searching for logic

Adam Morton

(November 2019 version)

copyright information

This material is circulated under a creative commons copyright. You may use it without charge and recirculate it with changes, but you may not make money with it or claim it as your own work. Moreover you should acknowledge your use of it. For more details of what you are committed to, in more careful language, by making use of this material, see the full contract after the main text.

contents

preface 5

Part one — search

Chapter 1: representing information 9

chapter 2: queries 47
appendix: mental arithmetic as basic mathematical thinking 94

chapter 3: Boolean search 97

part two – propositional logic

chapter 4: searching for models 137

chapter 5: truth assignments 165
appendix: defining propositions 191

chapter 6: logical consequence 209

chapter 7: derivations 243

chapter 8: direct and indirect arguments 277

part three – quantifiers

Chapter 9: quantifiers and variables 307

chapter 10: multiple quantification 341

Chapter 11: logical consequence in quantifier logic 389
appendix: a system of derivations for quantifier logic 423

index 433

planning a course using this book 437

copyright issues 442

preface

To students: my hope is that this book will help you to think thoughts that were not available to you before, and that it will be useful in practical activities of finding information and then expressing it clearly. It aims to influence both your grasp of language and your approach to searching and arguing.

To instructors: my hope is that this book will help you give a course where the students are interested in the content for its own sake and find it relevant to their lives and studies. They should comment freely and have their own ideas about some of the content. There are suggestions about planning and delivering a course using this material at the end of the book. I hope to post improved and corrected versions on the sites where I am posting this. So comments and suggestions are welcome. Send them to adam.morton@ubc.ca.

This text is based on a course I gave many times at the university of Alberta and the University of British Columbia. It was an unusual course, meant to show philosophy majors, and other humanities students whose main interest is not logic, that formal logic connects with issues they find interesting. The aim was to give a course which covers the standard symbolic logic topics but

— The wider focus is not on deduction but on search in databases. ("Database" in computer science is very closely related to "model" in logic.) Students are more interested in this. They do some form of database searching almost every day, and are frustrated at their inaccuracy. Students learn for example how to approximate Boolean search on Google. Logical consequence drops out as a special case.

— The class and the teacher actually discuss. You don't just put up your hand when you don't understand something.

— The course rhetoric is neutral about the place of deduction in reasoning. The professor does not have to sell any debatable claims about effective thinking to get the class to see that the topic is interesting and important. (Instead of assuming them you can discuss them.)

— Students get more confidence and more facility with mathematically flavoured thinking. By the end students should see that they can handle some topics that they would earlier have found frightening.

— There is attention to the linguistic obstacles to phrasing a statement so it has the consequences you want, or a search command so it gets the items you want.

So the emphasis is on organizing thoughts in words, what linguists and philosophers of language call logical form, and the effect that awareness of this can have on your thinking and problem-solving. In this connection traditional "logic problems", of the kind found in logic puzzle books and aptitude tests, but not usually found in formal logic textbooks, are discussed. This occurs mainly in the exercises. In fact, the exercises of every chapter contain one traditional logic puzzle.

The exercises are important: you will get much more out of the course if you do most of them, just as if you were learning a language. The exercises serve another function, too. They prepare ideas that will be discussed in later chapters. This is important because some of the more difficult topics in most logic courses come towards the end, when everyone is tired and there are many pressures on students. If you have done the earlier exercises you will be ready for the later ideas when you meet them. I have placed these anticipatory exercises fairly early among the exercises for each chapter, to make it harder to avoid them.

The book is shorter than it may seem. Putting it on the web allows me to be generous with spacing and type sizes, and there are many varied exercises. I use a lot of tables and diagrams. There are no more words in the chapters, ignoring exercises than in a rather smaller book.

It is useful to be able to think complicated thoughts. Science, mathematics and economics would be impossible without them. So the delicate art of expressing yourself accurately in language has a practical value. I think, if you will indulge me in a somewhat mystical idea, that there is another, less practical, value to clear and subtle expression. Having complicated stuff in mind is a central part of being human. Bees make honey, birds sing, ants make ant-hills, and humans have subtle and complex thoughts. When we write stories, create music, do mathematics, think about the universe, or make jokes we are doing some of what we are here to do. And if you can handle logic you can do another small part of all this.

I have had help and advice on this project from a number of people over the years. They include Mia Bertanjoli, Jaqueline Leighton, Lisa Matthewson, John Simpson, Mojtaba Soltani, and many students who made wonderful comments on what was and was not working for them. Alirio Rosales gave me a lot of help in the final preparation of the document.

.

Vancouver, October 2017

chapter one: representing information

1: 1 (of 11) logic

Logic is about information. We represent information, also called data, in the form of sentences, tables, diagrams which impose various structures on it, and then we extract the same or related information and use it in solving problems. Logic traditionally focuses on *arguments,* in which information in the form of sentences is *deduced* from *premises* or *assumptions*. For example we might have as assumptions .

the keys are either on my desk or in my backpack

the keys are not on my desk

and from them we could deduce the conclusion

the keys are in my backpack

The conclusion is a *logical consequence* of the assumptions: it can be correctly deduced from them. We discuss logical consequence in part II of this book.

>> "correctly deduced": that was a substitute for an explanation to come later. but what would it make sense to count as correct here?

>> throughout this book, the text will be interrupted by remarks and questions formatted like this one. if you are reading the text it would be a good idea to pause and consider your reaction to them. they are likely topics for questions and discussions in class.

>> what other conclusions could be deduced from these assumptions?

Deducing conclusions from given assumptions is often not the best way of solving a problem. The issue will be discussed later in the book, in chapter 6. But the central problem is that there will be many assumptions that can be deduced from any assumptions. And deducing as many as you can of these consequences will not show that some particular sentence can *not* be deduced from them. So given some information and a question to answer in terms of it, you might deduce away for a long time without getting an answer to the question, and still not know what the answer is. This should be clearer from the example below.

I am not going to offer you a magic problem-solving method. This does not exist. But the resources of logic are helpful in other ways besides describing logical consequence and deduction from assumptions. Two ways that we will discuss in detail are forming clear instructions for searching for information, and producing clear unambiguous language in terms of which assumptions, conclusions, and the facts that make them true or false, can be stated. These are our main concern in this first part of the book.

>> instructions can tell us how to do things besides search for information. which of these other tasks have searching as a part?

1:2 (of 11) an example

You are investigating a crime involving corruption in high places and you have exactly five suspects. You have put the relevant evidence for each in a numbered dossier, and you have a summary sheet of notes, indicating each suspect with one of the code-names Red, Green, Yellow, Blue, and Purple. To help you remember which powerful politician is which, without risking writing down their names, you have memorized the following three facts:

- The dossier for the politician you have named Red is either number 2 or number 5.
- The dossier for the politician Green has a higher number than the dossier for the politician Yellow.

- The dossier for the politician Blue has a higher number than the dossier for the politician Green and lower than the dossier for the politician Purple.

A quick-witted friend looks through the dossiers and says "dossier 3 is pretty incriminating, but I can see that there are some politicians it cannot be." Which politicians, given the information just stated, *cannot* be the subject of dossier number 3?

To begin answering this question, summarize the relevant information. As fully stated in language it could be overwhelming, so inventing a notation for writing down the relevant facts is helpful

the politicians are: R, G, Y, B, P

R = 2 or R = 5

G > Y

B > G

P > B

[suggestion 1: it helps to have the knack of inventing notations so you can focus on what is relevant]

>> "the relevant information": what information does this notation leave out?

Now we know that there is an ordering of the colours/politicians, but we do not know where R fits in it. There are two possibilities, which we can represent as follows:

	possibility A	possibility B
1	P	P
2	R	B
3	B	G
4	G	Y
5	Y	R

We could have described the reasoning that leads to these as a step-by-step process, but the way most people would do it is to see first that R has to be at 2 or 5 position, and then to fit the others in.

>> what are we assuming without stating it, in doing this? why is it best not stated?

suggestion 2: it helps to see that there are only so many possibilities, and then to represent them in a way, particularly a diagram, that allows us to use spatial or numerical thinking.

You tell your friend that the dossiers are in order, from 5 for the most guilty-seeming, to 1 for the least. P is obviously the least guilty, but who is the next lowest? Your friend reads through the dossiers and says "Red seems guilty as hell to me, the most likely to be the culprit." Assuming this is right, who is the second to least likely?

>> well, who?

>> suggest some other questions that now have easy answers

The reasoning you did to answer this second question probably took the form of "this therefore that" thinking between sentences. $R = 5$, $R = 2$ or $B=2$, if $R=5$ then $R \neq 2$; therefore $B=2$. This kind

.of thinking is a traditional topic of logic, and this book discusses it from chapter four. But it is important to see that we cannot often solve a problem simply by applying it.

1:3 (of 11) databases

This book is not about solving logic problems, of the kind found on SAT tests and other student-torturing devices. In fact, symbolic logic as taught in philosophy and math courses rarely pays any attention to them. But we will regularly mention them and related activities, in part because they illustrate points about reasoning and the structuring of information, and in part because the skills they require and develop are useful. In the rest of this chapter and the next, we are concerned with databases, like those the diagrams for possibility A and possibility B in the example represented.

Very often when we have a collection of information, from which we can get answers by asking suitable questions, the collection of information can be called a *database*. I am going to use the term “database” to cover a wide range of things, basically any collection of information (data) from which more specific information can be got by asking a precise question. In particular, the term applies to what everyone includes as databases, namely collections of information on computer systems that can be accessed by using programs designed to get information out of them. So every time you look something up on Google, or find the location of a book by using the library computer, you are dealing with a database. In logic, a database is very often called a *model*. I shall use both terms, usually saying “database” when we are searching *in* them, and “model” when we are searching *for* them, as we do later in the course. I shall rarely use the word “model” until chapter four, and then in five I shall switch and start saying “model”. This could be confusing, so I am warning you now.

>> what about diagrams? what kinds of diagrams will fit this rough description of a database?

Information in computer databases and other kinds of databases (such as telephone directories or boxes of index cards) can often be represented as a *table*. A table has vertical columns and horizontal rows. The intersections of the columns and the rows are called *cells*, in which we can store little nuggets of info. (So for example the cell at the intersection of the **a**lice row and the **H**ungry column in the table below stores the information YES.) There are several different ways of writing down a database as a table. One very simple way is possible when the information concerns a definite finite set of individuals and we want to know which ones have which of several *attributes* or *properties*. The set of individuals is called the *domain* of the database. For example suppose the domain consists of six dogs who – at some given time - can have the attributes of being hungry, or angry, or sleepy. Then we might represent information about them as follows.

dogbase1	**H**ungry	**A**ngry	**S**leepy
alice	YES	NO	NO
brutus	YES	YES	YES
caspar	NO	NO	NO
doodles	NO	NO	YES
eloise	YES	YES	NO
flossie	YES	NO	YES

>> the domain of a database is a "definite finite set of individuals". describe information that is not about a definite set of things. can there be information about an infinite set?

Call this an *object and attribute* table. (When I speak of a table without further explanation I will mean an object and attribute table.) The cells in the column to the left have names of *individuals*, and the cells in the other columns have *truth values* (true or false, yes or no.) Notice the way I have chosen a **bold** upper-case letter for each attribute and a bold lower-case letter for each individual. This allows a quick way of referring to individual cells: for example the cell for **A**ngry and **c**aspar is **Ac.** The content of **Ac** is NO, so the sentence "Caspar is angry", is false. We can save words just by saying "**Ac** is false" or "the truth value of **Ac** is False". If you want you can think of a single cell such as **Ac** as a tiny database, with one individual and one attribute. It is what we will later call an atomic proposition.

1:4 (of 11) queries

We can get a lot of information out of a simple table like this, if we ask the right questions. A helpful standard way to ask the questions is to begin them with the request "find". So

Find the hungry dogs.

Find some angry dog.

Find the dogs that are sleepy.

.Find the dogs that are not sleepy.

Take a moment to figure out the answers to these questions, easy though they are. Notice how you can find the answers by looking for the right pattern of YESs and NOs in the right places. So to find all the dogs that are not sleepy you go down the **S**leepy column and look for the NO cells, and then collecting the dogs on the same rows as these cells. So you get Alice, Caspar, and Eloise. I have repeated the table, below, for ease of reference.

>> do you always need all the objects fitting the description (criterion)? how would you word the question or instruction to ask for less?

These simple questions can be combined to get more complicated questions. We can ask

- Find the dogs that are both hungry and sleepy.
- Find the dogs that are hungry and not sleepy.
- Find the dogs that are either hungry or sleepy. [see the remark on OR in section 4 below]
- Find the dogs that are hungry and sleepy but not angry. [this is the same as "hungry and sleepy and not angry".]

- Find the individuals — dogs in this case — that are not hungry, and are angry.

- How do the hungry dogs compare to all the dogs (more, same, fewer)?

How do the dogs that are not hungry and are angry compare to all the dogs?

dogbase1	**H**ungry	**A**ngry	**S**leepy
alice	YES	NO	NO
brutus	YES	YES	YES
caspar	NO	NO	NO
doodles	NO	NO	YES
eloise	YES	YES	NO
flossie	YES	NO	YES

Can you figure these ones out? Do it yourself before checking just below. And notice how here too you can find a quick little routine with the YESs and NOs for each one. I'll write the answers with an obvious shorthand that will prepare for something later on:

H & S : b, f
H & not S: a, e
H or S: a, b, d, e, f
H & S & not A: f
not-H & A: the null set, **Ø**
H compared to all dogs: fewer - some but not all of them
not H & A compared to all dogs: none of them

Notice how the routine you use to check for "hungry and sleepy" differs from the routine you use to check for "hungry or sleepy". We'll come back to that. Notice also that when no individual satisfies the search criterion, there still is an answer, namely the null or empty set, usually written **Ø**. (If I ask you to bring me, in a bag, all the presents under the tree with my name on them, but no one has given me a present, you come back with an empty bag.)

"Find all" begins an instruction, asking for the objects meeting some criterion. The criterion can be simple, as in “find all the individuals that are hungry” or complex, as in “find all the individuals that are either hungry or not asleep”. Commands to find things meeting some criterion are called *queries* or *search commands.* There is another way of thinking of these searches, not as commands to find things but as questions about the identity of things. For example

“They are hungry and sleepy: who are they?”

“I am hungry and sleepy and not angry: who am I?”

Seen this way riddles, found in all human cultures and popular with small children, are a kind of query. “I have four legs in the morning, two at noon, and three in the evening: what am I?”, “Brothers and sisters have I none, but that man's father is my father's son.”, “What is it that you will break even when you name it?”

>> riddles have a charm that most other queries lack. why?

We can search for many things. Just considering databases, we can take one and search for the things in it that meets some condition or criterion. The examples In this section so far have been like this. Or we can take a database and a sentence and ask for its *truth value,* whether according to the database it is true or false. Note that I said "according to the database": many databases contain false information. For example in this database

	Elephant	**O**strich
london	YES	NO
beijing	NO	YES

London is an elephant and Beijing is an ostrich. But that is absurd. In some books you may see the phrase "true in database D" (or "true in model M": databases and models are really the same; we'll get to that). But I shall avoid this phrase because it can be confusing and instead I will talk of a sentence holding in a database, and sometimes a database (or model) making a sentence true. "London is an elephant" holds in the database above, although it is not true (in the real world).

Suppose we are given two databases, dogbase1 above and also dogbase2 below.

dogbase2	**H**ungry	**A**ngry	**S**leepy
alice	NO	YES	YES
brutus	NO	YES	YES
caspar	YEd	NO	NO
doodles	YES	NO	YES
eloise	NO	YES	NO
flossie	NO	NO	YES

Then if we are asked to find which of these databases "Eloise is hungry" holds in, we give the answer dogbase1, since it does not hold in dogbase2. Searching for databases is important, as it connects search, our interest in the first four chapters, to logical consequence, the traditional topic of logic. (In fact they are both forms of reasoning, and pretty easily converted into one another.)

Simple queries require you to come up with a list of things from a database. The activity you do in response to a query is a *search*. So if I say "go to the refrigerator and get me the vegetables that are red and not rotten" those words are the query, and the search consists in you going over to the fridge and getting, say, a fresh tomato and three radishes, leaving two rotten tomatoes to ooze in the bottom drawer. In this case you are physically going to the objects and getting the desired ones. One general image is of a net that you sweep through a pool of candidates, catching just the right ones. A different image is of a filter. You might pour the collection through a filter that only allows things of the right kind to pass through.

>> "Simple queries" fetch objects from the domain. queries which get YES or NO (T or

F) or sets of pairs or databases themselves do not. how can we describe these queries so that they too fetch things from a domain?

We search on the internet most days of our lives, whether on the well-known search engines such as Google or DuckDuckGo or using the search facilities of online stores, libraries, and specialised sites. The amount of information to be retrieved this way is so enormous that we have to think hard about how to formulate our queries so that we increase the chances of getting the information we want. Logic is very relevant here, as we shall see.

1:5 of 11) what we can search for

Databases concern individuals and their attributes. (And relations between individuals, as we will soon see.) So searches in databases are usually aimed at finding individuals having particular attributes, especially complex attributes defined in terms of simpler ones. But we can search for many other things also. Two important searches are for truth values and for databases themselves.

The *truth value* of a sentence is True if it is true, and False if it is false. (Now that does not sound surprising, does it?) We can take a database and search for the truth value a particular sentence. For example we can take the database of dogs and search for the truth value of "Doodles is angry". We do this by locating the individual d in the database and locating the attribute A, and then seeing whether the cell where these meet has a YES or a NO. If it is YES then the truth value is True, and if it is NO then the truth value is False.

To illustrate a search for a database consider the following three individual and attribute tables.

Mon	**A**ngry	a**S**leep		Tues	**A**ngry	a**S**leep		Wed	**A**ngry	a**S**leep
d	YES	NO		**d**	YES	NO		**d**	NO	YES
f	NO	YES		**f**	YES	NO		**f**	NO	YES

With these three databases, we can we can say "find the databases in which Flossie is asleep ", or "find the databases in which neither Flossie nor Doodles is asleep". We can also ask more complicated questions, such as "find the databases in which only the angry individuals are asleep". The answers to the three searches are {Mon, Wed}, {Tues}, and {none of them}.

The searches can be aimed at truth values, also. We can say "find the database where the truth value of **Sf** is True", and so on.

I described these as searches for databases, but each databases is named with reference to a sort-of individual. So we can rephrase them as "find the days of the week (within this range) when Flossie is asleep", and so on. This illustrates the point that searches for databases and searches for individuals are rather similar. In the early chapters of this book the searches are for individuals, but it should not be shocked when later on it is databases (models) that we search for.

In everyday life we switch easily between searching for individuals and searching for the locations within which we can find them. In fact, we do not make a sharp distinction between these. We can search in the rooms of the house for a lost phone, and also search for the rooms in which it might be. (Suppose that it has a GPS function that we can access on the Internet, which is accurate enough to tell us which room it is in but not where in that room.) For many purposes after identifying the room we will have to search in its for the phone, but there are purposes for which it would be enough to know which room it was in. (Suppose we want to avoid taking a nap in the room where the phone might ring.)

To repeat, the material in this section is just background for the following chapters. There will be more detail about searching for databases when it is needed.

1:6 (of 11) a remark on OR

In English sometimes when we say "or" we mean "one or the other but not both" - the "exclusive" sense of or — and sometimes we mean "one or the other and maybe both" — the "inclusive" sense. But, when in logic we say "or" we mean the second: "at least one of the two (perhaps one, perhaps both, just not neither)". It's what "or" means when you say:

> ”It will rain or it will snow.” Surely this is still true if it rains and snows.

> "You can come to the party if you are my friend or if you bring a present." Surely this doesn't mean that if you are my friend and bring a present we won't let you in.
>
> "He is either lying or confused." Still true if he turns out to be both lying and confused.

Notice that in the last of these examples the OR is inclusive, even given the "either". We can illustrate this point with examples about searching, too. Suppose I say "get me all the books that are either valuable or have lurid covers". You go into the next room and you see a book that is valuable and has a lurid cover. Do you bring it? Sure. (On the other hand when we say "for five dollars you can have soup or salad" we probably mean that you can't have both.)

>> there is an important distinction in the philosophy of language between statements which are false and those which are misleading, because they can lead people to have false beliefs. give some examples. how is this relevant to the two senses of OR? Which one does it suggest is basic? (hard questions)

1:7 (of 11) relations

Much information cannot be represented with an object and attribute table. For most information is based not on individual things having single attributes, but on several things bearing some *relation* to one another. So, to stick with our six dogs, we might want to know which dogs chase which other dogs, on a particular morning. Suppose that Alice chases Caspar. We cannot express this by saying just "Alice chases and Caspar is chased", as that would be true if Alice chased Doodles and Flossie chased Caspar but Alice did not chase Caspar. We need to have, as basic units of information "_ chases …". That is, we need the information how the dogs relate to one another. These are relations. "Chases" is a two-place relation, as are "loves", "is to the north of", and many others. There are also three-place relations, such as "is between" ("Calgary is between

Vancouver and Montréal", "NYC is between DC and Boston") and "is the sum of" ("17 is the sum of 9 and 8"). An important fact about language and about reality is that *we cannot describe the world without relations*. One-place attributes are not enough. A lot of the complication of language comes from the need to express relations, and to say which individuals bear which relation to which other individuals. Symbolic logic gives important insights into how we think with relations and what the facts we express with language are. The way these things are expressed in logic is rather unfamiliar to people used to normal spoken languages, though. Relations will be just a small incidental complication here at the beginning of the course, but they will become very important. So, beginning in this chapter, the way logic treats relations will be introduced bit by bit, so that it gradually comes to seem natural.

1:8 (of 11) relational grids

Information about relations can also be given in tables. Suppose that some of our six dogs chase some others, on a particular day. We can represent this with a different kind of table, with the same domain. (Watch out. This table looks like the object-attribute tables we have been using, but it represents information in a different way.)

Chases1	**a**lice	**b**rutus	**c**aspar	**d**oodles	**e**loise	**f**lossie
alice	YES	YES	YES	YES	YES	YES
brutus	YES	NO	YES	YES	NO	NO
caspar	NO	NO	YES	NO	NO	NO
doodles	NO	NO	YES	NO	NO	YES
eloise	NO	YES	YES	YES	NO	NO
flossie	YES	NO	YES	NO	NO	NO

Call this a *relational grid,* or for short a *grid*. Note that in reading it order is important: Alice chases Caspar but Caspar does not chase Alice. We start from the names listed vertically and we use the YESs and NOs to relate them to the names listed horizontally.

This is a reflection of another basic fact about language and the world: *many relations are intrinsically one way – they often hold between a and b but not between b and a*. The most famous such relation is “loves”: where would literature be, were it not for the fact that often he loves her but she doesn’t love him? (A relation like this, where sometimes one object bears it to another but the other does not bear it to the first, is called *asymmetric*. We live in an asymmetric universe: there are many such relations.)

The cells of a relational grid say of two individuals whether or not they are connected by the relation. We refer to, for example, the cell in the **C**hases grid at the meeting of the Doodles row and the Eloise column, as **Cde.** It has a NO, so that the sentence “Doodles chases Eloise” is false. Again we can save words just by saying “**Cde** is false” or “the truth value of **Cde** is False”. Or just “**not Cde**”. Note that the order is important: the content of cell **Ced** is YES, Eloise chases Doodles, and the truth value of **Ced** is T.

