give another—perhaps simpler—proof of this. $t(R) = \bigcap\{S : R \subseteq S \wedge S^2 \subseteq S\}$, or $\bigcap X$ for short. Notice that if R is symmetrical, then X is closed under taking inverses (the inverse of anything in X is also in X). And clearly the intersection of a class closed under taking inverses is symmetrical.

Question 8.35

Are the two following conditions on partial orders equivalent?

1. $(\forall xyz)(z > x \not\leq y \not\leq x \rightarrow z < y)$
2. $(\forall xyz)(z > x \not\leq y \not\leq x \rightarrow z > y)$.

Assume (1) $(\forall xyz)(z \leq x \not\leq y \not\leq x \rightarrow z \leq y)$ and aim to deduce (2) $(\forall xyz)(z \geq x \not\leq y \not\leq x \rightarrow z \geq y)$. To this end assume $z \geq x$, $x \not\leq y$ and $y \not\leq x$ and hope to deduce $z \geq y$.

$x \leq z$ tells us that $z \not\leq y$ for otherwise $x \leq y$ by transitivity, contradicting hypothesis. Next, assume the negation of what we are trying to prove. This gives us $y \not\leq z$. But then we have $y \not\leq z \not\leq y$ and $x \leq z$ so by (i) we can infer $x \leq y$, contradicting assumption.

The proof in the other direction is analogous.

Question 8.36

If $x \leq S(y)$ and $y \leq S(x)$ then x and y are neighbouring naturals. This is $R \cap R^{-1}$. x and y are related by the transitive closure of this relation iff there is a finite sequence $x_0, x_1, x_2 \ldots x_n = y$ such that each x_i is adjacent to x_{i+1}. But clearly *any* two naturals are connected by such a chain, so the transitive closure is the universal relation. For part 2, remember that x is related to y by $R \setminus R^{-1}$ if it is related to y by R but not by R^{-1}. In this case that means $x \leq S(y) \wedge y \not\leq S(x)$. This is $x \leq S(y) \wedge S(x) < y$. The second conjunct implies the first so we can drop the first, getting $S(x) < y$. Getting the transitive closure of this is easy, 'cos it's transitive already!

Question 8.??

Everybody loves my baby, so in particular my baby loves my baby. **My baby loves nobody but me.** That is to say, if x is loved by my baby, then $x =$ me. So my baby = me.

Question ??.14a

The answer is the relation that holds between k and $k+1$ for $0 \leq k < n$ and between n and 0.

Question ??.24

```
- fun f g b a = g a b;
val f = fn : ('a -> 'b -> 'c) -> 'b -> 'a -> 'c
- fun ff g = let fun fa a = let val (b,c) = g a in b end;
= fun fe a = let val (b,c) = g a in c end;
= in (fa, fe) end;
val ff = fn : ('a -> 'b * 'c) -> ('a -> 'b) * ('a -> 'c)
-
```

Question ??.23

Show that the largest and smallest elements of a totally ordered set with n elements can be found with $[3n/2] - 1$ comparisons if n is odd, and $3n/2 - 2$ comparisons if n is even.

First check this for a few small values. If $n = 2$ we need 1, for $n = 3$ we need 3, for $n = 4$ we need 4.

The induction step requires us to show that adding two more elements to a set requires us to perform no more than three extra comparisons.

So suppose we have a set X with n members, and we have found the top and bottom elements in $3n/2 - 1$ comparisons. Call them t and b. Let the two new elements be x and y. With one comparison we can find out which is bigger. Without loss of generality suppose it is x. Compare x with t to find the biggest element of $X \cup \{x, y\}$, and compare y with b to find the smallest. This has used three extra comparisons.

Question ??.32

Let $m = |F|$ and $p = |\bigcup F|$. Let $C = \{\langle x, A\rangle : x \in A \in F\}$.

Given $x \in \bigcup F$, pick $B \in F$ with $x \in B$. Let $Y_x = \{A \in F : x \in A\}$ and $N_x = \{A \in F : x \notin A\}$. The map $\lambda A.(A \Delta B)$ permutes F and swaps Y_x and N_x. Hence $|Y_x| = |N_x| = m/2$.

So $|C| = (1/2)mp$, as each x is in exactly $m/2$ A's. But each A contains $\leq k$ things, and one A contains none at all, so $|C| \leq (m-1)k$ whence $p \leq \frac{m-1}{m} \cdot 2k < k$.

Chapter 10

Indexes

10.1 Index of Definitions

10.2 General Index

Bibliography

[1] Jorge Luis Borges "Siete Noches" English translation by Eliot Weinberger New directions publishing 1982 ISBN-13: 978-0811218382

[2] Lewis Carroll "Through the Looking-Glass and what Alice found there"

[3] Martin Gardner "Squaring the square" Scientific American 1958.

[4] Martin Gardner "The Annotated Alice"

[5] Conway, Berlekamp and Guy, "Winning Ways", Academic Press 1982

[6] Dörrie, H. "100 Great Problems of Elementary Mathematics" tr. by David Antin. Dover

[7] Girard, Lafont and Taylor "Proofs and Types" Cambridge University Press (1989)

[8] Guillermo Owen: "Game theory"

[9] Quine, W.V. Mathematical Logic (revised edition) Harper torchbooks 1962

[10] Oliver Sacks "The man who mistook his wife for a hat" Picador 1985