>> this grid has a row consisting entirely of YESs, and a column consisting entirely of YESs. what does this show about the relation? can we have an all-YES row without an all-YES column?

1:9 (of 11) questioning grids

We can get information out of a relational grid by asking questions, too. Given the table above we can make the following queries:

1. Find the dogs who Alice chases.
2. Find the dogs who chase Alice.
3. Find the dogs who chase Caspar.
4. Find the dogs who are chased by Caspar (i.e. all the dogs who Caspar chases.)
5. Find the dogs who Brutus does not chase.
6. Find the dogs who do not chase Doodles.

7. Find the dogs who are such that Brutus chases them and they chase Doodles.

8. Find the dogs who either chase Brutus or chase Flossie.

9. Find the dogs who either chase Brutus or are chased by Flossie.

10. Find the dogs who chase themselves.

11. Find the pairs of dogs such that the first chases the second but the second does not chase the first.

Though these are fairly simple questions, they can be confusing. Part of the confusion comes from the English language, and we will soon introduce some notation to make things clearer. The answers to these questions are:

1. dogs who Alice chases: **a, b, c, d, e, f**

2. dogs who chase Alice: **a, b, f**

3. dogs who chase Caspar: **a, b, c, d, e, f**

4. dogs who are chased by Caspar: **c**

5. dogs who Brutus does not chase: **b, e, f**

6. dogs who do not chase Doodles: **c, d, f**

7. dogs who Brutus chases and who chase Doodles. **a**

8. dogs who either chase Brutus or chase Flossie: **a, d, e**

9. dogs who either chase Brutus or are chased by Flossie: **a, e, c**

10. dogs who chase themselves: **a, c**

11. pairs of dogs such that the first chases the second but the second does not chase the first: **(a,c), (a,d), (a,e), (b,c), (b,d), (d,c) , (d,f) , (e,b), (e,c), (e,d), (f,c)**

(You may wonder how Alice and Caspar manage to chase themselves. Well, in Alice's case, when all the others are hiding from her she gets bored and chases her own tail. Caspar on the other hand is a rather stupid and frightened dog, and sometimes he gets a glimpse of his own shadow and runs away from it. So we might say that Alice chases herself and Caspar is chased by himself.)

Notice how to answer some of these questions we have to scan columns looking for patterns of YES and NO, and to answer others we have to scan rows. To answer a few we have to scan both. This is linked to the difference between “chases” and “is chased by”. To find those who Alice chases – who are chased by Alice - you look along the Alice row; to find those who chase Alice you look down the Alice column.

>> which of these questions are harder to answer? why?

With databases in the form of relational grids we can search for truth values just as we can when they are individual and attribute tables. For example given **C**hases1 we can search for the truth value of "Brutus chases Elose", and find the answer is False. And given another grid as well, for example Chases2 below, we can ask which of them "Eloise chases Brutus" holds in. The answer is that it does in both of them.

chases2	**a**lice	**b**rutus	**c**aspar	**d**oodles	**e**loise	**f**lossie
alice	NO	NO	NO	NO	NO	NO
brutus	NO	NO	YES	YES	NO	NO
caspar	YES	NO	YES	NO	NO	NO
doodles	YES	NO	YES	NO	NO	YES
eloise	YES	YES	YES	YES	NO	NO
flossie	NO	NO	YES	NO	NO	NO

1:10 (of11) different ways of representing relations

This same information could be represented in different ways. We could use a somewhat different table.

Chases

alice	alice
alice	brutus
alice	caspar
alice	Eloise
alice	doodles
alice	flossie
brutus	alice
brutus	caspar
brutus	doodles
caspar	caspar
doodles	caspar
doodles	flossie
eloise	brutus
eloise	caspar
eloise	doodles
flossie	alice
flossie	caspar

This is a more cumbersome way of presenting this particular information. But it is quite standard in computer science, and has some advantages, for example with three and more place relations (see below). Call this a *list of tuples table.* (Because in this case it is a list of pairs of individuals. If the relation was three place we would have a list of triples of objects. Pairs, triples, quadruples, … are often called n-tuples, or just tuples.)

We could also present the same information as a set of sentences:

alice chases alice

alice chases brutus

alice chases caspar

alice chases eloise

alice chases doodles

alice chases flossie

brutus chases alice

brutus chases caspar

brutus chases doodles

caspar chases caspar

doodles chases caspar

doodles chases flossie

eloise chases brutus

eloise chases caspar

eloise chases doodles

flossie chases alice

flossie chases caspar

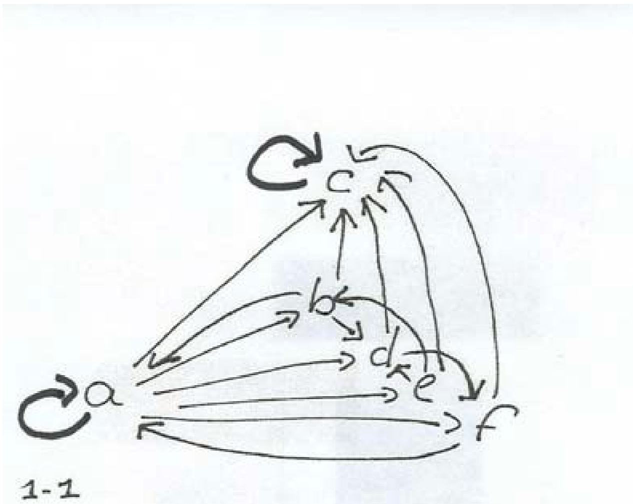
You can see that this is not a very good way to present information, if we want to get answers out of it without much trouble. No wonder that we use graphs, tables, and diagrams.

>> are there situations where this would be a useful way to present information?

The tables we looked at earlier listed objects by their attributes, and these tables we have seen in this section list objects by their relations to one another. Attributes and relations are both expressed in language by *predicates*, sequences of words which can be true of things. Thus "is human" is a one-place predicate true of Ada Lovelace (the first computer programmer), Beethoven, and of you, and "is a dog" is a one-place predicate true of Doodles and of Flossie, because, for example, "Ada is human" and "Flossie is a dog" are is true sentences. "is to the north of" is a two place predicate, corresponding to true sentences like "Toronto is to the north of New York City". And "is between" is a three place predicate, since there are true sentences like "Nanjing is between Hong Kong and Beijing"

1:11 (of 11) graphical representations

Sometimes it is more intuitive to give data in the form of a picture. With a two place relation we can draw a picture by using an arrow to connect two objects when the relation connects them. For example the chasing relation between the six dogs could be given by the picture below.



I will call a picture like this an *arrow diagram*. There is an arrow for each pair (x,y) where x has whatever relation we are describing to y. Sometimes an arrow will go from x to y but not back from y to x. Then x has the relation to y, but y does not have the relation to x, as when Xu sees Yang, from his hiding place, but Yang does not see Xu. Arrow diagrams are going to be important in this course, so it is worth taking the time to become comfortable with them.

The special roles of Caspar and of Alice are easy to see from this picture. (They're not hard to figure out from the relational grid table, either, but they do not exactly leap to the eye from the list of tuples table. And you might never notice them when the data is a set of sentences.) You can think of the darker arrows looping from each to itself as not only showing that Alice chases Alice and that Caspar chases Caspar, but also as saying "hey, notice these two: they're special." The arrow diagram has been laid out to show this: the graphical arrangement of a diagram makes a big difference to what information is easily got from it.

Another way of putting the point about the roles of Caspar and Alice is that the diagram makes it easy to see the results of some particular searches

> "Find the dogs that chase all dogs" gets **a**
>
> "Find the dogs that all dogs chase" gets **c**
>
> "Find the dogs that Alice chases" gets everything
>
> "Find the dogs that chase some dogs" gets everything.

(The question is not "...that chase all *other* dogs": it asks for a dog that chases each one of the dogs, and that will include chasing him or her self.) The interesting point now is how finding the dogs who chase all dogs and who are chased by all dogs is easy given the arrow diagram. Alice is at the centre of a star of arrows going to every dog, and Caspar is the target of a flock of arrows coming from every dog. It's not much harder to see this information from the relational grid table: Alice is at the beginning of a full row of YESs and Caspar is at the head of a full column of YESs. (But it takes a moment's thought to see why the row of YESs means that Alice chases all the dogs, or why the column of YES means that Caspar is chased by all the dogs.) So this particular relation has the special features that there is something that has the relation to everything, that there is something that everything has the relation to, and everything has the relation to something. There are exercises at the end of the chapter to make arrow diagrams more familiar to you.

>> think of an arrow diagram that is harder to work with than the corresponding grid

There are many ways of giving information visually, and different ones work best for different purposes and different kinds of information. We use many different kinds of tables, graphs, and diagrams. We often use diagrams that look like arrow diagrams without the arrows. And when we do include arrows or something serving the function of an arrow we often do not indicate whether when the relation holds between two objects in one order it hold in the reverse order, whether it holds between an object and itself, and so on. For example, in the diagrams for possibility A and possibility B of the example in section 2 I wrote P above R in possibility A, and so on, and R above G, without stating that this means that the "lower dossier number" relation also holds between P and G. This didn't need stating because it was clear from the meaning of the words in the problem. But if the words had unusual meanings, as they often do in logic and math, then it would have been necessary to state this.

Maps are a good example. Here is a map of the subway system of an imaginary city.

airport - **s**tadium - **h**appyGrove - **b**usinessDr - **c**ityHall - **t**rainstation - **d**epressionville – **r**ichland

The relation sign "-", between **a** and **s** and so on, could be several relations. It could say of **a** and **s**, for example, that they are neighbouring stops, or that passengers can get from one to the other, that the first is further out from the center than the second, or that the first is to the west of the second. Since you are used to reading maps and know how transit systems are laid out, you can get information about all of these from the map. (But if you were unfamiliar with maps or public transit it would be much less obvious to you.)

>> “neighbouring stops”, or “can get from one to the other”, “first is further out from the center than the second”,”first is to west of second”. are all of these equally likely interpretations of a map like this? if we wanted to block one of these interpretations how would we show this?

There are many examples of familiar types of diagrams. Family trees are another good example. When they represent a two place relation between individuals in a definite set, they can be translated into the arrow diagrams we will use throughout this book, though the arrow diagram .will sometimes have a confusing amount of "extra" information. There are exercises at the end of this chapter to make you more familiar with arrow diagrams and their connections with maps, family trees, and other diagrams. In later chapters we come to terms with relations that have more than two places.

words used in this chapter that it would be a good idea to understand: argument, arrow diagram, criterion of a query, deduction, database, domain of a database, object and attribute table, query/search command, predicate, relation, relational grid.

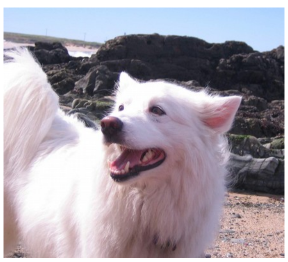
If you are uncertain about any of these you should ask.

The index at the end of the book says where terms are explained or defined.

And you thought I was making all this up ….



Alice Brutus Caspar



Doodles Eloise Flossie

exercises for chapter 1

These, like the exercises for most chapters, are divided into three parts. Part A is questions you should be able to answer. Part B is more of the same, in case you want extra practice. And part C is harder questions needing more sophistication or more reflection. It would be a good idea to look at the C questions even if you find them challenging, and to ask about issues that they raise. You will also find a few questions where you think "Hey: he did not say how to do this". This is deliberate. It is meant to make you think, sometimes in a way that prepares you for an idea in a later chapter. (Many things are easiest if you have figured them out for yourself before they were explained to you.)

A – core

1) Exactly six guideposts, numbered 1 through 6, mark a mountain trail. Each guidepost pictures a different one of six animals: fox, grizzly, hare, lynx, moose, or porcupine.

The following conditions must apply:

The moose guidepost is numbered lower than the hare guidepost.

The lynx guidepost is numbered lower than the moose guidepost, but higher than the fox guidepost.

Which of the following animals CANNOT be the one pictured on guidepost 3?

fox, grizzly, lynx, moose, porcupine

2) (In the same situation) if guidepost 5 does not picture the moose, then which of the following must be true?

- the lynx is pictured on guidepost 2
- the moose is pictured on guidepost 3
- the grizzly is pictured on guidepost 4
- the porcupine is pictured on guidepost 5
- the hare is pictured on guidepost 6

3) Describe your thinking for questions 1 and 2 in terms of (a) a diagram and (b) your use of the word "if".

4)

	Subway	**H**ockey	**F**rench-speaking
montréal	YES	YES	YES
toronto	YES	YES	NO
new york	YES	YES	NO
chicago	YES	YES	NO
paris	YES	NO	YES
t**r**ois rivières	NO	NO	YES

a) Find all the cities that do not have a subway.

b) Find all the cities that have a subway and do not have a hockey team.

c) Find all the French speaking cities that have a (NHL) hockey team.

d) Find all the French speaking cities that do not have a subway.

e) Find all the cities that are not French speaking and do have a subway) .

f) Find all the cities that either have a hockey team or are French speaking.

5) a) State a query which when applied to the database above will get Montréal and

Paris.

b) Write a query which will get Montréal, Toronto, and Chcago.

c) Write a query which will get all the cities to the east of Montréal.

6) Suppose you are searching for books on the library computer. (a) you enter "Morton" for "author", (b) you enter "Morton, Adam" for "author". Which request will give the longer list of answers? (More hits.) Why? Suppose you are doing a Google search and you enter "sex" and get zillions of hits. You add another keyword and enter "sex, chromosomes". Which will get you more hits? Why?

7) Fill in the blanks in the table below so that the question "I am rich and happy. who am I?" gets **b**o and **m**o, and "I am rich and not happy. who am I?" gets only **j**o.

	Rich	**H**appy
bo		
jo		
mo		

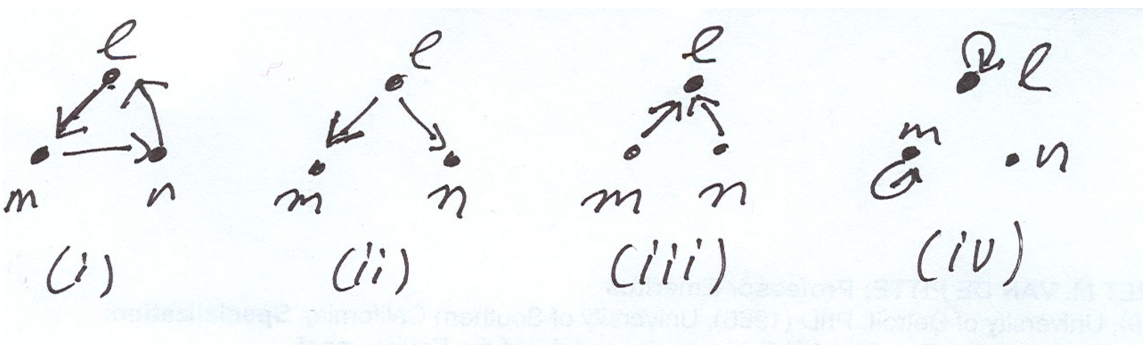
8) The database below has one blank cell. How must the cell be filled in so that "Find everyone who is either guilty or a suspect" gets the answer "zorro, yannis, xeno"? How must it be filled in so that "Find everyone who is neither guilty nor a suspect" gets "zorro". ("neither guilty nor a suspect" is a way of saying "not guilty *and* not a suspect.")

	Guilty	**S**uspect
zorro		NO
yannis	NO	YES
xeno	NO	YES

9) The database below has two blank cells. Suppose we know that before the cells were blanked out someone did a search for "Find everyone who is either dangerous or not crazy" and got Alfons, Casimir (and no one else). Can you fill in the blanks?

	Dangerous	**C**razy
alfons	YES	NO
bridget	NO	YES
casimir	NO	
dolores	NO	

10) Which grid ((a), (b), (c), or (d)) describes the same relation as each of the arrow diagrams below?



(a)	**l**	**m**	**n**		(b)	**l**	**m**	**n**		(c)	**l**	**m**	**n**		(d)	**l**	**m**	**n**
l	YES	NO	NO		**l**	NO	YES	YES		**l**	NO	NO	NO		**l**	NO	YES	NO
m	NO	YES	NO		**m**	NO	NO	NO		**m**	YES	NO	NO		**m**	NO	NO	YES
n	NO	NO	NO		**n**	NO	NO	NO		**n**	NO	NO	NO		**n**	YES	NO	NO

11) a) In the arrow diagram marked (ii) in exercise **7**, add arrows (perhaps in a different colour) to join all pairs that do *not* have the relation. That is, given any x and any y such that x is not joined by an arrow in the original diagram, join them by a (new, or additional) arrow.

b) In diagram (iv) circle those individuals that do not have the relation to themselves.

c) In diagram (ii) add arrows (perhaps in a different colour) to join all pairs such that the relation holds in one direction but not in the other. (That is, such that the first has the relation to the second but the second does not have the relation to the first.)

(This question describes a kind of search, defined in terms of relations rather than attributes. Do you see why?)

12) The table below has some blank cells. All the same, you can know what the answers to some searches are, and whether some sentences are true in the database (that is, whether the search for "what is the truth value of the sentence" gets the answer "True".)

	Cat	**D**og	**A**nimal
garfield			YES
tabitha	YES	NO	
macavity	YES	NO	YES
alice	NO	YES	YES

For each of the following queries, either give the answer, or state that there is not enough information in the database.

a) Find all the individuals that are either animals or cats.

b) Find all the individuals that are cats and not animals

c) Find all the individuals that are animals and not cats

d) Find all the individuals that are cats and animals

e) Find the truth value of "some cats are animals"

(warning: the database does not require that all cats are animals.)

B – more

13) In the table for exercise **9)** for each of the sentences below fill in the blank cells one way so as to make it true (the answer to "what's its truth value" is True) and another way so as to make it false.

all cats are animals.

no dogs are cats

all cats that are not dogs are animals

14) In which of the databases below do the queries listed below them have the given answers?

1	**Cat**	**Curious**	**Fish**
Moggie	YES	YES	YES
Molly	NO	YES	NO
Millie	YES	YES	NO

2	**Cat**	**Curious**	**Fish**
moggie	YES	NO	NO
molly	YES	YES	YES
millie	YES	NO	NO

3	**Cat**	**Curious**	**Fish**
Moggie	YES	YES	NO
Molly	YES	NO	NO
Millie	YES	YES	YES

a) Find the truth value of 'some cats are fish'. True

b) I am a curious cat: who might I be? **o**

c) I am either a curious cat or a fish: who might I be? **m, l**

d) Find the truth value of 'all cats are fish'. False

15)

Beat	**l**eafs	**o**ilers	**d**ucks	**h**urricanes
leafs	NO	NO	NO	NO
oilers	YES	NO	YES	NO
ducks	YES	NO	NO	YES
hurricanes	YES	YES	NO	NO

(These data are not meant to be accurate!)

a) Find all the teams that the Oilers beat.

b) Find all the teams that beat the Oilers.

c) Find all the teams that beat some team that beat the Leafs.

d) Find all the teams that either beat the oilers or were beaten by the Ducks.

e) I beat a team that either beat some other team or that did not beat the Oilers: what might I be?

f) Are there three teams such that the first beat the second and the second beat the third, but the first did not beat the third?

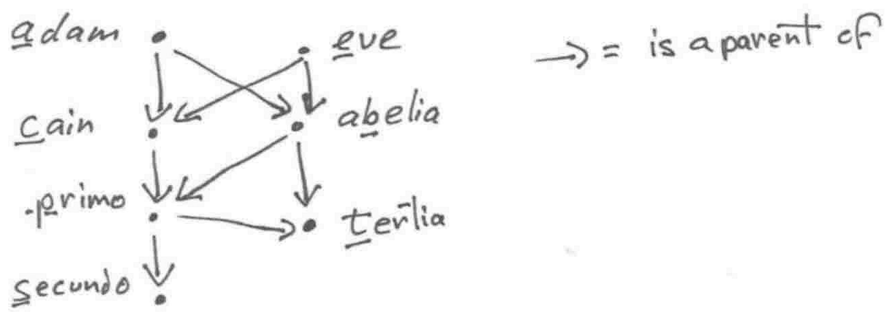
16) Find a query which will get the teams that beat some American team. [This is a type of problem that requires taking some factual information from outside the database and translating it into the language of the database. In case you need to know, the leafs are from Toronto, the Oilers from Edmonton, the Ducks from Anaheim California, and the Hurricanes from Carolina.]

17) Draw an arrow diagram containing the same information as the table in **12** above.

18) **a)** Who has the relation below to at least one individual ? **b)** Who to none ?
c) Who has the relation to an individual who has it to someone (is a grandparent)?

d) For which individuals are there two individuals that bear the relation to it (has two parents)?

e) find (pairs) x,y: x is a parent of y & y has just one parent (note that only this last question is asking for pairs) .



C - harder

19) Suppose that in the diagram of the subway in section 10 of this chapter the arrow had meant "you can get from the first station to the second station." What arrows would have to be added to make it an accurate map?

20) In the grid below one cell is blank. What answer does the query "Find everyone who admires Osman" have to get in order for that cell to be YES? Find a query such that if it gets the answer "Tom" then that cell must be NO.

Admires	**s**am	**t**om	**o**sman
sam	YES	NO	YES
tom	NO	YES	
osman	NO	YES	YES

21) In the database of question 18), which of the following get True, which False, and for which is there not enough information?

a) Find the truth value of "everyone admires himself".

b) Find the truth value of "everyone has an admirer".

c) Find the truth value of "all Tom's admirers are admirers of Osman".

d) Find the truth value of "someone has no admirer".

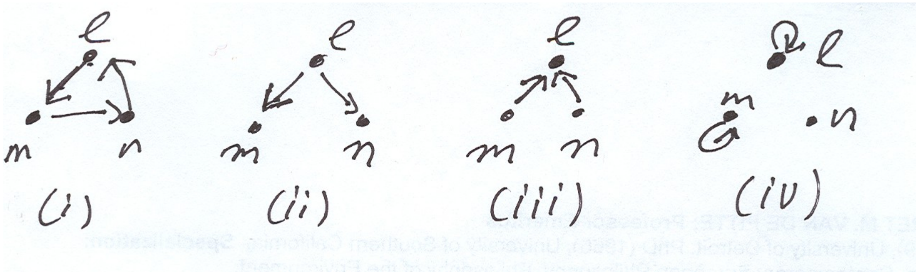
(Note: a person's admirers are the people who admire her, which may not be the same as the people she admires.)

22) In the database of **20**, how do we have to fill the blank cell in order to make "all Tom's admirers admire Osman" true? How do we have to fill it in order to make "everyone who does not admire Sam admires Osman" false?

23) a) Make an arrow diagram in which there are arrows joining all pairs which have the relation in (i) below and do not have the relation in (ii). (So by combining the relations in this way we have got another relation.)

b) Make an arrow diagram with arrows joining all pairs which either have the relation in (ii) or do not have the relation in (i).

(Before doing either part of this question you may want to think about exercise **11**.)



24) Some tables are given in section 2 are special in that every set of individuals is the answer to a question. When will this be the case?

25) Can you find a pictorial way of representing the "chases" information about the six dogs in section 8 of this chapter, that is not an arrow diagram?

26) Give examples of four and five place relations.

27) (indexed tables) I said that relations cannot be replaced with one-place attributes. That is true. But there is a way in which a set of object and attribute tables can represent a relation. We can make a separate object and attribute table for each individual in the domain, showing what other individuals it has the relation to. So with the Chases database we could have six tables, of which the first two are as follows.

	Alice-chaser
alice	YES
brutus	YES
caspar	NO
doodles	NO
eloise	NO
flossie	YES
	Brutus-chaser
alice	YES
brutus	NO
caspar	NO
doodles	NO
eloise	YES
flossie	NO

and so on. What would the other four tables look like? How could we extend this idea to represent a three-place relation in terms of two-dimensional grids? How could we use it to represent a three-place relation in terms of object and attribute tables?

chapter two: queries and searches

2:1 (of 9) scope

We get information out of databases by asking questions. But if you ask the wrong question you do not get the answer you wanted. In this chapter we discuss one aspect of carefully-worded questions, or queries, which feature in *Boolean* searches, focussing on the words “and”, “or”, “not” and “if”.

Consider again the simple object and attribute table from chapter one.

	Hungry	**A**ngry	**S**leepy
alice	YES	NO	NO
brutus	YES	YES	YES
caspar	NO	NO	NO
doodles	NO	NO	YES
eloise	YES	YES	NO
flossie	YES	NO	YES

Suppose we want to collect together the hungry dogs and the angry dogs. We want the hungry ones and the angry ones, so we should say “find the dogs that are hungry and angry”, right? Wrong. That search will get us the dogs which are both hungry and angry — Brutus and Eloise — while we want both the dogs that are hungry and the dogs that are angry — Alice, Brutus, Eloise, and Flossie. What we should say is “find the dogs that are **either** hungry **or** angry”. To put it another way: don’t confuse “if it is hungry include it and if it is angry include it” with “if it is hungry and angry include it”. The first is the one we want, and it is equivalent to “if it is hungry or angry include it”, as we will see later. (Stop for a moment and feel the difference in meaning between “if hungry then include and if angry then include" and "if hungry and angry then include.")

This confusion between "and" and "or" is very easy to make. It is not surprising that this can be confusing, as the English language is not very helpful here. In fact, this is a special case of a deep and general problem about English and other natural languages. They are not very clear in the way they indicate the *scope* of important words. Scope is a matter of the order in which words apply, which may not be the same as the order in which they are said or written. Imagine a conversation where we are talking about a would-be horror movie and I say "it is not very scary". You reply "that's under-stating it: it is very un-scary". I said *NOT VERY scary"* and you aid something different and stronger, VERY NOT SCARY. A movie at a scariness level of 5 on a 0-10 scale would be not very scary, but a movie at a level of 0 would be very not scary (very unscary). We will see examples contrasting *not all* with *all not*, *not believe* with *believe not*, *don't and* with *and don't*, *find both* contrasted with *both find*. So scope is a matter of which words would come first if you expressed yourself perfectly precisely. In the case we have just been discussing we have to distinguish between

> *Find all the dogs like this: each one is hungry and angry.*

and

> *Find all the dogs like this: each one is hungry. And Find all the dogs like this: each one is angry (and then combine the two)*

The first of these says "find (hungry and angry)" while the second says "find angry and find hungry". Or, to put it a third way, the first says

> *Find <u>all</u> of the dogs with <u>both</u> attributes: hungry, angry*

while the second says.

Do both of the following: find all of the hungry dogs, find all of the angry dogs

From this last way of putting it you can see why I said scope is a matter of what words would come first if we were speaking absolutely precisely. In “Find all with both” the “both” follows the “find all” — or as we say in logic “both” is within the scope of “find all” — while in “Both: find all hungry, find all angry” the “both” comes before the two “find all”s, which are within its scope. So it is the second that we should use when want to collect together the hungry with the angry. (And, as we’ll see soon, it can also be expressed with “or”.) There is scope in arithmetic too: half the square root of 9 is not the same as the square root of half of 9.

Suppose that you are getting things for me from a drawer. It has three red wool socks, two red polyester socks, two green wool socks, three blue silk socks, red shorts, and a diamond ring. I say to you "Get all the socks that are not wool and red." You should ask for more explanation before rummaging through the drawer. The request may mean.

(a) Get all the socks that are not wool and are red =

(NOT WOOL) & RED = the two red polyester socks

or it may mean

(b) Get all the socks that are not wool-and-red = NOT (WOOL & RED) = the two red polyester socks, the two green wool socks, the three blue silk socks

These are clearly different.

English and other spoken languages are usually rather unclear about scope, leaving it up to the common sense of hearer or reader rather than stating it explicitly. There are many examples. Suppose that you come home at 3 am and find a note saying

"DON'T come home late and take out the garbage".

This might mean two things.(i) Do not come home late, and also do take out the garbage. (ii) Do not do this: come home late and take out the garbage. The first might seem more likely but it is easy to imagine situations in which the second might be the message. (They bug you to take out the garbage but you never do, except when you come home in the middle of the night, singing, and decide to do it, clattering the garbage cans and waking everyone up.)

Or suppose one person says to another "I don't believe that there is a god." Is the person saying "I believe that there is not a god" or "It is not true that I believe that there is a god"? The second is what the person would mean if they thought there was not enough evidence to decide either for or against the existence of a god. To tell which one the first person means, whether they are an atheist or an agnostic, the second person may have to ask her to be more explicit.

>> which of these is the atheist's answer, and which the agnostic's?

Or suppose that one person says "all of my dates were not disasters". (Think of it as an angry reply to someone teasing him: all your dates have been disasters.) He may be saying "the following is not true: all of my dates were disasters" (that is, some of them were non-disasters). Or he may be saying "this is true of all of my dates: they were non-disasters" (that is, none of them were disasters.) You can't tell which if you take the sentence in isolation. A Volvo advertisement says "the world's first four-seat convertible with a three-piece retractable hardtop." When you first scan it you think "the world's first four-seat convertible" (and perhaps some part of your mind thinks "the world's first convertible with a retractable hardtop"). But all they are really claiming is to be the first convertible to have both 4 seats and a 3-piece retractable hardtop.

(A group of philosophers were planning a soccer game while I was writing an early version of this chapter. One sent an email to everyone saying "bring both light and dark T-shirts". The idea was that when we split into teams everyone could tell who was on which team. One person came wearing a striped T-shirt. Only a philosopher.)

These examples, and many more, show something that it helps to develop a feel for when studying logic (And which sharpens your awareness of what we communicate with language.) Soon we will develop a notation that makes it easier to be clear about these things. But the notation goes hand in hand with sensing these ambiguities in ordinary language: the notation makes it easier to sense them, and having a sense for them makes it easier to understand the notation[1].

I will sometimes use ways of writing sentences that are not regular English — and are not the official notation of logic either — which are meant to make scope distinctions easier to see. For example in English when we say "if you touch it, it will not break", we

1 In logic one item (sentence, predicate, or whatever) either is or is not within the scope of another. No overlapping, no halfway. Analogues of scope elsewhere in life are not so definite. Musical phrasing, for example, which puts one sequence of notes within another, is rather more fluid.

usually mean "it is not the case that if you touch it, it will break". That is to say, it is not fragile. But we might also mean "If you touch it, that will make it fail to break". That is, touching it will make it cease to be fragile. In the "Expanglish" ("English-ish", as I shall say in places) I will sometimes use, these could be written as

This is false. if you touch it, it will break

Suppose you touch it. then it will not break

And similarly we could say:

This is forbidden. You come home late and you take out the garbage🕐

You are forbidden to come home late. You are required to take out the garbage🕐

You must bring a dark T shirt. You must bring a light T shirt🕐

You must bring a T shirt. It must be light and dark🕐

The idea is that it is sometimes easier to be clear with a linked series of mini-sentences than with a single complex sentence. I will use this idea from time to time in later chapters, often without remarking on it. (If your main language is not English please note that this is funny Adam Morton language, and not regular English. Don't give it to other profs.)

>> express the difference between "I am indifferent to onion-flavoured ice cream" and "I avoid onion-flavoured ice cream" using the words "want" and "not" but without "indifferent" or "avoid".

>> "I must do it" and "I have to do it" mean (almost) the same, but "I mustn't do it" and "I don't have to do it" have different meanings. How can this be?

2 (of 9) Boolean connectives

We can use *and, or*, *not* to combine attributes or relations or whole sentences. (*If* too: I'll get to that.) The meanings of these words are clearest when we are dealing with individuals and attributes (one place predicates.) Consider a different object and attribute table (because you are probably getting bored with Alice and Caspar.) This is a table of actors

	Glamorous	**A**cts	**E**xpensive	**M**ale
philip	NO	YES	YES	YES
juliette	YES	YES	NO	NO
angelina	YES	NO	YES	NO
tom	YES	NO	YES	YES
natalie	YES	YES	NO	NO
dominic	YES	YES	NO	YES
cerris	YES	YES	NO	NO

Suppose that we are casting for a film and we want a male actor who is not too expensive for our budget. So we consult our database of actors — in reality it will have hundreds of names — and we look for the profile NO, YES in the Expensive and Male columns. We have to give some instructions to the computer, or to an overworked clerical assistant who is going through file cards. So we say "find all the actors who are not expensive and male". Now computers are very literal and clerical assistants can get very tired, so we must be clear that what we want is "not-expensive and also male" rather than "not (expensive and male)". In the first case we get just Dominic, and in the second we get Juliette, Angelina, Natalie, Dominic, and Cerris, which is surely not what we wanted.

Consider some other simple searches. We might want to find all the actors who can act and who are expensive. To do this we might first collect all those who can

act — all the YESs in the *acts* column — getting Philip, Juliette, Natalie, Dominic, and Cerris. Then we could refine our search, selecting from these those who are expensive, so we keep only those who also have a YES in the *expensive* column. The ones with a YES in the *expensive* column are Philip, Angelina, and Tom. And so the ones we want, the ones in both lists, consist of just Philip.

Or we might want to find all the actors who can either act or are expensive. So again we work with the list of actors who can act — Philip, Juliette, Natalie, Dominic, Cerris — and the list of actors who are expensive — Philip, Angelina, and Tom. But this time we want to take all the actors who are on either list. So we get Philip, Juliette, Angelina, Tom, Natalie, Dominic, Cerris: everyone.

Or we might be making a very low budget film and all we want are actors who are not expensive. So we start with the *expensive* list — Philip, Angelina, Tom — and we include everyone who is *not* on this list — Juliette, Natalie, Dominic, Cerris. Notice that this is a search that can be done in two stages: first search for the expensive individuals, and then search for everyone who is not on this list.

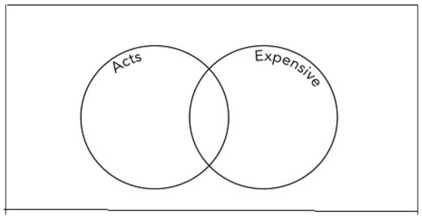
These are the three basic Boolean operators: AND, OR, NOT. In terms of tables, AND collects together two areas of a table to give the individuals that are in both. OR collects two areas and gives the individuals that are in either. NOT takes one area and gives everything except the individuals who are in it. So when we ask

Get all who are A and B we want: YES , YES, that is both of A, B columns.

Get all who are A or B we want: YES on one of A, B columns (doesn't matter which, can be both).

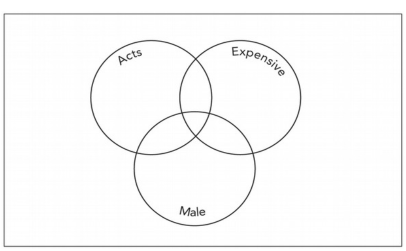
Get all who are not A we want: NO on the A column.

Another way of putting this is in terms of *Venn diagrams*, such as the one below.



In this diagram the square frame represents some larger set of individuals (the "universe" of the diagram), and each circle represents a set of individuals within that larger set. The two circles have a region which is in both of them. That is called the *intersection* of the two sets and corresponds to AND. (An individual is in the intersection if it is in the first set and in the second set.) There is also a region which contains all the individuals in either of them. That is called the *union* of the two sets and corresponds to OR. (An individual is in the union if it is in the first set or in the second set.) There is also for each of the two sets a region that contains everything (in the universe of the diagram) that is not in that set. It is called the *complement* of the set and corresponds to NOT. (An individual is in the complement of a set if it is not in the set.)

Some people find Venn diagrams very intuitive, and some are more comfortable with tables. We can also draw Venn diagrams with three attributes. (With more than three it gets messy.) For example



You should be able to see for yourself in this diagram the intersection of *expensive* with *act*, the union of *act* with *male,* the complement of *male,* and other similar regions. It is also clear from this diagram that if we start with some regions and define new ones by taking unions, intersections, and complements then we can define yet further ones by taking unions, intersections, and complements of the regions we have just got. For example once we have the intersection of *expensive* with *act* we can take the complement of that set. And what is that? — the set of all actors who are not both expensive and capable of acting. And once we have the union of *act* with *male* we can take the intersection of that set with the complement of *expensive*. What we get then is the set of actors who both either can act or are male and are not expensive. (As you see from this example, the words can get confusing. It is easiest to think of this one with the sets in a different order. It is the set of actors who are inexpensive and also either are good actors or are male. The complexity is still there, though; that is one reason for inventing a special clear notation.) And so on, we can define extremely complex sets using union, intersection, and complement repeatedly[2].

>> on a Venn diagram for the combinations of two attributes mark the patterns of YES and NO for each area.

>> what can you do in terms of YES/NO patterns that you cannot do with a Venn diagram?

These complex repeated Boolean operations can be easier to grasp if you think of them in terms of search questions. So instead of "the intersection of the union of act and male with the complement of expensive" we can think in terms of a three-stage search:

Find all actors who either can act or are male

Find all actors who are not expensive

---

2 Boolean operators or connectives are named after the English mathematician George Boole, 1815-1864, who was influenced by Ada Lovelace, who wrote the first computer program, and who himself influenced the English philosopher John Venn 1834-1923, the inventor of Venn diagrams. For an up-to-date take on the example of this section see http://www.nytimes.com/2009/06/28/business/media/28steal.html?ref=technology

Find all actors who are in both of these two sets.

We can compress these three stages into one structured search as:

Find all actors who are both A and B:

A: they either can act or are male

B: they are not expensive

This way of representing complex Boolean combinations in terms of stage-by-stage searches is related to scope distinctions. Remember the difference between "not-expensive and also male" and "not (expensive and male)" earlier in this section. That is the same as the difference between the two searches,

Find all actors who are not expensive

Find all actors who are male

Find all actors who are in both these sets.

and

Find all actors who are expensive

Find all actors who are male

Find all actors who are not in both of these sets

2:3 (of 9) mathematical mentality: thinking through searches

When you have a database and you are answering a "Find" instruction, one way of doing it is to repeat the instruction for every cell of the database. This is time consuming and is likely to leave you confused; at some point you may forget what it is that you are supposed to be doing just because your mind is getting overloaded. A better way is to think first which cells are going to be relevant; perhaps you will need to consider only one column or one row. Then you think what is going to guide your decision at each cell, what will make you save a particular individual has one that the search has found, and what will make you discard an individual. Then instead of thinking again about each case you go through the relevant parts of the database, following the decisions you have just made, and not thinking about each of them them as you carry it out. This way you do not need to do a difficult thinking except at the very beginning, and you do not have to keep much in mind except what you will need at the end.

In effect you are programming your mind to be a little automaton which can carry out the task without detailed supervision. You can save your sophisticated thinking power for programming your naïve thinking. A lot of mathematical thinking is like this: you think conceptually about how to use your spatial or symbol manipulating skills, then you use them automatically and then you reflect conceptually on what you have got[3]. (This last stage is important also, as you may have made a mistake in the automatic part and your answer may be absurd.)

Here is an example from high school math. You have an algebraic equation to solve. You think what your procedure is going to be: whether you are applying something like the quadratic formula or doing the same thing to both sides of the equation and then rearranging, or whatever. Then you carry out this procedure without thinking about what it means. You certainly do not think, for example "I am looking for a number which when

3 There is a basic dispute about how to teach mathematics in early grades, between those who advocate learning routines and those who advocate understanding what one is doing. Often parents are in the first group and teachers in the second. But if children have to guide each step of routine by understanding they get confused, and if they have no understanding they cannot tackle novel problems. I am suggesting a good combination is a rough understanding of a precise procedure that is best done by rote.

multiplied by itself and added to three times itself gives a total of 12" and let this guide you at each stage. Instead, you fix on a strategy, do it, and then check whether the number you have got is such that when squared and added to three times itself adds up to 12. The crucial thing is the mixture of conceptual and mechanical. You have to find the mixture that works for you particular kind of task. (Exercise 14 of this chapter connects with the topic.)

2:4 (of 9) sequential and branching searches, shortcuts, the structure of tasks

We can search using a particular criterion, and then search in the results of that search using a different criterion. That is like pouring the data through one filter, and then pouring what you get through another. We can also search for individuals that satisfy both criteria. That is like putting the filters together and pouring the data through the combined filter. The two are equivalent: they both amount to searching with AND. The first says:

Find all individuals satisfying F

then among the results

Find all individuals satisfying G

The second says:

Find all individuals satisfying F

Find all individuals satisfying G

Take the *intersection* of the results of these searches

Suppose we are looking in a domain of men for handsome unmarried individuals. If we do it the first way then stage one will get all handsome men, and stage two will get all the handsome guys who are unmarried. If we do it the second way then we will get, separately, all the handsome men and all the unmarried men, and then by taking the intersection of these we will end up with all the handsome men who are unmarried: the same set.

In fact, we could do it a third way, searching first for unmarried men and then searching among the results for handsome ones. Though these three ways will get the same results, one might be easier to perform than the others, especially if the domain is large. Suppose that you have a list of two hundred people who match the criteria you have entered on a dating site. You want to find a handsome unmarried man. (Assume for simplicity that people tick a box if they are good-looking — and they do so honestly!) The site will give you lists of those candidates who are male, of those who are married, and those who are handsome. One way of finding what you want is to write out the names of the married men: you find there are sixty. That allows you to write out the forty names of the unmarried (not-married) men. Looking in these forty we find five who are handsome. That's a lot of work to come up with someone to have a drink with while listening to his life story. On the other hand you could look first for handsome people: you might find there are ten. If you write these out and check which ones are unmarried we get the same five guys. That is a lot less trouble.

>> how do you deal with sites that give more search results than you can handle? do you sometimes move from automated search to manual search?

This was an ugly bunch of men. If the proportions had been different the opposite procedure might have been the one that was easier. If we search mechanically, by brute force, we will often do more work than we need to. A little thought in advance will save time.

I have been illustrating searching in *sequence*, using one filter and then another. (I shall also sometimes say searching in series.) We can also search in *parallel*, applying each filter independently of the other and then combining the results. (I shall sometimes call this branching search.) This amounts to searching with OR. So in a different search we could collect the unmarried men, and also collect the handsome men, and then combine the two collections. We would be following the instructions:

Find all the individuals satisfying F

Find all the individuals satisfying G

Collect everything that results from either search

or equivalently

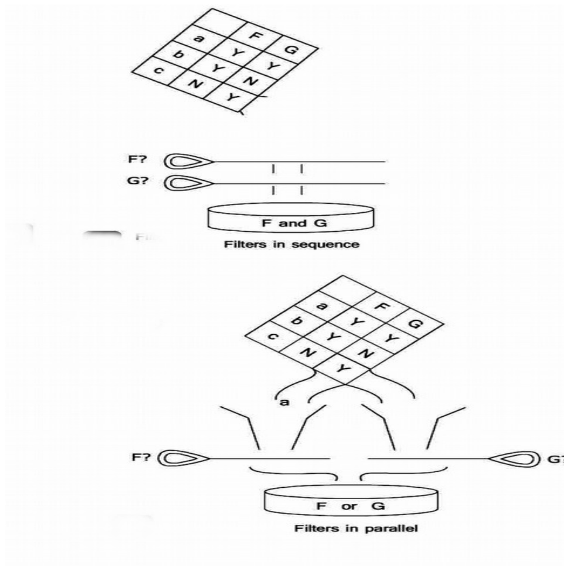
Find all the individuals satisfying F

Find all the individuals satisfying G

Take the *union* of the results

These two instructions do the same search, for F OR G. They are different from the sequential search for F AND G described earlier.

The difference between filters in parallel and filters in sequence is important, and will return in later chapters when we discuss search trees and derivations. Here are images of the two kinds of filter.



2:5 (of 9) and/or/comma

The contrast between searching in sequence (or series) and in parallel, and between AND and OR, links to a common experience when searching the Internet. Many search engines have as their primary mode of entering a query a list of search terms separated by commas. "music, classical, cello", for example. Usually the comma functions — to a first approximation only — as AND. Then we get an ordered set of Internet sites in which all of the search terms occur. On some sites, though, the comma functions as OR. Then we get sites in which one or another of the terms appear. This use of the comma is becoming less common, since the Internet has grown so enormous that the set of sites where any of a list of even quite rare terms is found will usually be unmanageably big. (The rare word 'defenestration' means being thrown of a window, but once entering 'sex, defenestration' into Google I got 528,000 results. And Google claims that its comma means AND. And entering 'axolotl, defenestration' I got 17,800 results. All those pages about throwing amphibians out of windows!)

>> how would you test whether comma, used in a search engine's entry box, is closer to AND in its effect or to OR? see also exercise 5.

Here too the AND/OR contrast is a series/parallel contrast. If a search engine interprets comma as AND then each search opens a window in which items are listed which satisfy the criterion "search_term1 & search_term2 & ... ". But often we want to do an OR search instead. One reason could be that we are not sure which of two variants on a criterion is more likely to have the result we are looking for. For example, we might be torn between searching for "Louis the fourteenth", "Louis quatorze", or "Louis 14", for an essay on French history. (Or between "Sun Yat Sen" and "Zhongshan" for an essay on the influence of that leader. Or between *Title, pdf*, and *Title, epub*. There are many examples.) The solution is easy: open a separate window for each search, do it, and paste the results into a single file. Then you will get items satisfying the criterion "search_term1 OR search_term2 OR ... ". (You usually won't get all items satisfying this criterion, especially if you only copy the first page of each window, but the results will have what you want, a mixture of items containing the various search terms.)

>> give some examples from topics that interest you, where this parallel technique would be useful

>> money-making idea: write an AND-to-OR program that automates this procedure. see how studying logic can pay off.)

>> how would you use this procedure to search for
"(term1 AND term2) OR (term3 AND term4)"? (see exercises 20, 21.)

It is very useful to know whether a search site or program in which terms are separated by commas interprets the comma as AND or OR. For example if you are looking for a free download of a book and search with “TITLE, .epub, .pdf” on a site where the comma is

AND, it will exclude many search results which have one but not the other. But if the comma is OR, then this is an efficient way to survey the possibilities.

Lists can often be interpreted conjunctively, that is, as AND, and also disjunctively, that is, as OR. The OR interpretation usually includes more possibilities ("OR is MORE"). In everyday life we often state a list without being clear whether we mean it as AND ("all of the above") or as OR ("some of the above"). There is a very general reason why people are often unclear, in fact often confused, about this. It would take us some way from the topic, but see exercise **12** of chapter 7. (See also exercise 3 for this chapter.)

2:6 (of 9) but <u>if</u> it's a …

Here is a kind of search that is often useful. Suppose that you are moving to a new apartment, and you have decided to take along all your books, except for your science fiction collection, of which you are going to keep only the books by Philip K Dick. You are instructing a friend to put the books that are moving with you into boxes. You say "find all the cookbooks, and all the books about logic, in fact all the books, but if it's sci-fi it has to be by Philip K Dick." Your request is in the form of an *if* (it's a *conditional*, as logicians say): <u>if</u> a science fiction book gets included <u>then</u> it is by Dick.

The effect of this if-condition is to refine the search we have already made. We cut "all books" down to "all books except that if it is science fiction then it has to be by Philip K Dick". Consider the consequences for the choice of particular books. Your friend picks up a Shakespeare play and includes it, picks up a detective novel and includes it, similarly for a logic textbook, but when she picks up a science fiction book (instantly recognizable by their weird covers) she has to check who wrote it. What about books by Philip K Dick that are not science fiction? (For example his *Selected literary and philosophical writings*.) They should be included, since the rule applies only to science fiction books, whatever their authors. So we can find uses for a query that says “Find everything such that *if* it has attribute A then it has attribute B”. The effect of using this search is to include everything, *except* things that have A but do not have B.

Queries involving IF make most intuitive sense when they are combined with other queries. Suppose I say: (a) get all the books from my apartment (b) if they are by Philip K Dick then they must be sci-fi. Then you will know to get all books satisfying both the criterion that they are in my apartment and the criterion that if they are by PKD then they are SF. Books not by PKD but in my apartment get taken. But if I say “get me all books such that if they are by PKD then they are SF”, not in conjunction with any larger search, you are likely to respond with puzzlement. What about books not by PKD? In logic we understand IF even when it is in isolation in a way that is more common in everyday language when it is an extra proviso to another query or statement. So if a logician tells you to get all books such that if they are SF then they are by PKD, you roam the world picking up books by Shakespeare and Atwood and other authors, the Bible and the Koran, texts on logic and history and other subject, PKD's SF works, leaving out only the books — Asimov, Pratchett, Robinson, and others — that are SF but not by PKD. A tall order. Some of the exercises for this chapter are meant to help you get used to IF used in isolation like this.

>> suppose I say "if a book is by PKD then put it in the box", all alone, not joined to any larger instruction, and then when you go to the bookshelf there are no books by PKD, but lots by Tolstoy, Plato, Asimov and Stephenson. what would you put in the box?

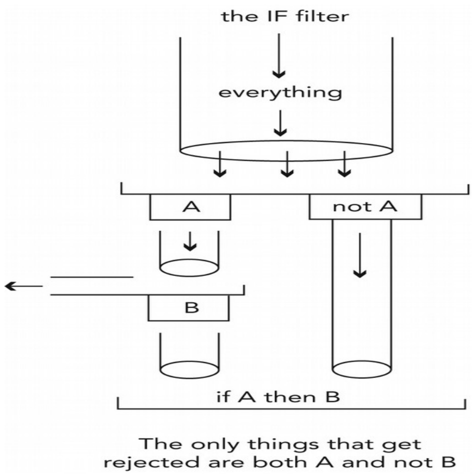
Suppose I am telling you to get socks from a drawer. I say "Get me the socks from the drawer: but if they are wool they (must be) red." The drawer has some red wool socks, so you bring them. It also has some green polyester socks so you bring them. It also has some blue wool socks and you do *not* bring them.

Suppose I say "Bring me what's in that drawer, with a proviso: if it's a sock it will be red." The drawer has some red socks, so you bring them. It also has some green shorts so you bring them. But you leave out the blue and yellow and black socks. You find a diamond ring, and you bring that along too. (So a diamond ring satisfies "if it is a sock it is red"!)

Suppose I say "Get everything such that if it is a sock it is red". The query is GET: IF SOCK THEN RED. So you get the three red wool socks, the two red polyester socks, the shorts, the diamond ring.

But suppose I say "Get everything such that if it is red it is a sock". Then the query is GET: IF RED THEN SOCK. So you get the three red wool socks, the two red polyester socks, the two green wool socks, the three blue silk socks, the diamond ring. We can see that these two queries are clearly different. It makes a difference which way round we take the *if.*

These searches too can be pictured as physical filters. An if-filter "if it is A then it is B" filters only A things. Anything that is not A gets through automatically. But A things are tested; if they are B they get through, but if they are not B they are blocked. We can picture it as below:



>> anticipating a topic from later in the book: how can you do an IF search using AND, OR and NOT? does this suggest how to apply the "open another window" technique to IF searches?

2:7 (of 9) the if of logic

We mean many things by “if” in everyday language. In logic we fix on one simple meaning, as explained in the previous section. (If-sentences are called "conditionals". The "if" of logic is sometimes called the material conditional.) To help make it seem natural consider the following.

Alice tells Bill and Carrie “if it’s raining tomorrow, be sure to wear a hat”. It is not raining tomorrow. Bill wears a hat and Carrie does not. Which one has followed Alice’s instructions? Logicians say: both.

Aiko makes two predictions “if it rains tomorrow, Bojia will wear a hat” and “if it rains tomorrow Cho will wear a hat”. It does not rain the next day. Bojia wears a hat, and Cho does not. Which of her predictions was true? Logicians say: both.

Artemisia says "Bring in stuff out of the trunk of the car, please. The only condition is that if it's a phone it must be a Samsung, since my sister's iPhone should stay there." Bruno brings in a Samsung phone, and Carlo brings in a used bubble gum. Which of them has followed her instructions? Logicians say: both.

The last of these examples uses the search command "GET: if A then B" familiar from the previous sections. (I could have said "Find" instead of "Get".) The other examples show that it fits with a general attitude to the word "if". The two themes are (a) "if A then B" is always true or false, never in between or neither, and (b) when A is false, count "if A then B" as true.

One feature of "if" understood this way, is that, as pointed out above, it is asymmetric: "If A then B" is not the same as "if B then A". If Artemisia had said, telling Bruno what to bring, "but if it's a Samsung then it must be a phone", and Bruno had brought in the iPhone, he would have been following her instructions, though when she says "if phone then Samsung" this is just what she does *not* want. This contrasts with both "and", and "or", both of which are symmetrical. "Get me everything that is a Samsung and a phone" is the same as "get me everything that is a phone and a Samsung"; "list all the days where it is either raining or Bill is wearing a hat" is the same as "list all the days where either Bill is wearing a hat or it is raining". Because "if" is asymmetric, we sometimes need a way of saying that we mean *if B then A* rather than *if A then B*. We use "only if" for this. We say "I'll be happy only if Robin comes to the party", meaning that if Robin does not come to the party then I will not be happy. We could also say "if I will be happy then Robin will have come to the party", but since this is awkward to say we prefer to use the "only if" construction. (There is definitely something confusing about *if* versus *only if*. Exercise **24** is meant to give more familiarity with this. Linguists tell me that there are languages which have the same word for *if* and *only if*, so that one has to consider the context in which the word is used to know which meaning it has.)

>> on a Venn diagram shade in the areas corresponding to "If A then B" and "IF B then A". where do they overlap, where do they differ?

2:8 (of 9) making new predicates

When we search we get a collection of individuals, those that have (or satisfy, as we often say) the criterion we used in the search. This gives us another attribute. For example if we search in the actors database for "**G**lamorous & NOT **M**ale" we could write the result with another column as follows:

	Glam	**M**ale	**G**lam **& NOT M**ale
philip	NO	YES	NO
juliette	YES	NO	YES
angelina	YES	NO	YES
tom	YES	YES	NO
natalie	YES	NO	YES
dominic	YES	YES	NO
cerris	YES	NO	YES

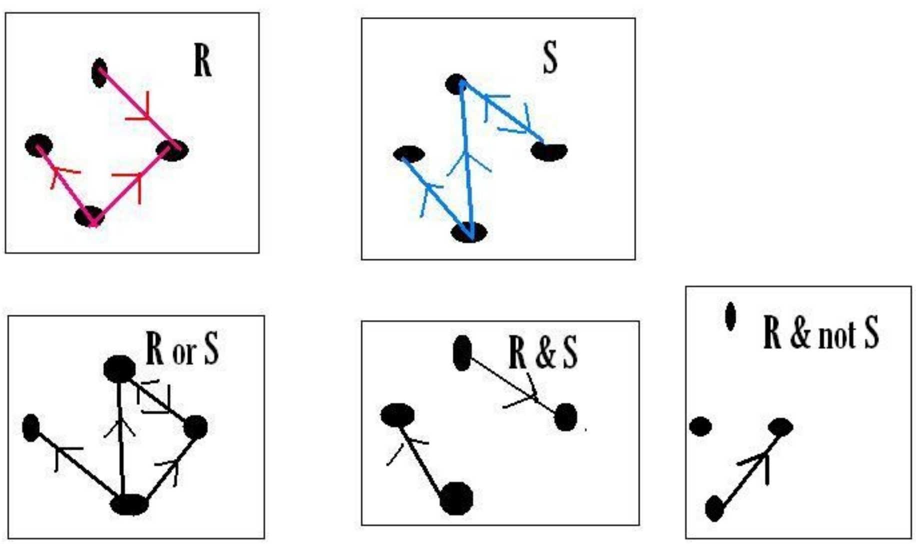
>> don't take my word for it that this is the right column of YESs and NOs. check it.

It is important to see that the result of a search can be a 1-place attribute even though the search criteria are expressed in terms of a 2-place relation. And the result can be a 2-place relation even though the criteria are in terms of attributes. (More generally, searches can result in changes either to a greater or a smaller number of places, whatever we start with.) For example, we can search for all pairs such that the first is glamorous and not male and the second is not glamorous and male, thus getting the relation that holds between two individuals that we might informally state as "she looks even better when she's compared to him".

>> other ways of putting this relation into loose language?

In the next chapter we will see a better way of stating these queries, so it would be a waste to spend time finding the right terms to express them now. For now, and setting us up for that, here are some visual representations of searches with relations.

Begin with a domain of four objects, with two two-place relations, R and S between them, as shown below, with the red arrows for R and the blue arrows for S.



Now we draw, in black, arrows to join individuals related by (a) either R or S (b) both R and S and (c) R and not S.

>> there is a double arrow that turns into a single arrow. why?

What you have drawn is the arrow diagram of the results of these three searches. They are all two place relations got by combining the two 2-place relations. There is an important point: the result of a search for individuals gives the set of individuals that *fit or satisfy* the criterion — for example which are both large and angry, or which are either large or angry but not pregnant — and this can define a new attribute. (We could invent the word "langry" for individuals which are large and angry.) And the result for a search for pairs of individuals can define a new relation. (We could invent the word "bangrier than" for the relation between individuals when one is both bigger and angrier than the other: bad news.) This process is important in the idea of mathematical structure, where we find in the facts about one topic parallel the facts about another. For example if we "lose" some of the details about the objects |, ||. |||, … and about a series of penny coins, we find that the relations between each of these have a lot in common, thus suggesting the idea of numbers and counting. (This is the idea of abstract structure. When relations have the same abstract structure they can be studied using the same mathematics.) Exercise **30** is meant to make this vivid for you.

>> why does losing some details from different databases bring out what they have in common? why does defining new attributes and relations help us to do this?

2:9 (of 9) many-place relations

Not all relations are two place, like "loves" or "tangos with" or "is north west of". There are also three place relations (and four place, and three hundred place.) The two place relation "… chases _" from the previous chapter, can be extended to the three place relation "… chases _ at time *" (Alice chases Caspar at noon). We also have the four place relation "… chases _ at time * in place $^{+}$". Alice chases Caspar at noon in the yard).

Often the individuals in a many-place relation are very different kinds of things, for example people, places, and times, which makes it easier to think about them. An example is "*Robin* kissed *Jo* for *8 seconds* in *New York* in *2002* near the intersection of *4th Street* and *10th Avenue*", a seven place relation! (*Who* kissed *whom* for *how long* in *which city* at the meeting of *which latitude* and *which longitude*.) Exercise 13 and 26 at the end of the chapter involve many-place relations. As you might guess, this is to sneak into your minds something that will be useful later.

>> how many places can a relation have before we can only understand it by thinking of it as a combination of simpler relations?

>> does it have to be true that *Robin* kissed *Jo* for *8 seconds* in *New York* in *2002,* when *Robin kissed Joe for 8 seconds (sometime),* and *Robin kissed Joe in New York in 2002* are true? why did I ask this?

Often when we use a many place relation we simplify by not mentioning one or more of the things we are searching for. For example if it is the year 2002 and we are at the intersection of fourth Street and 10th Avenue we may just say "Robin kisses Joe for eight seconds". This point is connected to the difference between searching for individual things and searching for databases in which sentences hold. A simple example of this is given by attributes and times. We could say who was happy on two consecutive days with a grid as follows:

Happytimes	**m**onday	tuesday
albert	YES	NO
bertha	NO	YES

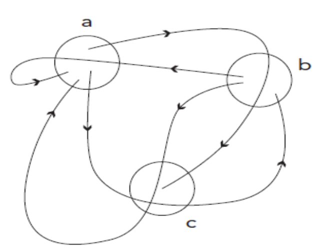
Or we could use a pair of tables:

happy-mon	**H**appy
albert	YES
bertha	NO

happy-tues	**H**appy
albert	NO
bertha	YES

These really come to the same. But this shows that searching for databases and searching for individuals are closely related. For we can get the same information by using the query "find days such that Albert is happy on those days" or by using the query "in which table does 'Albert is happy' hold?"

The diagram below shows a way of making an arrow diagram for a 3-place relation, using circles instead of dots for individuals, so that an arrow passes through three circles when the three individuals have the relation.



a fairly simple
3-place relation

There is also a way of depicting many-place relations that is not a generalization of arrow diagrams for two-place relations. Suppose for example we have the four-place relation "x smiled at y at time t in place l" (for example 'Mo smiled at Bo on Monday on the boardwalk", and we are using it to relate four people, three days, and three locations. Then we line them up in columns and draw lines (arrow heads are not needed) to make a diagram like the one below.

Smiler	Smilee	Time	Location
mo	**m**o	**m**on	**a**quarium
bo	**b**o		b**o**ardwalk
carol	**c**arol	**t**ues	mi**r**ror
dmitri	**d**mitri	**w**ed	**s**eawall

>> did Mo smile at Bo on Monday on the boardwalk? did Carol smile at anyone?

>> was there a mirror in the aquarium?

It is worth getting used to different ways of drawing many-place relations. Although the number of ways you can do it makes it not a topic for easily graded test questions, this does make it a useful skill when solving informal logic problems. They also make the idea of a many-place relation familiar in a way that will pay off when you think about the material in later chapters.

>> what are the advantages and disadvantages of representing relations this way, as opposed to using an arrow diagram?

words used in this chapter that it would be a good idea to understand (and ask if you do not) Boolean connective, Boolean search, complement, conditional, intersection, material conditional, parallel (branching) and sequential (series) search, n-place relation, query, scope, union, Venn diagram

exercises for chapter two

A – core

1) Explain the different things each of the following could mean. (All real quotes.)

(a) Take one tablet twice a day. (The alternative meaning is weird.)

(b) Living snakes are found on every continent except Antarctica, in the Pacific and Indian Oceans, and on most smaller land masses

(c) Take 2 tablets once a day.

2) (i) An angel appears on a mountain and says “O people, brush your teeth”, then disappears forever.

(ii) An angel appears on a mountain and says “O people, do not brush your teeth”, then disappears forever.

(iii) An angel appears on a mountain, says nothing, then disappears forever.

In which of these three cases has the angel told us to brush our teeth?

In which of these three cases has the angel told us not to brush our teeth?

In which of these three cases has the angel not told us to brush our teeth?

In which of these three cases has the angel not told us not to brush our teeth?

3) A notice from a car sharing company says:

"A: You may park for FREE in any authorized parking location provided it is in a:

1. Dedicated parking location for our company
2. Permit only parking spot in a residential area
3. Resident only parking spot in a residential areas
4. Area with no parking signage and no restrictions

B: When ending your trip always be sure that:

1. You are parked in an authorized parking location (see above)
2. You shut all doors, roll up windows, and turn off lights
3. Scan out with your member card on the windshield reader
4. Wait for the reader to say Trip Completed "

One of these lists is a conjunctive, AND, list and the other is a disjunctive, OR, list. Which is which?

4) Here is a relational grid where the five individuals and the relation are given just by letters. **(a)** Give *four* things it could mean which would make sense of the patterns of YES and NO, two about people and two about numbers.

R	**a**	**b**	**c**	**d**	**e**
a	YES	YES	YES	YES	NO
b	YES	YES	YES	NO	NO
c	YES	YES	NO	NO	NO
d	YES	NO	NO	NO	NO
e	NO	NO	NO	NO	NO

(b) What do you conclude from the fact that a database about people can also give facts about numbers?

(c) Which of these interpretations is likely to remain accurate when the domain (of people or numbers) contains many more than five individuals?

5) The comma separating the terms you enter in an internet search site sometimes makes an AND list (conjunctive), and sometimes an OR list (disjunctive). How could you test which it is on a particular site?

Note 1: playful approach: think of search terms such that it is very unlikely that any document will contain both.

Note 2: some search engines claim that their comma is always conjunctive. I doubt these claims. How would you test them?

6) Facts:

- Martha is 152 cm tall (just under 5 ft), Jurgen is 205 cm tall (approx 6' 8"), Sumiko is 180 cm, and Rosario is 165 cm.
- Taller people eat more ice-cream than shorter people (in this sample).
- The more ice-cream a person eats the less alcohol they drink.
- Evan is taller than Martha and shorter than Sumiko.
- Bo is the same height as Sumiko.

Questions: which of these people drinks the least? Do these facts determine which of Evan and Rosario is taller?

This is not a difficult problem. The point of it, though, is in this instruction: get the answer by using some of the facts to fill out a database from which the answer can be deduced. The database should be as economical as possible. That is, it should give the least amount of information possible while still being a complete database for the four individuals and giving an answer to the question.

Follow up: give a different interpretation, perhaps in terms of shapes or numbers, though it does not have to be, for the terms of the database and the answer. (Teaching suggestion: who can come up with the most interesting such interpretation?)

7) The second sentence of this chapter said "If you ask the wrong question you do not get the answer you wanted." Find two examples, from an internet search or from your library site, where careless use of "not" "and" and "or" can get a very different set of results from the one intended.

8) We have a collection of rocks, which can be enormous (boulders), big (rocks), average (stones), small (pebbles) or tiny (grains). **a)** In terms of these attributes define the 2-place relation "is bigger than". (Suppose for simplicity that all the rocks of each kind are the same size.) What else besides rocks would this definition work for
b) In terms of these attributes define the relation "is at least as big as". How is "at least as big as" different from "is bigger than"? Try to answer this in ways that will work with other series, such as metropolis/city/town/village/hamlet. It will help to draw arrow diagrams.

9) In a junk store there are:
100 cheap plastic spoons, 10 valuable silver spoons, 1 valuable silver plate, 1 valuable plastic lamp [rare kitsch: collector's item], and 5 cheap silver ear rings. Find:

a) all the things that are either plastic or not valuable

b) all the things that are both plastic and cheap

c) all the things that are both plastic and cheap and also not spoons

d) all the things such that if they are cheap they are silver [more usual way of saying this "everything, but if it is cheap it has to be silver."]

e) all the things such that if they are silver they are cheap

f) all the things such that if they are silver they are either cheap or plates

g) all the things that are <u>not</u> such that if they are cheap they are silver

10) You are searching a database for books satisfying various criteria.

Which of the commands (a) –(i) below would you use to search for each of (i) to (iv) below?

(i) the intersection of books by Tolstoy and books about bears

(ii) the complement of the union of books about bears and books about ducks

.

(iii) all books co-authored by Tolstoy and Shakespeare

(iv) all books by Tolstoy or by Shakespeare that are not about bears

(v) all books by Tolstoy about either Shakespeare or Tolstoy

(a) author = Tolstoy AND author = Shakespeare

(b) NOT (topic = bears AND topic = ducks)

(c) author = Tolstoy AND topic = bears

(d) NOT (author = Tolstoy AND NOT topic = bears)

(e) NOT (author = Tolstoy AND topic = bears)

(f) NOT (topic = bears OR topic = ducks)

(g) (author = Tolstoy OR author = Shakespeare) AND NOT topic = bears

(h) author = Ducks AND topic = Shakespeare

(i) (author = Tolstoy AND author = Shakespeare) AND NOT topic =bears

(j) (author = Tolstoy AND topic = Shakespeare) OR (author = Tolstoy AND topic = Tolstoy)

(k) author = Tolstoy AND topic = Shakespeare OR topic = Tolstoy

11) Which of these commands will get the same socks, however they are distributed between drawer A and drawer B? (There are only these two drawers, and all the socks are red, blue, or green.)

i) Get the red socks in A and the blue socks in B

ii) Get all the socks as long as they are not red socks in A

iii) Get the socks that are either red and in A or blue and in B

iv) Get the socks that are neither red nor in B

v) Get the socks that are both red and in A

vi) Get the socks that are both red and in A

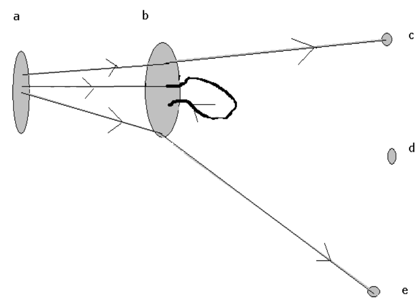
vii) Get the socks that are (a) bue or green and (b) in B

viii) Get the socks that are neither blue nor green nor in B

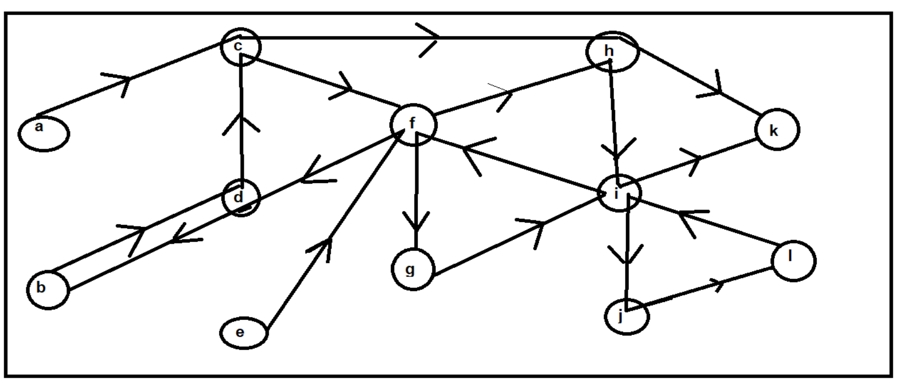
ix) Get the socks that are either (a) blue or green or (b) in B

12) Using the two arrow diagrams in section 8 of the chapter, draw new diagrams indicating (d) individuals that have R to at least one individual (an attribute) (e) pairs of individuals that have S either to i or to ii (a 2 place relation) (f) triples of individuals where the first has R to the second and the second has R to the third (a 3 place relation).

13) In the 3-place relation below, which of the following triples have the relation ("stand in the relation to one another"): **(c, b, a), (a, b, b), (, b, c) , (a, b)**?



14) The appendix to this chapter was about arithmetic, not logic. This is an exercise where following the strategy behind the suggestion in the appendix makes less mental strain in performing a complicated search. Consider the arrow diagram below for a 4-place relation **R**. (Diagrams like this were discussed in section 9 of the chapter.)



The task is to find individuals (from a, b, c, d, e, f, h, i, j, k, l) that occupy the first place of R when the second place is occupied by an individual that also occupies the third-place of some four individuals. That is, we are looking for individuals **l** such that there is another, **m,** where **Rlmxy** and **Rzsmt** — **x, y, z, s, t** can be any individuals at all. This sounds confusing, and if you thought it out anew for every individual when considering whether it fits the criterion you would get a headache. The idea is in fact simple, and illustrates how familiar language is rather clumsy at expressing simple but precise ideas. Instead of getting a headache, think first what you are looking for and fix that in your mind. In the arrow diagram all arrows are part of four-link chains. We want individuals **I** at the beginning of chains where the second stop of the chain — the individual that the arrow from **I** leads to — is occupied by something that is the second to last member of some chain. So you find beginnings of chains and check whether their first destination is also touched by an arrow whose next destination is its last one. When you find such a chain you write its first individual on your list. (Another way of describing it: you are collecting beginning points of arrow chains whose second points are also the second last of some other chain.)

>> most of the chains of arrows connect four individuals, as you would expect. but some of them go round in a triangle. how can this be?

That is not hard to do. But the important thing is to understand it just once. You fix it in your mind as a definite procedure: find chains, check second destination, check for contact with their destination, write or not. Do it now. (I have written the answer at the end of these exercises. But do it yourself first. No one will know if you make a mistake.)

B - MORE

15) "The cars in the lot are green and red or yellow." Give unambiguous English ways of stating the two meanings of this sentence.

16) Explain the two meanings that “I don’t want to become an architect” can mean. Give short conversations in which the sentence takes each of these meanings. One person says “So you want to become an architect. The other replies “No. I … ” and the “…” can show two different things that might be meant by “No”.)

17) Recent newspaper headline:

Sudan woman spared flogging for wearing trousers

What are the two meanings of this sentence?

Another headline:

Labrador and Newfoundland to boost number of moose hunting licences

What are the sane and the crazy readings of this sentence? (When I read the sentence it was the crazy meaning that came to me first, so I was puzzled.)

(Neither is a good English sentence. Newspaperese is even more subject to scope ambiguities than spoken English.)

18) In a used car lot there are:

Old Red Jeeps, New Black Saabs, New Black Toyotas, Old Red Saab, New Black Jeeps.

Find all the cars that are

Both Jeeps and black

Either old or not Toyotas

Both old and Saabs

Such that if they are red then they are Jeeps

Such that if they are Jeeps then they are red

Such that if they are red then they are not Jeeps

(You can assume that no car is of two brands or two colours or two ages.)

19) The student list for a course lists students by name, major, and year. Compare these two search requests:

(i) Find the names of all the students who if they are math majors are seniors

(ii) Find the names of all the students who if they are seniors are math majors

a) Which of these will get the names of senior history majors?

b) Which will get the names of junior math majors?

(Understanding "if" in the way explained in this chapter.)

20) You are searching for sources to use in a term paper on the ecology of Brazil. You have got a list by searching on the internet for "Brazil". But it has thousands of items. And some of them are about brazil nuts instead of the country Brazil. Which of the following would give a smaller list?

a) Brazil AND ecology

b) Brazil OR ecology

c) IF Brazil THEN NOT nuts

d) Brazil AND NOT nuts

e) Brazil AND ecology AND NOT (IF Brazil THEN nuts)

Which of these is the best choice for your paper? (This question assumes you are using a search engine which can do complete perfect Boolean search. It also assumes some common sense facts about files to be found on the internet.)

21) "Find everything that is green AND (round OR NOT smelly)"

Which of the following sequences of searches corresponds to the single query above?

(i) Find everything that is green AND round. Then find everything that is either among these or NOT smelly

(ii) Find everything that is NOT smelly. Then find another collection: everything that is round OR among these. Then find everything that is green AND in that other collection.

(iii) Find everything that is green OR not smelly. Then find everything that is among these AND round.

22) In the database given by the table below, which man attended on every day? On what day did no woman attend?

	Man	**W**oman	attended on **M**on	..**T**ues	..**W**ed	**..T**hurs	**..F**ri
abel	YES	NO	NO	NO	NO	NO	NO
beth	NO	YES	YES	YES	NO	NO	YES
charlie	NO	YES	NO	YES	NO	YES	NP
des	YES	NO	YES	YES	YES	YES	YES

What does this show about the meaning/ambiguity of the question "who attended every day?" and the statement "someone attended every day" ?

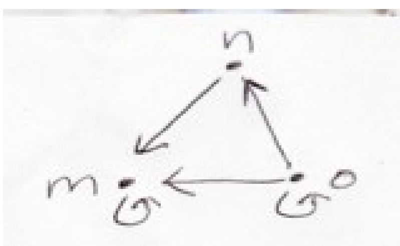
C - HARDER

23) List the individuals in the arrow diagram below such that

a) if they have the relation to anything then they have it to themselves

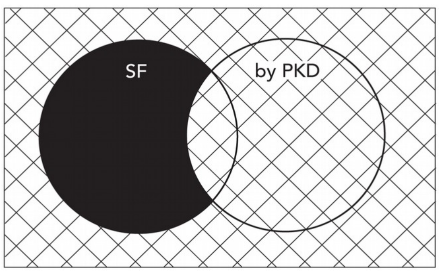
b) if they have the relation to **m** then they have it to everything

c) if they have the relation to everything then they have it to **m**.



24) In English sometimes when we say "if" we mean something symmetrical, namely, "if and only if". "A if and only if B" (sometimes written "A iff B") means "if A then B and if B then A". We often lazily say "if" when we mean "if and only if". (For example in definitions: we say "a shape is a triangle if it has exactly three straight sides", but we really mean "if a shape is a triangle it has exactly three straight sides, and if a shape has exactly three straight sides then it is a triangle.") You'll find, if you search, that I have fallen into this sloppy way of expressing myself at places in these notes. There is a Boolean operator that corresponds to "if and only if". Draw the Venn diagram for it.

25) You can get another hold on the logic meaning of "if" by thinking in terms of Venn diagrams (helpful for some people, but confusing for others.) In the diagram below the whole square represents books that you own, and the two circles are science fiction books and books by Philip K Dick. The area that has been cross-hatched represents all the books such that if they are science fiction then they are by Dick. Notice that it includes books that are not science fiction, books that are not by Dick, and books that are by Dick but not science fiction. What it does is exclude books that are science fiction but not by Dick.



The cross-hatched area is neither the union nor the intersection of the two areas it is defined by. Call it the *eclipsky* of the two: the eclipsky *from* one area *to* the other. So in this diagram it is the eclipsky from "science fiction" to "by Philip K Dick".[4] The eclipsky, and the philosopher's sense of *if*, is best thought of as a way of excluding some things: the eclipsky from A to B includes everything *except* things that have A but do not have B.

(a) what area is the eclipsky from "by PKD" to "science fiction"?

(b) why is this area different from the area for "science fiction iff by PKD"? (See question 19.)

26) Other ways of making graphical representations of 3-place relations are possible if

4The name "eclipsky" was invented by Susanna Braund in a moment of poetic inspiration, because it looks like the sky around an eclipse of the sun. The sky, note, *not* the darkened segment of the sun. Pronounce it "eclipse-sky".

we allow diagrams to be three-dimensional. Make a way of graphing a 3-place relation in three dimensions, so that for example it could be constructed out of Lego.

27) Which of the following are equivalent to which others?

a) If it chases mice then it is a cat

b) If it is a cat then it chases mice

c) If I get a pay increase then I will be happy

d) If I will be happy then I will have had a pay increase

e) It chases mice only if it is a cat

f) It is a cat only if it chases mice

g) I will get a pay increase only if I will be happy

h) I will be happy only if I get a pay increase

i) I will only be happy if I get a pay increase

j) I will only get a pay increase if I will be happy

(These look simple, but are confusing. I'm not sure why.)

28) How can we make Venn diagrams for four or more attributes?

29) The professors of four small advanced courses at the University of Nowhere give very different grades, as shown in the table below for the same 10 anonymous students. You are writing a research paper on the relation between study habits and grades, and you want to find A+ students and failing students in order to give them a questionnaire about their study habits. You are torn between two considerations. (a) some professors are very reluctant to let researchers see their grade lists — they give reasons about students’ privacy rights but you suspect they just cannot be bothered — and so you have a reason to study only those courses which will provide a good pool of subjects. But (b) your research will be taken more seriously if you pick your subjects at random, and so you have a reason to take all four courses equally seriously, not choosing one in advance.

	Anthroposophy470	Bioluminescence610	Casuistry900	Embrydontics550
student1	C	B	C	A+
student2	D	B	C	B+
student3	B	B+	B	B
student4	A-	C	B	B
student5	C	A	A+	A-
student6	D	A+	F	A+
student7	B	F	F	B+
student8	B	F	B	A
student9	B	A+	B+	A+
student10	B+	B	A+	A

Are there courses for which no data for the study can be found?

Does some course provide enough data to be used for your study?

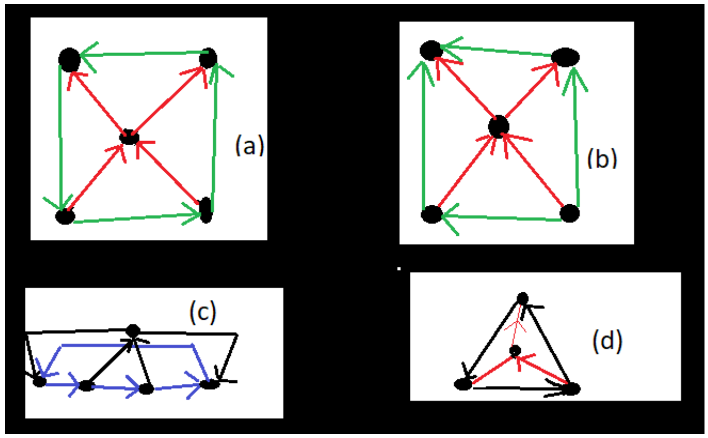
Are there courses with A+s and courses with Fs but no course with A+s and Fs?

Does every course provide data of both kinds?

How would you formulate searches to answer these questions? (The point here is more describing the searches than finding the answers.) Which answers would give you confidence that a strategy in accordance with (a) or with (b) was workable.

The table above is not a standard relational grid as we have defined them. How could it be rewritten as one? Are we dealing with a 1-place attribute, a 2-place relation, or a 3-place relation, or something else?

30) Each of these four arrow diagrams describes two relations, drawn in different colours. By eliminating individuals from some of them we can make them have the same structure as others. And by eliminating a relation from some of them we can make them have the same structure as others have if a relation is eliminated from them. Which relations (in which colours) have the same structure when we eliminate individuals, and which ones have the same structure when we eliminate a relation?



The individuals that the query in question 14 finds are a, e, f, g, l.

Boolean signs

everything's forbidden in Slovenia

knives, hearts, and branches

no and yes

nos and a yes

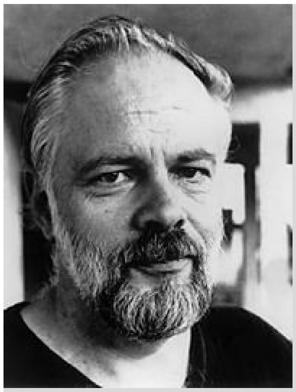
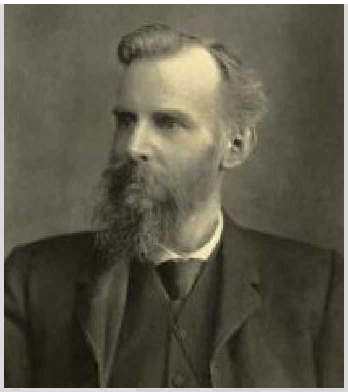
In all of these pictures a sign has a generally Boolean force. That is, you have to understand it in terms of NOT or AND. The bottom two say

NOT A and NOT B and C.

I think "knives, hearts, branches" says **NOT carving AND NOT breaking**, but someone might take it to mean **NOT (carving AND breaking)** .

Note that the last two do not actually forbid dogs without leashes. I think the sign-maker wanted to say **IF dog THEN leash**, but all she actually said was

OK: leashed dog. This is a topic for later chapters; see especially chapter 6 section 7 and chapter 8 sections 1 and 2.


which is Philip K Dick, which is John Venn, and which is the window commemorating Venn at the Cambridge college where he taught?

appendix to chapter 2: mental arithmetic as basic mathematical thinking

Here is a suggestion about one reason why children can find mental arithmetic difficult, sometimes leading to a lifelong fear of anything mathematical. (Studying symbolic logic sometimes shows people that although they are not comfortable with numbers they can handle formal, symbolic, ways of thinking. This appendix is entirely about arithmetic, but its themes are applied to logic in exercise 14.) In school we are taught to do arithmetic on paper. But when we try to transfer the routines we use on paper to a purely mental procedure, we find that we cannot hold the information reliably in our short-term (working) memories. For example in multiplying 23 x 32 on paper we write down

```
  23
  32
  --
  46
 69
 ---
 736
```

but when we try doing this in our heads we find it hard to keep the vertical arrangement fixed in our minds, jumbling which numbers are in which columns, and concentrating on it makes us forget the partial products, 46 and 69, before we can do anything with them. Keeping the spatial arrangement in mind gets in the way of doing the calculations. And one reason is trying to remember several multi-digit numbers as visual patterns — seeing them in your mind — at the same time as trying to keep the arrangement of the calculation in your visual imagination.

The solution is to use a different routine, to avoid the collision of two or more images. So think of 23×32 as (23 × 30) + (23 x 2). This is the "logical" aspect, so you want to keep it in a separate place in your mind from the calculations. One way is to remember three separate instructions (instructions, not images)

first: multiply   23 x 30
second: multiply 23 x 2
third: add these

Keep the three instructions separate in your thinking and do not do any calculations until you have them firmly remembered. But then do them. So you go: 23 × 30 is 690, 23 x 2 is 46. So add 690 and 46 to get 736. (I chose the example so as to be not too easy to do the traditional way and not too hard to do the variant way. Of course there are more challenging examples.)

You may find that even doing it this way the first product in particular has faded by the time you get to the addition. That is because you are trying to hold that as a visual image too, or the sound of the spoken number. Try thinking of it as the number it is. To do this you have to give the numbers characters, things to remember them by. 365 days in the year, 12×12 is 144, 4×4×4 is 64, 185 is a dollar and three quarters and a dime, and so on. Develop a collection of these so that they appear without effort. Then in the simple example above 690 is 10 less than 700, or 3×2, 3×3, 0, 10 times a sexual practice, or whatever else will stick in your mind. And 46 is three short of 7×7 or 8×8 backwards, or 1° more than halfway up. (Multiplication tables and the like are useful here, not as helps to calculation but as ways of giving numbers familiar faces. Whatever works for you.) This needs work in advance, over a period of time, but it is useful for many things, for example remembering addresses and phone numbers. (Alternatively, you might think of the numerals as spoken sounds rather than as written. You have to find what works for you.)

Mental arithmetic illustrates two related contrasts between mathematical procedures and those of every day thinking and communication. The first is the way that even in an elementary mathematical context the mental resources needed to perform a procedure and the resources needed to remember what it is that needs to be performed can interfere with one another. The second is the way that in every day understanding we can absorb a whole sentence, phrase, or more, and then go back in memory or reading and get clear about some details of. But in many presentations of mathematical ideas you have to get each detail 100% right before proceeding to the next. Bringing the whole thing slowly into focus is not an option. This second contrast sets up the first, because the accumulation of early details swamps the working memory needed to figure out the later ones. (This happens much less with spoken languages, for reasons that I would not claim to understand.) But we can minimize the problem. We can separate general descriptions from detailed instructions, to make it possible to digest the former slowly with successive approximations while performing the latter precisely when the time comes. And we can learn to understand mathematical statements so that we treat some parts as we would a story or a description of a scene, to be slowly absorbed, and other parts as instructions where we should snap to attention and follow them.

chapter three: Boolean Search

3: 1 (of 8) find x

A query has two parts: first an instruction to find individuals, and then the condition (criterion) that they have to meet. Often the criterion is complex, and then we have to be clear for example when we ask for people who are sad lovers whether we are looking for people who are lovers who are also sad, or people who are lovers and people who are sad, or pairs such that one loves the other and both are sad. In everyday language we make this clear, when we need to, by saying things like "find people such that each is a lover and each is sad" for the first, "find people such that each is a lover and people such that each is sad" for the second, and "find pairs of people (couples) such that the first loves the second and both are sad". (Of course since it is spoken language there are more meanings besides these.)

There are two important devices here: the separation of the search command from the criterion, and the use of variables — pronouns like 'they', 'each', 'it' in natural languages — to connect the command to the criterion as well as to structure the criterion. In this chapter and following ones we will express this by writing "**Find**" plus a variable — any letter would do but we will stick to letters from the end of the alphabet such as **x**, **y**, **z** — joined to a criterion in which the variables also appear. For example the three queries from the previous paragraph could be written:

Find x: Lx & Sx
Find x: Lx and **Find y: Sy**
Find (x,y): Lxy & Sx & Sy

(It is the first and the third that are really of interest.)

>> express some of the other possible meanings of "find sad lovers" in notation like this

These are really very close to some forms of everyday language. The first could be rephrased as "Find for me any one, such that he/she is a lover and he/she is sad". The "he/she" and the "one" are really the same variable, which English makes us use different words for in different places in a sentence. (In many people's spoken English we could find "Find anyone who is a lover and they are sad. "One", "who" and "they" are all doing the work of **x**.) The third could be rephrased as "Find (me) couples, a first and a second, where the first loves the second and that first is sad and the second is also sad." In **Sx** the variable **x** is said to be "free": it is available for referring to anything. And in **Find x: Sx** the **x** is said to be "bound", and bound by the "variable-binding operator" **Find x**. It is like the distinction between a bare pronoun "she", as in "she works with computers", which could refer Ada Lovelace (first programmer) or Grace Hopper (wrote the first compiler) or millions of others, and on the other hand pronouns in constructions such as "if a person works in high tech then she may change job frequently", or "Find me three female executives such that they earn more than a million a year." There are many variable-binding operators, and we will meet others.

>> what about pronouns that are linked to a particular name, as in "if Mary discovers you did it she will be furious." how do they fit in?

In moving to this way of writing down queries we are beginning symbolic logic, where the aim is to use symbols to express logical ideas as clearly as possible. Sometimes this is meant to replace our use of natural spoken language, and sometimes it is meant to augment natural language when we need to make a point or a distinction very clearly[5].

5The most important creators of symbolic logic were the philosophically-minded mathematician

Many philosophers and psychologists — but not all — think that symbolic logic can give insights into what happens in our minds when we think and use language.

In natural spoken languages pronouns, and the operators that bind them, are often special-purpose. We use "he" and "she" only for people, forcing us to choose between masculine and feminine, and we almost never use "it" for people.[6]. Instead of saying "find all individual locations x such that there is a bus stop at x", we say "where are the bus stops?". In logic we use the same variables and bind them with the same operators whatever the topic. Moreover search commands are not very sensitive to differences between singular and plural: we can equally well say "find all the phones costing less than $200" or "find each phone that costs less than $200". This gives advantages as we can state general patterns that apply to all topics. And it emphasizes grammatical features that vary less from one language to another.[7]

The main aim of this chapter is to say carefully what results various queries will get. To do this we have to state the queries precisely so it is clear what queries they are. This is the point of using symbols. But we want to state the rules in general, not the rules for

Gottlob Frege 1848-1925, and the mathematically-minded philosopher Bertrand Russell 1872-1970. Programming languages, such as Prolog or C++, are descendants of symbolic logic. An early programming language was called Ada, after Ada Lovelace. There are also languages for managing computer databases, of which the most widely used is SQL, and these have many features taken from symbolic logic.

6 If we introduce the pronoun *xe* with the *x* pronounced as in the pinyin spelling of Mandarin, then it sounds halfway between *he* and *she*. Wouldn't that be a good idea? And it fits the use of x as a variable!

7 Linguists describe languages as SVO, VSO etc. depending on the order of Subject ,Verb, and Object. (All orders are found among the world's languages.) English is SVO, except for some poetry and old sayings. (The first sentence of *Paradise Lost* is OVS.) Latin is generally a SOV language ("Fortune favours the brave"= *fortuna fortes juvat*); so are subordinate clauses in German, a holdover from an earlier stage of the language when whole sentences were. Logic resembles a VSO language (though it makes no distinction between subject and object.) This is simpler when dealing with relations holding between several objects.

any particular attributes and relations, referring to particular databases. I will use colours for this: when I write a symbol for an attribute in red I will mean that this is the pattern for any attribute symbol.)

Of course we can search for fewer than all the individuals satisfying a criterion. We can say "find one x such that ..", or "find eight x such that ...". We are not going to study these variant searches in any depth. (Searches for just one individual are discussed in chapter 4.) One interesting such search command, though, occurs when we search for individuals with some extreme characteristic, such as the maximum of a set of numbers, or the best looking of a set of people. The search then is "**Find x: Max$_S$x**", or "**Find x: for any y in S A**sAtractive**xy** ". I mention these not because we are going to use them at all, but because they are related to ideas that will appear later. (So think about them for a moment and then do not worry further about them.)

One **Find x** search is of enormous importance in the history of mathematics: algebra. Beginning in ancient Babylon mathematicians developed ways of solving riddle-like problems along the lines of "If my weight is added to three times my weight then it is twice my weight plus fifty kilos. What do I weigh?" Beginning about 1500 these were expressed as solutions to equations such as "x +3x = 2(x+50)". These solutions are answers to **Find x** queries where the domain is numbers. (The domain kept expanding, from positive integers to positive and negative, to real numbers, and then to complex numbers, to decrease the range of queries that have a null outcome.)

3:2 (of 8) Boolean queries

We can combine queries using the Boolean connectives AND, OR, NOT and IF. For the sake of brevity, and because it is traditional in logic, we abbreviate these with the symbol, **v**, **~**s **&**, and ⊃. [8] I discuss each in turn. I shall use some slightly weird English, in *italics*, which may help get the symbols to feel like language. (Don't learn the weird English: it's just there for the helpful effect it may have.)

negation: NOT

Find x: ~Px   Note the symbol ~ (called a tilde) used for "not". (You can think of **~** as someone shaking their head to say No.) For **P** we can substitute any predicate that has the variable x, for example **Ax**, or **Bx**, or as we will see **Ax & Bx**. (And similarly for, e.g. **Find y:~Qy** . If we were being hyper-rigorous we would have used dummy variables as well as dummy attributes.)

Suppose we have a database with four individuals who may or may not be **S**mokers.

	S
arthur	YES
basil	NO
cassandra	NO
dilma	YES

Then **Find x: ~Sx** gets **b**, **c**. All the NOs.

Find every person x such that that it is not the case that x is a smoker.

Find certain people. they do not smoke

>> this database has only four individuals. suppose it had thousands, and only two, as in this one, had **S**. what would the list for **Find x: ~Sx** look like?

conjunction: AND

Find x: Px & Qx

Note the symbol **&** (called an ampersand) used for "and".

---

8 There are other systems of notation, too many to list all of them. OR is always **v**, but the symbols for AND, NOT and IF (see below) can vary.

Suppose we have a database with two attributes, **S**mokes and **F**amous, and the same four individuals, as follows:

	S	**F**
arthur	YES	YES
basil	NO	YES
cassandra	NO	YES
dilma	YES	NO

Find x: Sx & Fx asks for everyone who smokes and is famous, namely **a**.

Find everyone such that s/he is a smoker and s/he is famous

Find those who are smokers and are also famous

disjunction: OR

Find x: Px v Q

Note the symbol **v** (called a "wedge") used for "or". (You can think of this as a fork in the road — this way *or* that, though you could take both — referring back to filters in parallel, or anticipating the using a forking lines of argument that will appear in chapter six.)

Applied to the Smokers-Famous database the query **Find x: Sx v Fx** gets **a, b, c, d**, the individuals with YES in the **S** column plus those with YES in the **F** column. (OR is more.)

Find everyperson such that that person smokes or that person is famous.

Find the people, x, satisfying "xe smokes or xe is an athlete."

Parallel things can be said for IF queries, but I'll give them the next section all to themselves. What we have said so far can be summed up in three rules

Individuals satisfy a negative query (**~**) when, and only when, they do not satisfy the negated criterion. **~** turns YES to NO and NO to YES.

Individuals satisfy a conjunctive query (**&**) when and only when they satisfy both conjuncts. & needs YES for both criteria.

Individuals satisfy a disjunctive query (**v**) as long as they satisfy at least one disjunct. v needs one YES among the two criteria.

These may seem obvious, trivial even. But I have worded them so they apply to a large range of cases. Consider a database with **S**mokers and **F**amous people, tables repeated below, and also a relation: some of these people are **J**ealous of others.

	S	**F**		**J**	**a**	**b**	**c**	**d**
arthur	YES	YES		**a**	YES	YES	YES	YES
basil	NO	YES		**b**	NO	NO	YES	YES
cassandra	NO	YES		**c**	NO	YES	NO	NO
dilma	YES	NO		**d**	NO	NO	NO	NO

Then the following searches get the results shown.

>> if you do not see why any of these is right, ASK.

Find x: Jbx xb gets **a, c**

Find (x,y): Jxy gets **(a,a), (a,b)**, **(a,c)**, **(a, d)**, **(b,c)**, **(b,d)**, **(c,b)**

Find (x,y): Sx & Fy & Jxy gets **(a,a)**, **(a,b)**, **(a,c)**

(see the remark on **&** and parentheses below)

Find x: Sx v ~Jxx gets **a, b, c, d**

(notice how although **Jxy** is a 2 place relation, we can do a **Find x** search by making the 1-place attribute **~Jxx** from it.) gets **c**, **d**

Find x: J

>> so do we really need one place attributes?

3:3 (of 8) a remark on Find x

This short section is meant to help you think in the symbolic terms that are used in modern logic. And I hope it can be useful in helping you appreciate mathematical forms of expression in general. There are similar sections scattered throughout the book. We have used the variable-binding operator **Find x** to state queries that in English would use the instruction “Find” (often “please find for me”, I hope) followed by a criterion picking out the things that are sought, which will often use a pronoun as a free variable, as in “can you get me news stories about the Mayor of Toronto: they should be less than two years old and they must have eye-catching photos.” Here the repeated “they”, referring back to “stories” is like a variable such as **x**. Variables and pronouns link the search command and its criterion while keeping them separate. We might want them linked but separate if we were doing something more complex than just writing down their names. For example the request “get all the striped shells out of the bucket, count them, and transfer them to the bag” we have the criterion “striped shells in the bucket” which is the target and the separate two instructions “count them” and “transfer them”.

A variable binding operator typically says to take the objects — or the objects in a given domain — satisfying a criterion and do something with them, such as just list them or put them in a bucket. In mathematics there are many such operators, asking us do such things as find the greatest (maximum) object satisfying a criterion, or to add or multiply the numbers satisfying a criterion. (I return to this theme in chapter 10, section 5.) One advantage of keeping the operator and the criterion separate is that it may be easier to think about one of them, typically the criterion, if it is distinct from the other. It can be easy then to see that the criterion is really very simple. Suppose we are looking for the individuals that are lazy and either friendly or not friendly — **Find x: Lx & (Fx v ~Fx)** —

in the table below. A cumbersome way of finding them would be first to collect the individuals that are YES for **L** — **b**, **c** — then the individuals that are YES for **F** — **a**, **b** — and the individuals that are NO for **F** — **c, d**. Then we combine the last two of these and take the intersection with the first. But if we consider the criterion as a whole we see that a simpler way of finding individuals satisfying the criterion is to look for YES in the **L** column on the same rows as either YES or NO in the **F** column. But everything is either YES or NO in any column, so this is just the individuals that are YES in the **L** column — **b**, **c**.

	Lazy	**F**riendly
arthur	NO	YES
baxter	YES	YES
cai	YES	NO
delilah	NO	NO

This also shows one of the advantages of writing the criterion using simple letters rather than full words. It is much easier to see what to do and which details are irrelevant. This is one of the reasons for mathematical notation in general. It is also related to what I will call the "outside in" method of evaluation in chapter five. A comparison with mental arithmetic may also help. It is like figuring out $\sqrt{(32)^2}$ (the square root of 32 squared). You could waste time starting with the 32, but if you start with the square root and the square nullifying one another you see right away that the answer is 32. Think of it as $\sqrt{x^2}$, where x happens to be 32, or in the strange English this book sometimes uses "The square root of the square of something. that something is 32". Or, even more simply, it is like adding 7+8. If (like me) you don't memorize arithmetic tables but recalculate simple sums every time, then the slow way — analogous to the first search method — is to go "plus one more" eight times starting with seven: seven, eight, nine, ten, eleven, twelve, thirteen, fourteen, fifteen. It takes a while, and it is easy to get confused. A better method — the method built into the traditional abacus — is to think in fives: 7+8 = (5+2)+(5+3) = 10+5 =15. If you know the sums of numbers less than five and know that two fives make ten, you can do this in an instant with no danger of confusion. The moral: look at the whole expression, not just the smallest pieces[9].

3:4 (of 8) the conditional: IF

It is useful to have one more symbol, besides **~, &, v**. The symbol ⊃ is used for "if", with the meaning that was explained in the previous chapter. We call it the material conditional; we will see a lot of it from now on. The query

> **find x: Sx ⊃ Fx**
> *Find every individual such that if that individual smokes then that individual is famous*

takes the whole set of individuals and excludes everyone who smokes but is not famous. So it gets the list {**a, b, c**}. This is the same as "either does not smoke or is an athlete" or, equivalently "such that s/he does not both smoke and fail to be an athlete".

>> do we need a special symbol for the material conditional, given that we can define it, both in terms of **~** and **v** and in terms of **~** and **&** ?

It is worth going slowly here. The rule for ⊃ is very simple. **Find x: Px ⊃ Qx** gets all individuals which either do not satisfy **P** or do satisfy **Q**. Or what is the same, all individuals that are not both **P** and not **Q** (everything *except* the things that are **P** without being **Q**.)

---

9 A story of the great mathematician Gauss, as a small child. His teacher asked the class to add up the integers from 1 to 100, hoping for a quiet hour, and was amazed when five minutes later little Gauss handed him this (in effect):

n = 1+...+100, 2n = (1+...+100)+(100+...+1) = 101x100, so n = 10100/2= 50,500 .

That seems simple enough. Confusion sets in, though, when we start thinking of ⊃ as "if". The ⊃ of logic does have a right to be thought of as a version of the subtle and slippery English word "if". After all, "if it's a duck it goes quack" is pretty much the same as any of the following "either it goes quack or it's not a duck", "either it's not a duck, or it is and goes quack", or "it can't be a duck without going quack". And all of them amount to "either not duck or quack" or "not both duck and not quack". (Why are these the same? Chapter five.) One difference is the amount of freedom or discretion, or open-ness to other considerations, that informal *if* as opposed to ⊃ , allows. Suppose I give you a big container of animals and say to go through it and for each thing if it is a duck, to kiss it. (Think of kissing as a kind of finding: perhaps you are marking things with lipstick.) You go through the container and when you find ducks then you, being a very obedient follower of my instructions, kiss them. But what do you do with the things that are not ducks, the frogs and slugs and puppies? If you are taking me to be speaking ordinary English you'll think I haven't told you what to do so you may feel free to make your own decision in each case. But if I give you a container and I say "kiss each thing satisfying (if it is a duck then it is black)", meaning "if" as the material conditional of logic, that is different. For each thing that you find in there you have to find out whether it satisfies the *if* sentence, and when it does you must kiss it. You'll kiss the frogs and slugs, since by not being ducks they qualify. You'll kiss everything except the ducks that are not black. No room for discretion.

We are less used to this scope relation between IF and FIND in ordinary language. We are confused by "FIND (IF A then B)" when we expect "IF A then FIND B". One advantage of the first, wide-scope, version found in logic is that we can repeat the IF. We can say FIND (IF A then (IF B then C)), and other similar things, just as we can say "FIND (A AND (B OR C))". This is one reason why handling IF in the way that the material conditional, ⊃ , requires, encourages logic-style thinking: it means having a single rule that applies to all cases including complicated ones. This helps bring hidden assumptions out in the open. The exercises and chapters that follow are sprinkled with ⊃. It's good for you, but you may sometimes have to pause and re-adjust.

>> what about conditional questions, predictions, and commands besides "find"? how do we normally interpret them?

Here are some conditional queries and their results. I'll repeat the database, so you can check it easily.

	S	**F**		**J**	**a**	**b**	**c**	**d**
arthur	YES	YES		**a**	YES	YES	YES	YES
basil	NO	YES		**b**	NO	NO	YES	YES
cassandra	NO	YES		**c**	NO	YES	NO	NO
dilma	YES	NO		**d**	NO	NO	NO	NO

Find x: Sx ⊃ Fx gets **a, b, c** (everything except what is **S** and not **F**)

Find x: Fx ⊃ Sx gets **a, d** (everything except what is **F** and not **S**)

Find x: Jdx ⊃ Sx gets **a., b, c, d** (since **Jdx** is false for all **x**)

Find (x,y): Jxy ⊃ Jyx gets **(a,a), (b, a), (b. b), (b,c), (d,a), (d,b). (d,c), (d.d)**

(all pairs except those where J holds in one order and not the reverse.)

Find x: Fx ⊃ Jxx gets **a** (everything except what is **A** and does not have **J** to itself)

Find x: (Sx ⊃ Fx) ⊃ (Fx ⊃ Sx) gets **a, d**

(everything except those for which **Sx ⊃ Fx** and not **Fx ⊃ Sx**.) Notice how you can get these from the first two results.

3:5 (of 8) clear scopes

When a query is written in these terms there should be no scope ambiguity. As long as we use brackets to make sure that **&**, **v** or ⊃ never connect more than two sentences and **~** never applies to more than one sentence, the scope is always clear. For example

Find x: ~ (Ax & Sx)
Find x: ~Ax & Sx

are different. The first gets everyone of whom it is false that they are both an athlete and a smoker. (*Find everyone such that it is not the case that that one is an athlete and that one is a smoker*.) The second gets everyone who is not an athlete and is a smoker. (*Find everyone such that that one is not an athlete and that one is a smoker*.) In the database in section 1 above, the first would get **h, t, w** and the second would get **t**. Contrast this with the English "find everyone who is not athletic and smokes", which could be interpreted either way.

Similarly

Find x: (Ax & Bx) v Cx

Is different from

Find x: Ax & (Bx v Cx)

Contrast this with the English "Find every thing that is A and B or C" which can be taken either way. (See exercise 9 for a search where these are different.)

It is not hard to give exact rules for using Boolean connectives so that the result is an unambiguous query. The main idea is that there should never be any doubt what sentences a connective is joining. ("Sentences" includes open sentences like "**Sx**", with free variables, as well as closed sentences like "**St**": both "he is a smoker" and "Toshiro is a smoker.") I will not give the rules here, though, as the idea is clear from examples, and I do something closely related in the appendix to chapter 5.

The following are good clear queries. ("Well-formed" as logicians say.)

Find x: (Ax & Bx) v (Ax ⊃ Bx)

(get the things that are either both **A** and **B** or if **A** then **B**.)

Find x: (Ax v Bx) ⊃ (Ax & Bx)

(if its either then it has to be both)

Find (x,y): (Ax & By) ⊃ Rxy

(get all pairs, provided that when the first is **A** and the second is B, the first has relation **R** to the second)

And the following are *not* well-formed. (They are "ill-formed".)

Find x: (Ax & Bx v (Ax Bx))	but **(Ax & Bx) v (Ax ⊃ Bx)** would be ok
Find x: (Ax & Bx v Ax ⊃ Bx)	but **(Ax & Bx) v (Ax ⊃ Bx)** would be ok
Find x: (Ax & Bx v Ax) ⊃ Bx))	but **(Ax & (Bx v Ax)) ⊃ Bx))** would be ok
Find x: (Ax v (Bx) ⊃ (Ax & Bx)	but **(Ax v Bx) ⊃ (Ax & Bx)** would be ok
Find (x,y): Ax & By ⊃ Rxy	but **(Ax & By) ⊃ Rxy** and **Ax & (By ⊃ Rxy)** would both be ok

>> "But some of these *look* well-formed." famous examples of English sentences that at first look grammatical, but which we find we cannot give meaning to, are Lewis Carroll's "'Twas brillig and the slithy toves, did gyre and gimble in the wabe", and Noam Chomsky's "Colourless green ideas sleep furiously". compare these to each other and to the examples above. both of these have English grammatical structure, unlike say "altogether elsewhere vast"[10].

10But this fragment can be continued to make an English sentence. In fact it is from a poem by W H Auden, whose ending runs "Altogether elsewhere vast / Herds of reindeer move across /

English is not quite as ambiguous as this may suggest. If you take care you can usually say something that can have only one meaning. But sometimes you have to take a lot of care, and the result is more complicated than what we normally say. One feature of English that reduces ambiguity is the fact that we not only have the single word connectives "and", "or" and "if", we also have the pairs of words "both …and", "either … or" and "if … then". We can use these to get the effect of brackets. So instead of the ambiguous "find everyone who is not athletic and smokes" we can say "find everyone who is both not athletic and smokes" for one meaning, and "find everyone who is not both athletic and smokes" for the other. And instead of "Find everything that is A and B or C" we have a choice between "Find everything that is either both A and B, or C" and "Find everything that is both A and either B or C". We use "then" together with "if" when leaving it out would make something impossible to understand. We never would say "Get me something such that if it is A if it is B then it is C". Instead we would say either "Get me something such that if it is A then if it is B it is C" or "Get me something such that if, if it is A then it is B, then it is C". (Would you actually really ever say either of these? Well, yes. "Get me a dessert, but if it is not in an insulated box then if it is melting it has to have some ice with it." "Get me a dessert, but if it melts if it doesn't have an insulated box then it has to have some ice with it." Think hard enough and you'll see that these are different.) Moreover in spoken language we use pauses, speech rhythm, and intonation to make our meanings clearer. These are not available when we write, which is one reason that written language tends to be more formal.

One last point before leaving these formal queries. **&** and **v** (conjunction and disjunction)

Miles and miles of golden moss / Silently and very fast." That's part of the appeal of poetry, the way it is often on the edge of nonsense.

should strictly only join two terms. But “Find everything that is both sweet and both cold and tasty” is the same query as “Find everything that is both sweet and cold and also tasty”. So I shall allow us to write

Find x: Ax & Bx & Cx instead of

Find x: Ax & (Bx & Cx) or **Find x: (Ax & Bx) & Cx**

And similarly I will allow

Find x: Ax v Bx v Cx

without any internal brackets.

3:6 (of 8) automated Boolean search

We can do a lot of precise searching using AND, OR, NOT. Searching using these is called *Boolean search*. Many search programs available on the internet or elsewhere provide the capacity to do at least some Boolean searches[11]. Very often this is part of an option labelled “advanced search” or the like. (See exercise 21.) Sometimes then you can do full Boolean search using AND, OR, NOT. (Warning: after you enter your search you often get some “sponsored” hits before the ones that are responses to your search. Ignore these; they are produced by money rather than logic.)

Boolean search routines are also available on the databases of many university libraries. But they are often kept somewhat hidden, and it can take some special knowledge to find them.

>> students are often grateful for help with their library's search functions, which are often presented more in order to give an impression that the library has a large collection than to help students find useful books and articles. naming no names. (I have worked

11In Google, Boolean search is an option under "advanced search". To get it, search for "Google advanced search" using Google or any other search engine and on the resulting page way down on the lower left there is a "use operators" option.

at one institution where many of the results of a naive search were not actually in the library.) so you may want to indicate to your professor that you would like a class directed at

- finding the Boolean or otherwise advanced search options on your library's site
- making the site confess where the books actually are
- how to use the advanced search options to make a reading list for a term paper

Once you find the advanced search page, it often looks something like this

search box for text	pull-down menu for title/author/keyword/
pull-down menu for AND/OR/NOT	
another text box	another title/author/keyword menu
another AND/OR/NOT menu	
another text box	another title/author/keyword menu
boxes for other options, such as Date, Language, format (book, article, recording etc)	

You

You might for example fill in the boxes as follows.

Middlemarch	title
AND	
George Eliot ("Eliot, George": depending on the site)	author
OR	
Mary-Ann Evans	author

Date: 1870-1880

Language: English

format : book

You might use this query in a large research library if you wanted to know if there were any editions of the novel *Middlemarch*, which Mary-Ann Evans published under the name of George Eliot, that appeared in her lifetime under her real name. (I'm sure the answer would be No. But see note 2 below.

Note 1: NOT on these sites usually means "AND NOT": if you enter information A in a box and next choose NOT before entering information B in the following box, you get (something like) A AND NOT B.

Note 2: These sites are becoming more and more idiot-tuned. (Not, I suspect, to accommodate students but to accommodate librarians.) One consequence is that their capacity to do real Boolean search is declining. Another is that the sites are becoming more Googlish, in that items that do not fit your criteria but which the program thinks might interest you tend to show up.

>> is real Boolean search more common where standards are higher? the library sites of most elite institutions are on the web. check out the sites of higher and lower-status institutions to see whether there is any correlation with the precision of the literature

searches you can do. (My conjecture is that the answer is No.) Note 3: Repeated AND/OR/NOT choices open up scope problems. How do we know that the search above will get "Middlemarch & (Eliot v Evans)" rather than "(Middlemarch & Eliot) v Evans"? The only way to be sure is to try some queries where you know the answers.

>> how would you find out what the implicit bracketing conventions on a Boolean search site are?

A library computer is a good place to work on the difference between AND and OR in searches. Suppose that you want to find all the books written by Margaret Atwood, and also all the books written by Stephen King. Think what characteristics each of the books you want to retrieve has. Is it written by Atwood and by King? No. Some of the books are by the one author and some by the other. So you want to search for all books which are either written by Atwood OR by King. (Author = Atwood, Margaret OR Author = King, Stephen.)

Suppose you did search with "Author = Atwood, Margaret AND Author = King, Stephen". Then you would get books such that of each one of them it is true both that it is written by Atwood and that it is written by King. That is, you would get books co-authored by them. (*The Handmaid's Vampire, The Shining Assassin, The Edible Sematary?*)

There are many commercial and academic sites with advanced search options. (See exercise 18.) "Advanced search" can mean just that one can enter queries in a more structured way than just as a string of terms separated by commas. But in some of them a degree of real Boolean search is possible. The qualification ""a degree of" is there because there are usually limitations. I list three.

First, Boolean connectives are usually written in capital letters. AND, OR, NOT, though sometimes "-" is used for NOT as well. Connectives need brackets, and there are usually limits to how many brackets, and so how complicated a search, you are allowed.

Second, NOT presents difficulties. Most of the terms we will use for queries apply to a fairly limited number of items, and NOT will pick out everything except the items in the criterion it negates. So it will usually generate many hits: be satisfied by many items in the database. And the internet is enormous. So if on the search box of a search engine you enter, "-cat" or "NOT cat" it is likely to reply "not found".(Which doesn't mean that it cannot find anything that fits your search. It means "get lost: too many responses".) On the other hand "cat, -siamese" will get responses.

Third, "IF" has a similar capacity to produce an overwhelming number of hits. Combine this with the tendency of our minds to get confused by it, and the result is that many search sites simply do not allow conditional searches, even though these are often what we need to get nearest to the criteria we have in mind. Very few sites have an explicit IF search option, and there are usually obstacles to indirect ways of formulating a conditional search. (I have just entered "-cat OR siamese" into a search box and got the response "Your search '-cat OR siamese' did not match any documents." A reasonable refusal to search, expressed in misleading language.)

>> it may sound as if I am saying "search engines treat you like an idiot and prevent you finding the documents you want". that would be unfair. give some reasons why the procedures of search engines are in the interests of most of their users. do these reasons extend to university library sites?

3:7 (of 8) search engines

Most search facilities on the internet meant for the general public, including most search engines, do not support full Boolean search[12]. In fact they usually give very much impoverished search capacities, compared to what one could find in a search program meant for use by database professionals. One reason is that they want to be simple and friendly-seeming. Another, perhaps more profound reason, is that there is such an enormous amount of material on the web that most simple searches will get more results than users can handle. (This factor is increased by the fact that web pages are not organized in a way that makes it clear whether a page meets a criterion. Search for "cat" and you get not just pages about cats but pages using the word "cat". You'll even get pages which say "'cat' has three letters".)

So the order in which the results are presented becomes crucial, and then it seems that a program that presents results in a good order makes Boolean structure less essential. There *may* be a very deep fact here, that the kinds of searching that stem from Boolean logic, and in fact from systematic logical thinking as we traditionally think of it, work best on comparatively small databases.

>> even if there is a deep fact here, my formulation is likely to be an over-simplification. why?

A common strategy for search engines, is for the user to enter a series of search terms,

---

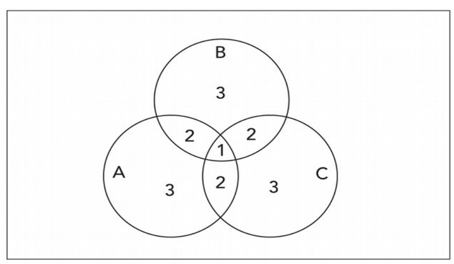
12 Two useful Internet sites with information about Internet searches are:
http://www.virtualsalt.com/howlook.htm
https://searchenginewatch.com/
on the latter, links to articles relevant to the issues in this chapter are mostly at
https://searchenginewatch.com/static/tips

separated by commas. The program then searches and presents result in an ordering. First come results which match all of the search terms, followed by results which match some but not all of them, followed by results which match even fewer of them. So for example if one enters terms A, B, C the first results are in the intersection of (items matching) A, B, C, thus ones that would come from

Find x: Ax & Bx & Cx. These are followed by results in the intersections of A with B, A with C, and B with C, thus from **Find x: Ax & Bx**, **Find x: Ax & Cx**, and

Find x: Bx & Cx. Lastly come results in the union of A, B, and C, thus from

Find x: Ax v Bx v Cx. (Of course the ordering must also avoid duplication between these, and any real search engine has many detailed tricks, some of them secret, to bring the results of interest to the average user to the top.)



Order in which a typical search engine presents results

So the rule is: first big conjunctions, then smaller conjunctions, then disjunctions.

>> search engines will sometimes claim that the comma is interpreted as AND alone. (this does not deny that the order the results are presented is independent of the Boolean interpretation of the comma.) how would you test whether this was true? (See also exercise 22.)

(Note that search engines will sometimes present results in a different order if one searches for "A, B" then they will with "B, A", although **Find x: Ax & Bx** should get the same results as **Find x: Ax & Bx**. I suspect this is the result of the commercial motives behind the details of the search algorithms.)

We can use these facts to get the effect of Boolean search on search engines directed at the general public. After all, we are more sophisticated people than the target clientele of most search engines, and we know a little logic. But it will help to know just a bit more logic than we do at this stage. I return to the question after we have discussed propositional logic, a topic that might seem to have little connection with search.

A basic reason for wanting to use a precise search criterion on the Internet rather than a vague and general one is that a too-general criterion will produce a very large number of results. The item you are looking for will probably be among these results but finding it will require another search. (This is the topic of the following section.) On most commercial sites, including Google and Amazon, the order in which the results is presented depends not only on the criterion you have used but also on payments between interested parties and the owners of the sites. As a result, if your further search is only among the items that are presented first, what you find will be influenced not only by the criterion and your searching technique but also by other people's financial interests. So careful searching, sophisticated logical thinking, can save you money and give some freedom as a consumer.

3:8 (of 8) extra: iterated searches

This is not a topic that we must cover at this stage. But it introduces ideas that will be useful later. So read it trying to get as much of the point as you can, but not worrying too much about the details.

Starting with a two place relation **R** we can describe a search as **Find x: Rxy** . This binds the variable **x**, leaving the variable **y** free. So we can put on another **Find**, binding the **y** to get **Find x: Find y: Rxy**. Our language for queries allows these: but how are we to understand them, what should they mean? We can give them a meaning that lines them up with central ideas of this course. Suppose for example **Rxy** is "x is to the north of y" and the domain is **g**lasgow, **l**ondon, **m**ontreal, **b**oston, **n**ew york. Then **Find y: Rxy** asks for a different search for each **x** in the domain. For **g** it asks for all the cities that Glasgow is to the north of, and so on for **l**, **m**, **b**, **n**. It describes five different searches, for cities that **g**, **l**, **m**, **b**, and **n** are north of. Write their results as (**g: l**), (**m: b, n**), and (**b: n**), where the first name shows which search in the range of **x** we are describing, and the following are names of the results of that search. Note that there are no entries for **l**, and **n**. That is because the **l** and **n** searches get no results. So **Find y: Rxy** gives five searches, and then **Find x: Find y: Rxy** asks for individuals **x** that are what we get when we do these searches. "Find those x for which you find y such that Rxy". It therefore has the result **g**, **m**, **b**.

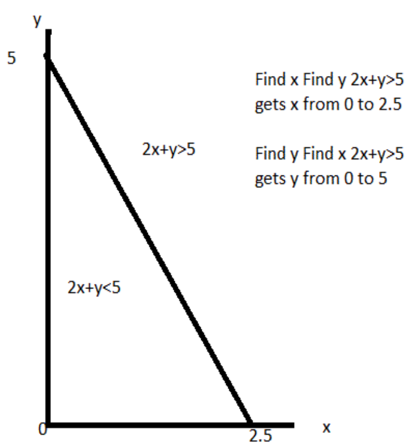
It is important to see that this query, **Find x: Find y: Rxy** , gives a different result to the query with the search operators in the opposite order:
Find y: Find x: Rxy. The first asks us to find x for which we find y such that **Rxy**, cities with cities they are north of, and that gives **g**, **m**, **b**. The second asks us to find y for which we find x such that **Rxy**, that is, cities for which we find cities (in the domain) north of them, and that gives **n**, **b**, **l**. The only city on both lists is Boston, because it is the only city that is both north of cities and has cities north of it. Seeing that these two queries give different results is the main reason for defining them in this way. (We could have defined them differently.) It begins an idea that will be useful in the final chapters of this book, so although we will not use it at this stage you should think about it until

you understand it.

>> how could we have defined them differently?

>> could someone argue that **Find x Find y Rxy** does not make sense as a query?

A different, more mathematical, way of thinking about the contrast between **Find x: Find y: Rxy** and **Find y: Find x: Rxy** may be helpful to some. It is simplest now to use a domain of numbers and a relation between them. So let the domain be the integers 0, 1, 2, ... and let **R** be "2x + y < 5". **R** corresponds to a region in the plane, of all points (x,y) where 2x + y < 5. See the diagram. We want to search to find out what points are in this region. So we perform a series of searches for various x from 0 and up, in each case finding y such that **Rxy** for the x we have chosen. (It's like using sonar to locate fish.) This reveals that, for positive x, $0 \leq y < 5$. So **Find y: Find x: Rxy** gets y from 0 to 5. This is exploring **R** "from below", with searches that vary along the horizontal axis. We can also search "from the side", asking for y such that searches for x find x for which — for that choice of y — **Rxy**. This reveals that, for positive y, $0 \leq x < 2.5$. So **Find x: Find y: Rxy** gets x from 0 to 2.5. The two results are not the same.



>> this example depends on the choice of domain. find a domain for which **Find x: Find y: Rxy** and **Find y: Find x: Rxy** are the same, using the same relation as in the example.

words used in chapter three that it would be a good idea to understand:

&, **v**, **~,** ⊃ Boolean connective, Boolean search, conditional, conjunction, criterion of a query, disjunction, **Find x,** material conditional, negation, query, variable-binding operator.

exercises for chapter three

A – core

1)

	Conservative	**P**rogressive	corr**U**pt
ralph	YES	NO	NO
stephen	YES	YES	NO
terri	NO	NO	YES
ulrich	YES	YES	YES

a) **Find x: Cx**

b) **Find x: Cx & Px** (see remark below)

c) **Find x: Px & ~Ux**

d) **Find x: (Cx v Ux) & ~Ux** (see remark below)

e) **Find x: ~(Cx & Ux)**

f) **Find x: Ux ⊃ Cx**

g) **Find x,y: (Ux ⊃ Uy) & (Uy ⊃ Ux)**

(This wants pairs (x,y) meeting the criterion. Ask yourself what the criterion is: what is the relation between x and y?)

Say in plain English what this query is looking for.

h) Find a search that gives {**r**}

i) Find a search that gives {**r**,**u**}

REMARK: section 3 of the chapter is relevant to **b)** and **d**).

2)

is **R**icher than	**R**alph	**s**tephen	**t**erri	**u**lrich
ralph	NO	YES	YES	NO
stephe**n**	NO	NO	YES	NO
Terri	NO	NO	NO	NO
ulrich	YES	YES	YES	NO

a) **Find x: Rxr**

b) **Find x: Rrx**

c) **Find x, y: ~Rxy**

d) **Find x: Rxu v Rux** (note that **u** is ulrich here, not a variable)

e) **Find <u>one</u> triple x, y, z: Rxy & Ryz & ~Rxz**

f) **Find x: Rxr ⊃ Rxu**

g) **Find** (all pairs) **x,y: Rxy & ~Ux & Uy** (For this one you have to combine the table above with the one in 1).

[hint: this question is made easier if you arrange the individuals in order from the richest to the least rich. We can do this because it is a 'transitive' relation, discussed in a later chapter.]

3) a) You are in a dispute about what is the highest mountain on earth, not in terms of the altitude of the summit but in terms of the height difference between base and summit. How would you search for an answer?

b) You want information about diseases that are incurable but not fatal. What search terms could you use.

c) You want to learn all you can about musical keys, and want to avoid information about keys for locks, etc. What queries can help?

d) You want to learn about explanations for UFO appearances, but do not want to waste time with craziness. How do you search?

4) Take this family tree

Kim Il-sung -- Kim Jong-sook
|
Kim Jong-il –- Ko Yong-hui
|
Kim Jong-un -- Ri Sol-ju
|
Kim Ju-ae

as a diagram of the three place relation **x and y are parents of z** . **(a)** Write a query in our formal notation which will capture the two place relation **x is a child of y**. **(b)** Write a query which in this database will capture the two place relation **x and y are spouses**. Why is its application specific to this database. **(c)** do the same for **x is a grandparent of y**. What does the database-specificity of (b) and (c) suggest about the limitations of Boolean search? **(d)** write a query which will capture the 2 place relation ("is a blood relative of"), assuming absence of inbreeding.

5) You are searching on the internet for kings of England who were also kings of Denmark. You enter these search terms: king of England, king of Denmark. The search contains some names of people satisfying both criteria, but you also get annoyingly many hits where one person is king of England and another is king of Denmark. (You find the same problem searching with ["king of England", "king of Denmark"] and with [king, England, Denmark] .) Say as clearly as you can what is going wrong. Suggest some ways of handling the problem, and state some of their disadvantages. Will wild-card symbols (*, +) help? Would the seriousness of these disadvantages depend on the purpose of your search?

6) Facts: Outlands is one stop due west of Supershop on the light rail system.

Suburparidise is one stop due East of Supershop on the LRS. Supershop is one stop due west of City Hall, which is one stop due west of Concert Place, Bad News is two stops due East of City Hall. There is also a north-south line which meets the east-west line at City Hall, and which goes one stop north, to Cañada Valley, and two stops south, first to Texada and then to Rio Grande. There are only these two lines.

a) Name two stations that are five stops apart.

b) Name four pairs of stations which are exactly two stops apart, but such that you cannot get from one to the other except by changing lines.

These are not difficult with a suitable diagram. Draw the diagram.

c) Does Cañada Valley have to be nearer to City Hall than Outlands is? What does your answer reveal about the diagram?

Follow up: give a different interpretation, perhaps in terms of shapes or numbers, though it does not have to be, for the terms of the database and the answer. (Teaching suggestion: who can come up with the most interesting such interpretation?) How are the relations in this interpretation similar to those in the light transit interpretation?

d) Draw the relations "one stop due east". "one stop north", and "two stops south" as arrow diagrams. Draw the relations "is east of" and "is south of" as arrow diagrams. How are these different? If you were drawing a diagram of the light rail system to solve a problem how would you keep it simple?

What about the relations "is at least as far east as" and "is at least as far north as"? How would their arrow diagrams differ from the others? What about "is one stop west of"? State informally how "is east of" is different from "is west of"and how "is one stop east of" is different from "is east of". How is "is one stop east of" different from "is the same or one stop east of”? (We come back to this in a later chapter.)

7) Given these tables

	in North **A**merica	In **E**urope
edmonton	YES	NO
glasgow	NO	YES
juneau	YES	NO
oslo	NO	YES
petersburg	NO	YES

is to the **N**orth of	**e**dmonton	**g**lasgow	**j**uneau	**o**slo	**p**etersburg
edmonton	NO	NO	NO	NO	NO
glasgow	YES	NO	NO	NO	NO
juneau	YES	YES	NO	NO	NO
oslo	YES	YES	YES	NO	YES
petersburg	YES	YES	YES	NO	NO

a) **Find x: Ax & Nxe**

b) **Find x: Ex v Nex**

c) **Find x: Ex v Nxe**

d) **Find x: Ax ⊃ Ex**

e) **Find x: Ax & ~Nxe**

f) **Find x: ~ (Ax & ~Nxe)**

g) **Find x, y: Nxy & Nyj**

h) Find all the cities that are to the north of some city in North America

i) Find all the cities that are to the north of all cities in North America

j) Write a query that in the tables above will get the set **{e}**

k) Write a query that in the tables above will get the set {**p,j**}

[the hint for question 3 may help here too.]

l) Find a x in North America such that x is north of all other cities in North America.

Find a y such that y is to the north of all other cities in Europe. Find a z such that it is to the north of y and not to the north of x.

8) "Get me all the socks that are green and wool or polyester" is an ambiguous instruction. Using **G** for "green" and **W** for "wool" and P for "polyester" write in the **Find x** notation the two instructions it can mean.

Do the same for "get me all the socks that are not polyester and green".

9) In the database

	A	**B**	**C**
a	NO	NO	YES
b	YES	YES	NO
c	NO	NO	YES

what is the difference in search results between

Find x: (Ax & Bx) v Cx

and

Find x: Ax & (Bx v Cx) ?

10) We can throw together the symbols for defining queries in crazy ways, so that they don't describe any search that could be attempted. Logicians call a string of symbols that has a meaning "well formed", or a "well formed formula" (otherwise it is "ill-formed"). I have not given any rules for well-formedness, but you should be able to tell which combinations of symbols make sense. (Comparison: "There are three mice in the bread-basket." is a well-formed English sentence, while "Mice there are-bread basket." , ".-.basket mice", and "mice basket bread" are not.) Most ways of throwing words randomly together don't make sense. But note that many good English sentences are ambiguous, while a well-formed formula is meant to have only one meaning.

Which of the following are well-formed queries?

Find x: (Ax ⊃ Bx) & Cx

Find x: Ax ⊃ Bx & Cx

Find x: (Ax ⊃ ~&Bx) & Cx)

Find x: (Ax ⊃ Bx v Cx & Dx)

Find x: ~(Ax ⊃ Bx) & Cx

Find x: (~Ax ⊃ Bx) & Cx

Find x: ((Ax ⊃ Bx) &) Cx

Find x,y: (Ax ⊃ Bx) & Cx

11) You are searching on your library advanced search page, for a book on ethics by a contemporary Australian philosopher called Smith. But you can't remember his full name or the title of the book. So you try with Author "Smith" and Subject "Ethics". How many books does this get? Too many. You can't see how to search for contemporary Australians, but you realise this rules out the great 18th century Scottish writer on ethics and economics, Adam Smith. How can you work this into the search? Try it. How much have you reduced the list? Then you remember that one word in the title was "problem". How do you include this? What do you get?

12) We can use "maximum" or "best looking", and similar ideas, in searches with a relation. Given a domain of people we can ask, for example "Find the best looking x such that for the best looking y, x is taller than y". To find this x, we must first for each x in the domain find the best looking y such that x is taller than y, and then we must choose the best looking of these. Describe a set of people such that "The best looking x such that for the best looking y, x is taller than y", understood this way, is not the same as "the best looking y such that for the best looking x, x is taller than y." Here as often, when we get precise we find that the order in which we say things makes a difference. (Suppose someone says "find the best looking pair such that one is taller than the other." Give several things they could mean. What about "find the best looking person who is taller than a best looking person".

13) This is a continuation of exercise 14 of the previous chapter. So if you did not do that one it would help to look back at it, mostly to get the purpose and spirit of this kind of exercise. One purpose of a formal notation for queries is to bring out the common structure of many searches that use different vocabulary. Both "find all the cats that are either black or not sick" and "catch the black cats cats and while you are at it those that are not sick" invoke the query **Find x: Cx &(Bx v~Sx)** . So if you find an efficient procedure for either it will apply to the other. A good way to find such a procedure is to consider a very simple database and find an easy routine for applying the query to it. So consider this database

	C	**B**	**S**
a	YES	YES	YES
b	NO	YES	YES
c	YES	YES	NO
d	YES	NO	YES

Apply the query to this database. (What does it get?) If you think about how you decided whether individuals satisfied the condition you will arrive at something like the following rule. "If **C** is NO stop and go to the next. If **C** is YES look for YES under **B** and NO under **S:** if you see one include the individual and go to the next, otherwise stop and go to the next." Do you see why this gets the right answers? Once you have formulated the rule you can apply it to more complicated databases. You proceed in this intellectual matter as you would in many practical matters: you think what to do, then you do it automatically, then if need be you think again. Try the quick searching routine with the following two databases. It is important to keep yourself moving through them quickly.

#1	**C**	**B**	**S**		**#2**	**C**	**B**	**S**
q	YES	NO	NO		**l**	YES	NO	NO
w	NO	YES	YES		**k**	YES	YES	NO
e	NO	NO	NO		**j**	YES	NO	YES
r	YES	NO	YES		**h**	NO	YES	NO
t	YES	NO	NO		**g**	NO	NO	NO
y	YES	YES	NO		**f**	YES	NO	YES
u	YES	NO	YES		**d**	YES	NO	NO
i	YES	NO	NO		**s**	YES	YES	NO
o	NO	NO	NO		**a**	YES	YES	YES
p	NO	YES	NO		**p**	YES	NO	NO
a	YES	NO	NO		**o**	NO	YES	NO

B – more

14) You want a book co-authored by Zeno Vendler and Aristotle Chan on flu and polio viruses, but not their book on computer viruses. Which of the following commands is most likely to produce a small list containing the book?

a) author = Vendler, Zeno OR author = Chan, Aristotle AND topic = viruses AND NOT topic = computers

b) author = Vendler Zeno AND author = Chan, Aristotle AND topic = viruses OR NOT topic = computers

c) author = Vendler Zeno AND author = Chan, Aristotle AND topic = viruses AND NOT topic = computers

d) author = Vendler Zeno AND author = Chan, Aristotle AND topic = viruses AND topic = not computers

15) Which of these commands in English (i to vi) is the same as which query in symbols (a to f)?

(i) Find all individuals that are C and D and not E

(ii) Find all individuals that are not both C and D

(iii) Find all individuals that are both not C and not D

(iv) Find all individuals, but if they are C they have to be D

(v) Find all pairs of individuals, the first member of the pair being C and the second D

(vi) Find all pairs of a C individual and an individual that is not D

Find y: ~(Cy & Dy) **b) Find x: Cx $\supset$ Dx**

Find x: Cx & Dx & ~Ex   d) Find y: ~Cy & ~Dy

Find xy: Cx & ~Dy   f) Find xy: Cx & Dy

16) Facts: Alirio is the father of Elizabetta. Marta is the mother of Alirio. Wei is the uncle of Elizabetta. Li is married to Wei. Wei is the father of Qiang.

a) Name two people who are cousins. Name one person who is a grandmother of Elizabetta. Name two people who are siblings.

b) Do the facts require that Li is the mother of Qiang? Do they require that Wei is the brother of Alirio? Do they require that Li and Elizabetta are not siblings?

In answering these draw a diagram of the relations between these people. What about the diagram has to be left unspecified so as not to say more than follows from the facts?

What definitions of relationship words do you need in order to use the diagram?

Follow up: give a different interpretation, perhaps in terms of shapes or numbers, though it does not have to be, for the terms of the database and the answer. (Teaching suggestion: who can come up with the most interesting such interpretation?)

17)

	Cynical	**D**epraved
mauricio	YES	YES
nessa	NO	NO
opheia	NO	YES

Find x, y such that x is **C**ynical and y is not **D**epraved

(all pairs where the first member is C and the second is not D)

Find x, y such that x is not **C**ynical and y is not **D**epraved

Find x, y: (Cx & Dy) v (~Cx & ~Dy)

Find all x for which there is a y such that: if y is Depraved, x is Cynical

(this last is meant to be a tricky one, though the answer is simple and there is a quick way to get it)

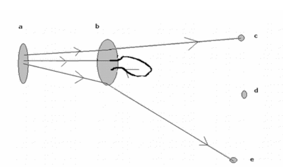
18) Go to the second hand book seller Abe-books (from Victoria BC – every student should know about them) at www.abebooks.com . Go to the "more search options page" and click to turn on the Boolean search option. Suppose you have seen a book by Helen Exley on paintings by cats. (Yes! It's hilarious.) You want to know if there are other books on cats painting. How do you use the search program to find out? You wonder if there are books on this not by Exley. Are there? Since cats paint, perhaps they dance. Are there books about cats dancing? Are there books about cats dancing and cats painting? What about cats and music? Books about cats and dancing and music? Cats and music and painting? In each case write down the search terms. For some of them, for example cats dancing and cats painting, there are two ways of phrasing the search.

19) Using the arrow diagram below, answer the following questions

(a) there are two individuals, x, such that Rabx. Which are they?

(b) there is one individual, y, such that Rayb. Which is it?

(c) there is one individual, z, such that Rzxx for some x. Which is it?



C – harder

20) Google claims that the comma "," in Google searches, is an AND. So "X, Y" should get just items satisfying both X and Y. I suspect this is not true, and that there is an OR element in Google's treatment of the comma. How would one investigate this? What searches would provide relevant evidence?

21) Not long ago search engines doing general searches of the web (as opposed to search functions on particular web sites) allowed explicit use of Boolean operators in searches, so one could search for expressions with forms like "(X AND Y) OR Z". Sometimes one could even embed one operator within the scope of another, so that there were searches such as "X AND (Y OR Z)" or "X OR NOT (Y AND Z)". These have become increasingly rare, and web sites such as "search engine watch" that give comparative information about search engines give a confusing picture. Can you find out where Boolean search survives, and whether embedded searches are possible on any search engine now? (Search engines can have un-obvious ways of doing such searches, and they often have more options available under 'advanced search" or the like.)

22) Many search programs on particular web sites (as opposed to the all-purpose search engines of the previous question) use a "list with commas" format, and then, as discussed in section 6 of this chapter, the query "X,Y" will get items satisfying both X and Y, and also items just satifying one of them, usually with the former listed first. There are differences of balance, though. Sometimes there will be only a few AND results before the OR results begin. How can one test a particular program to see what its balance is? How, in fact, can one test to see whether the comma is AND, OR, or a combination?

23) Can you find a better way of making arrow-diagrams for 3-place relations?

chapter 4: searching for models

4:1. (of 5) models

We can search in a database for individuals satisfying some criterion. That has been the main focus in the preceding chapters. We can also search for databases that make some sentence true. When we do this we usually do not call them databases but models. For our purposes the two terms mean the same: models are just databases under a different name. But in this chapter we begin the transition to issues about logical consequence — the topic of chapter 6 and important from then on — where "model" is the standard term. The main purpose of this chapter is to get you to see that the issues about searching in databases for individuals, which we will now start calling models, are fundamentally the same as the issues about searching for models (databases) where sentences are true. Seeing this continuity in the topic is at least as important as understanding any of the details of this chapter.

The individual and attribute tables we saw in chapter 1, relational grids, and arrow diagrams are all ways of describing models. Any sentence is true in many models, infinitely many in fact. So finding all the models that make a sentence true would be asking a lot.

>> do you see why any sentence is true in infinitely many models? take a very simple model for "Flossie is angry" and make it a little bit more complicated. then make that one a little bit more complicated. do you see how you could go on doing this forever?

Nearly all the searches we have seen so far try to find all the individuals in a database/model satisfying some criterion. But we can also search for just one individual satisfying a criterion. We can look in the model and stop when we find something that fits. This is an easier task than finding everything that fits, and we will never have to find infinitely many things. But it is still sometimes difficult, especially with a complicated model or a complicated criterion. For a simple example, consider the model and the criterion below. We want to find some individual in the domain of the model satisfying the criterion of the query. To vary the terminology slightly, we can also say that we want to find an individual that the criterion is *true of*.

Criterion: **Ax ⊃ Bx**

Model:

	A	**B**
a	YES	YES
b	YES	NO
c	NO	YES
d	NO	NO

Remember that **Ax ⊃ Bx** is satisfied by all individuals except that when they are **A** they must be **B**. So if we go through the individuals looking for just one which satisfies the search the search, which by analogy we could write **Find some x Ax ⊃ Bx** , we will see that the first one, **a**, satisfies the criterion. We can stop after the very first step. This model was special in that it had all combinations of YES and NO individuals or its two attributes. We will see more models like this in section 3 below and in the next chapter.

That example was deliberately simple, to give the idea. But I was saying that when the criterion is complicated finding just one individual can be difficult enough. So here is an example of that

Criterion: **~((Ax v Bx) ⊃ (Bx & Cx))**

Model:

	A	**B**	**C**
a	YES	YES	YES
b	YES	NO	YES
c	NO	NO	YES
d	NO	YES	YES
e	YES	NO	NO

Going through the individuals in the model, we see that **a** satisfies **Ax v Bx** and also satisfies **Bx & Cx**. So it satisfies **((Ax v Bx) ⊃ (Bx & Cx)))**. And therefore it does not satisfy the criterion. **b** satisfies neither **Ax v Bx** nor **Bx & Cx**, so it does not satisfy **((Ax v Bx) ⊃ (Bx & Cx))**. (Remember that Ax ⊃ Bx applies to everything except things satisfying A that do not satisfy B.) So it does satisfy the criterion. We have found the one individual we needed that satisfies this criterion in this model. But finding it was quite a lot of work. You would not want to go through these considerations for all five individuals, let alone apply the criterion to a larger model in this way. Luckily, when it is models that we are looking for there is a simple way of *making* them, which will provide a model when one exists, and will show that there are none when no models exist.

>> we can say that an individual satisfies a criterion or that the criterion is true of the individual. these mean the same. "true of" is like" true": a criterion is true of an individual when the sentence you get by putting the name of the individual in the criterion is true. for example the criterion “is a fast horse" is true of Secretariat because "Secretariat is a fast horse" is true. what happens if we use two names for the same individual? are there individuals without names, and if so how can we modify the definition of "true of" for them?

Some terminology. I shall use "sentence" both to refer to queries, which ask what individuals there are satisfying a criterion, and to refer to assertions, which ask what models (informally, situations or circumstances) make them true. You can think of a query as saying "what is like this?" And you can think of an assertion as saying "how can this be?". As we have been writing queries, they separate the **Find x:** part from the criterion specifying what is to be found. Assertions are not usually presented in terms like this. But we could state queries using just a question mark, so that instead of **Find x: R**ed x **& S**ock **x** we wrote something like "Red socks: are there any?", though this would be less clear in some situations. And we could write assertions with something like a **Find** prefix, so that instead of saying "Alvin is wearing red socks" we would say "find situations in which Alvin is wearing red socks. (We can turn a criterion into a sentence by inserting a name. We can also do it by using a quantifier, which is rather like the Find x: prefix of a query. We will see quantifiers in chapters 9 to 11.)

(I shall also use the word "proposition" in later chapters, to mean the things that get joined by Boolean connectives in propositional logic. That will be explained when we get to it, but to avoid confusion I should say that "proposition" is used in a broader sense in some other books.)

Assertions are true or false *in* models. I will also say that models *make* assertions true or false. The criteria of queries are true or false *of* individuals in models (or pairs triples, etc. of individuals if the criteria have relations) I will also say that individuals *satisfy* the criteria of queries. (All this terminology has been used in earlier chapters, but it is as well to make it explicit here where it matters.) Truth and satisfaction are closely related: if a criterion is satisfied by an individual then the assertion you get by inserting a name of the individual into the criterion is true. For example, if Alvin satisfies "x is wearing red socks" then "Alvin is wearing red socks" is true. So we can slide between truth and satisfaction very easily.

In chapter 1 I pointed out that if we name databases after individuals then the line between searching for individuals and searching for databases becomes even more vague. The example there was a set of models, as we will call them from now on, each of which describes the attributes of the dogs of that chapter on a particular day of the week. Then instead of asking for models which for example make "Flossie is angry" true, we can search for days d which satisfy "Flossie is angry on day d". (What are we searching in? That could get complicated if we want to be precise: a larger model got by putting the individual models for each day together. We will not go down that route.)

>> suppose "the sun is shining" is said on Monday and is true. is "the sun is shining" true of Monday?

4:2 (of 5). making models

We want to make models where individuals satisfy the criteria of queries and where the sentences using names of these individuals are true. The very simplest criterion would be simply a single attribute, say **P**, which would be satisfied by an individual **a** when **Pa** is true. **P** is an atomic criterion, and **Pa** is an atomic sentence. It is true in a model if the individual a is in the domain of the model and the model attributes P up to it; in an individual and attribute table this will be when the cell Pa has YES. **~Pa** is also an atomic sentence, true in a model when the corresponding cell has NO. (But more complicated sentences, involving Boolean connectives resides **~** ae not atomic.) To see how we can make models, begin with an atomic sentence **Pa**, consisting of just one attribute applied one individual. There is always a model for such a sentence, given by the table with one row for **a** and one column for **P**, with YES in the cell where they meet. It is just as simple to make a model for **~Pa**. Again it will have one row for **a** and one column for **P**, but this time it will have NO in the cell.

We can build sentences by combining atomic sentences with the Boolean connectives **&**, **v**, ⊃ , and **~**. For each of these there is a way of finding a model for it if one exists. (Or, what really comes to the same, of finding a model with individuals satisfying the criterion that is made by combining atomic criteria in the same way, but to keep it simple I will talk about sentences.) Consider first the conjunction A & B. (A might be for example **Pa** and B might be **Qb** , so that A & B would be **Pa & Qb.** But A and B might also be more complicated combinations of atomic sentences. I shall use red letters when I want to show something that applies to all sentences, however they are constructed.) A model for this sentence is given by a table with two rows, one for **a** and one for **b**, and two columns, one for **P** and one for **Q**. Write these one above the other, below the sentence we want a model for, as follows:

the conjunction rule

<u>A & B</u>
A
B

This tells us that to make a model for A & B we take a model for A and a model for B and combine them into a single model. This may not be possible, for example if A is **Pa** and B is **~Pa**. But the procedure as a whole will tell us whether it is possible, so just wait. If the sentence is an assertion then A **&** B is true in the model. If the sentence is the criterion of a query then A **&** B is true of some individual or individuals in the model. (I am suppressing a detail, since the aim of this chapter is to make the link between finding and assertion. See exercise 16. In the rest of this chapter I will ignore the difference between truth and satisfaction, except for the occasional hint that both are in the picture.

>> why is the **&** not in red?
>> why can models for **Pa** and for **~Pa** not be combined a single model making their

conjunction true?

A model for the disjunction A v B gives us two possibilities. The sentence is true if the model makes A true, and also if it makes B true. We have more flexibility if we keep both possibilities open. To show that both are possible models, we write this as follows:

The disjunction rule

```
 A v B
  / \
 A    B
```

This tells us that there are two models that we can use to make A **v** B true. One is a model for A and the other is a model for B.)

>> the "OR" of logic is inclusive: it applies as long as one of the disjuncts applies, including when both disjuncts apply. why does this rule not make disjunction exclusive?

A model for the conditional A ⊃ B also gives us two possibilities. The sentence is true if the model makes A false, and also if it makes B true. (Remember what the material conditional means.) Again we have more flexibility if we keep them both open. We write this as follows:

The ⊃ rule

```
 A ⊃ B
  / \
~A    B
```

This tells us that there are two models that we can use to make A ⊃ B true. One is a model for ~A and the other is a model for B.)

The story I told for **~Pa**, turning YES into NO, applies only to the negation of an atomic sentence such as **Pa**. In general we need three rules, one for each of the Boolean connectives **&, v, ⊃**.

The negation rules

~(A & B)	~(A v B)	~(A ⊃ B)
/ \	~A	A
~A ~B	~B	~B

These rules make intuitive sense. For a conjunction to be false both of the conjuncts must be false; for a disjunction to be false it is enough that one of the disjuncts be false; a material conditional is false only when the antecedent is true and the consequent is false. There will be more background for these in the next chapter.

We apply these rules to a sentence over and over again until it cannot be broken into any smaller parts. The result is a branching structure, in which each branch is a mixture of complex sentences and the atomic sentences they are made from. The structure looks like an upside-down tree, where one or more branches are connected to what I shall call

the base of the tree, indicated by being underlined, which has the sentence we start with. For example, if we begin with **P & (Q v R)** we get the following tree.

```
Pa & (Qb v ~Pa)
     Pa                    (applying the conjunction rule)
  Qb v ~Pa
     / \
   Qb   ~Pa                (applying the disjunction rule)
         X
```

Note several things about this tree.

– It begins by applying the appropriate rule to the whole sentence at the base of the tree, in this case **Pa & (Qb v ~Pa)**. If the sentence is a conjunction as in this case we use the conjunction rule, if a conditional we use the conditional rule, and so on. The rule we use corresponds to what in the next chapter is called the central connective, so there will be more about it there. Then we apply the rules that apply to the sentences that result, and so on. This gives us the branches, in this case **Pa, Qb v ~Pa, Qb** and **Pa, Qb v ~Pa, ~Pa**.

– We cannot extend it by applying the rules (at any rate the rules we have seen so far) any further, since we have broken a sentence down into its atomic parts.

– It has two branches. Each of them has a combination of atomic propositions: **Pa** and **Qb** on the left branch, and **Pa** and **~Pa** on the right branch.

– The right branch is marked with a red **X**. It does not describe a model because it has a contradiction — **Pa**, **~Pa** — on it.

– The left branch does not contain any such contradictions. If we make a model for **Pa** and a model for **Qb** and then combine them, we get a model for the compound sentence at the base of the tree.

The simplest models for **Pa** and **Qb** are

model for **Pa**	**P**		model for **Qb**	**Q**
a	YES		**b**	YES

We can combine these into a model for the complex sentence just by sticking them together, to get

	P	**Q**
a	YES	YES/NO
b	YES/NO	YES

(I have marked some cells YES/NO because it does not matter whether they are YES or

NO: the sentence is true in the model whichever way we fill in the cells.)
This is what always happens when we apply the rules to a complex sentence. We get branches (sometimes as few as one, and often far more than two). If a branch has an atomic sentence and also its negation then it does not describe a model. We say that the branch *closes* in this case. But if it does not close — it does not contain an atomic sentence and also its negation — then we can construct a model for the sentence we started with.

The general recipe for making models for a sentence gives us a model for every unclosed branch of the tree. For every (a) atomic proposition or (b) negation of an atomic proposition on a branch we make a cell with (a) YES or (b) NO and we then put them together to make the model. The cells are independent of one another so that this is not difficult. I have not proved that we always get a model this way, if there are any models for the sentence we start with. You can either trust me on this point, or you can do exercise 18.

>> it seems so simple. what might go wrong?

4:3 (of 5) general patterns of truth and falsity

The abstract patterns of searches for individuals and searches for models are the same, where Boolean connectives are concerned. So we could have used trees like the ones we have been discussing to construct models/databases where there are individuals satisfying the criteria of queries. But we will concentrate on the abstract patterns, as summed up in the six rules, the ones that I have been writing using RED capitals. These show the patterns that connect whole classes of sentences and queries to classes of models.

An important concept, at this level of generality, is that of a *truth table*. Remember that in section 1 above I used a model that had all combinations of YES and NO for the sentences involved. We can always use such models, and when we are dealing with sentences in general we can ignore the details of the cells that make a sentence true, or a criterion be satisfied by individuals, in a particular model. All that matters for many purposes is the pattern of truth and falsity.

These patterns are often presented in terms of *truth tables*, which are most easily explained taking the Boolean connectives one by one. They are closely related to the tree rules above. (But we need only one rule for **~**, rather than the three **~** rules.) The simplest is the truth table for **~**. A standard way of writing it is as the table below (This is quite different from an object and attribute table. Do not think of them as the same sort of thing.)

P	~P
T	F
F	T

This truth table says that a sentence ~P is true in a model if and only if P is not true of it. (If P is true then ~P is not, and if P is not true then ~P is true.) This must be understood so that it applies both to sentences and queries. This also means that a sentence is false in a model, or the criterion of a query is false of an individual in a model, if and only if its negation is true, or is not satisfied by that same individual.

Now the truth table for v:

P	Q	P v Q
T	F	T
F	T	T
F	F	F

This truth table says that a disjunction is true (of an individual in a model) if and only if at least one of the disjuncts is true. In other words, the disjunction is false only when both disjuncts are false. The connection with the tree rule for disjunction is easy: the disjunction is true in two cases, the case where the first disjunct is true and the case where the second disjunct is true. Provided that one or the other of these applies , then the disjunction is true.

The truth table for conjunction is

P	Q	P & Q
T	T	T
T	F	F
F	T	F
F	F	F

This truth table says that a conjunction is true (of an individual in a model) if and only if both of the conjuncts are true. It is false in all other cases. The connection with the conjunction rule is that a model fits a conjunction under only one condition, when both conjuncts fit it. And when both conjuncts do fit the model then the conjunction fits it.

One last Boolean connective is left: the material conditional. Its truth table is

P	Q	P ⊃ Q
T	T	T
T	F	F

F	T	T
F	F	T

This truth table says that a conditional is true (of an individual in a model) in all cases except when its antecedent is true and its consequence is false. The connection with the conditional rule is that in all the cases where the conditional is true either the antecedent is false or the consequent is true. And the other way around when either of these conditions is met in a model the model makes the whole conditional true.

Truth tables will return in the next chapter. For now, the point is how much sentences and queries have in common. We can make truth tables explaining the Boolean connectives for both of them; we can define search trees for both of them; and we can construct models for both in parallel ways. The resemblances go deep, and in fact the differences are superficial.

>> do truth tables really explain what Boolean connectives mean? suppose that someone did not understand a connective: would the truth table allow them to understand it? is there more of a problem with some connectives than others?

4:4 (of 5) models for multiples

Search trees as defined in section 2 constructed models for single sentences. But it is easy to adapt them to make models for collections of sentences. The same rules apply, and we can apply them to any sentences above a sentence on a branch, writing the result beneath it. For a very simple example consider the tree that begins with the two propositions **P** and **P ⊃ Q**.

P
P ⊃ Q
**/ **
~P Q
X

The second line is got from the conditional **P ⊃ Q** using the conditional rule. We do not need to use a rule for **P** since it is as simple as it can get. But the result is that the left branch closes, since it contains both **~P** and **P**. That means that the only models for the collection {**P**, **P ⊃ Q**} are models that make **Q** true. Interpreted in terms of sentences this means that all models for this pair of sentences are models for **Q**. In chapter 6 we will describe this by saying that **Q** is a logical consequence of the pair of premises **P, P ⊃ Q**. Interpreted in terms of queries it means that any successful search using these two queries will find results which satisfy **Q**. For example, if we search with both **Find x: Cat x** and **Find x: Cat x ⊃ Black x** we will get black cats, and thus only black things (not *all* black things, of course.). This makes sense as we can think of the first query as gathering everything with the proviso that if it is a cat it has to be black, and think of the second query as adding to this the requirement that we only collect cats. The result is black cats.

For a slightly more complicated example consider the tree that starts from the pair **P & Q**, **P v Q**. Here it is, in two versions

```
 P & Q                              P & Q
 P v Q                              P v Q
 ------                             ------
  / \                                 P            ( &
P     Q        (v rule)               Q            rule)
P     P        (&                    / \
Q     Q        rule)                P   Q          (v rule)
```

The differences between these two are not important. In the tree on the left the disjunction rule is applied before the conjunction rule, and in the tree on the right the conjunction rule is applied before the disjunction rule. But it really does not matter in which order the rules are applied. The result is the same: the tree has two branches and on each of them both **P** and **Q** are found. Interpreted in terms of sentences, this means that both **P** and **Q** have to be true in any models that make both **P & Q** and **P v Q** true. Interpreted in terms of queries, this means that if we search with both together we will get models/databases where both **P** and **Q** are satisfied.

4:5 (of 5) width

We can use trees beginning with several sentences to make an important point that concerns both when sentences are true and what the results of a search will be. Call one search wider than another in a particular model/database when it will find more than the other. Everything that the second search, the narrower one, finds is also found by the wider search. Wider searches are not always more useful, as they may find many things besides what you are looking for. You want information about feline leukemia and so you search on the internet with "cats" or "cat diseases": searching with "cats" is obviously too wide, since you will get millions of results and nearly all of them will be irrelevant, but searching with "cat diseases" will also swamp you with unwanted information. More is not always better. So there is an art to wording a search so that it is wide enough to give the information you want but not so wide that you have so much information that you cannot separate out the items that you really need. (The ideal, of course, would be to use a query that finds exactly what you want, not too much and not too little, but that is usually not possible.)

Some searches are intrinsically wider than others: they will get at least as many results in any model. For example **Find x Cat x** is intrinsically wider than **Find x (Catx & Blackx).** Even if you are applying these queries to a model where there are only black cats the wider search will never gets fewer than the narrower one. In exactly the same way some sentences are wider (or logically weaker, philosophers sometimes say) than others. The models where "Nero is a cat" is true will never be fewer (never be a proper subclass of) those where "Nero is a black cat" is true. The width of sentences will return in chapter 6 in terms of logical consequence.

Trees enter here. A tree breaks down the sentence or criterion it starts with into a number of branches. Each of these is, taken alone, wider than the starting sentence. For a simple example consider black cats again. We have the tree

Cat & **B**lack
Cat
Black

(where **C**at & **B**lack could be **Find x (Cat x & Black x)** or "Nero is a black cat", **C**at could be **Find x Cat x** or "Nero is a cat", and **B**lack could be **Find x Black x** or "Nero is a cat".)

Notice that both **C** and **B** are wider than **C & B**. The models for **C** (or for **B**) always include the models for **C & B**, and usually more. So the entries on the one branch of this tree are wider than the starting point. We get another perspective on this by considering a branching tree. For example

Cat ⊃ **B**lack
/ \\
~**C**at **B**lack

Now the items on the branches are *less* wide than what we began with. Models for

Cat ⊃ **B**lack include models for **~C**at, and also include models for **B**lack, but not necessarily the other way around. But there is not really any tension between this and what I said about the **C & B** tree. Items that are found on *all* branches of a search tree are wider than the sentence at its base. The **C & B** tree had only one branch, so the two items on it are wider than **C & B**. The **C & B** tree has two branches, so this is not true of it. (There is a relation between each branch and the sentence at the base, but it will not concern us until chapter 6.)

Width takes an interesting form when we begin a search tree with several sentences. A very simple example is

```
A v B
 ~A
_____
 /  \
A    B
X
```

The only item on the only unclosed branch is **B**. But **B** is wider than neither **~A** nor **A v B** . It is less wide than **A v B**, and it is neither wider nor less wide than **~A** .This may seem puzzling, given what was said just above. But the puzzle is easily resolved. **B** is wider than **A v B** and **~A** *taken together*. When a tree begins with a collection of sentences, then items that occur on all branches are wider than the intersection of all them: if *all* of the sentences at the base are true in a model then anything that is found on all branches is also true in it.

This is an important fact. It is one reason why logical thinking is powerful. And the example above shows how. Two instances of the tree are (a) "it is either in the basement or the storage room. But it is not in the storage room. So it must be (somewhere) in the basement." and (b) "find everything that is in the basement or in the storage room. But do not look in the storage room. So look in the basement." (a) and (b) are clearly very similar, which illustrates how close reasoning with statements and reasoning with queries are to each other. In this case they both fit a pattern of reasoning where we begin with something that applies in many cases (**A** and **B**) and use further information to narrow down the facts or the target for our search into something manageably small.

Here is an example of this process. Three investigators working for three intelligence agencies are trying to track down a hacker who they have identified on the web. Investigator A knows that he is somewhere in Europe, so he could search with

Find x: Europe x. But that is millions of places and people. Investigator B knows that the only place in Europe the hacker could be is Spain, so B can formulate the query

Find x: Europe x ⊃ Spain x. But that is also an enormous search: it includes everyone and everywhere on the planet except people and places in European locations that are not in Spain. Investigator C has intercepted an email saying that his safe house in Spain is 1492b Avenida Boabdil in Grenada, apartamento 28. When he is in Spain, which you may or may not be at the moment as far as C alone knows, that is where he will hang out. So C can formulate

Find x: Spain x ⊃ 1492b-28 Boabdil x. Note that this too is satisfied by millions of people.

>> why do millions satisfy this last query?

Finally the superiors of A, B, and C allow them to compare notes and they can jointly formulate the search

Europe x
Europe ⊃ Spain
Spain ⊃ 1492b-28 Boabdil

And taking all three together they can narrow down the search to **1492b-28 Boabdil**, which will allow them to go and arrest their quarry. The combination of three wide searches gives a very narrow one.

>> make an appropriate search tree and show that this address is on all unclosed branches.

This illustrates how we can get narrow searches by combining queries that are individually wide. In this case the combination happened by accident, though the investigators may have been hoping for something along these lines. But making narrow searches by combining wide queries is a feature of intelligent thinking. We will see a parallel feature of axiom systems in chapter six.

words used in this chapter that it would be a good idea to understand: conjunction rule, conditional rule, disjunction rule, model, negation rule, search tree, truth table exercises for chapter four

A core

1) Here are six search patterns, which could be found either in searches for individuals or in searches for models. Which of them conform to the rules for search trees described in the chapter, and which do not? When they do not, say why they do not.

```
A v (B & C)              P ⊃ Q                 ~(A & B)
 /      \               /     \                  ~A
A      B & C           P      ~Q                 ~B
         B
         C

A v (~A & B)            (P ⊃ Q) & R          ~(A v B) & C
   /    \                  /   \               ~(A V B)
  A   ~A & B             ~P     Q                  C
        ~A                R     R                 ~(A V B)
         B                                           ~A
         X                                           ~B
```

2) a) What are the search trees for each of the queries below? (Remember that you ignore the **Find x:** and work just with the criterion.)

i) **Find x: C**at**x & (M**ale**x v T**ough**x**)

ii) **Find y**: **S**mall**y** ⊃ **B**lack**y**

iii) **Find z: Ax ⊃ (Bx v Cx)**

iv) **Find w: (Bx & Cx) ⊃ ~Ax**

b) What individuals will iii) get that iv) will not? What individuals will iv) get that iii) will not?

c) For each of i) –iv) use the search tree to produce a model in the form of a table where the query will have a non-null result.

3) Given the model described by the table below, what truth values do **Pa**, **~Qb**, **Pa v Qb**, **Qa** ⊃ **~Pc** have?

	P	**Q**
a	NO	NO
b		YES
c	YES	YES

4) You are faced with an enormous model (database) with hundreds of individuals classified under six categories, **A** ,**B**, **C**,**D** ,**E** ,**F**. You are supposed to find an individual — just one will do — such that if it is **F** then it is not **A**. Your boss says to do this by starting with the first individual and then working through all of them working out the truth value of **Fx** ⊃ **~Ax** for each of them. This will take hours, so you suggest a shortcut. What is the shortcut?

5) a) You want to catch all the Feral cats in a park and some Coyotes too, but only those that have had a rabies Vaccination. Which of these queries describes your aim?

Find x: Fx & Cx & (Cx ⊃ Vx)

Find x: (Fx v Cx) & (Cx ⊃ Vx)

Find x: Fx v (Cx ⊃ Vx)

b) Write search trees for each of these, and verify with a sample of cats, coyotes, and vaccinated animals that your answer to **a)** is right.

Pay attention to the difference between AND and OR in this question, and to what an IF query asks for. What is the difference between the second and third queries?

What will the third search get, as well as cats and vaccinated coyotes?

6) For an essay assignment on Japanese literature, you have to get books on Murasaki Shikibu (Lady Murasaki). At the same time you have an assignment in a different course on the history of cities, and you decide to make the reading of the two coincide by getting books on 10'th century Kyoto. You would like to combine the two searches "Murasaki Shikibu, poetry, novel" and "Kyoto, history, pre-modern". Describe several (at least three) searches that will get all the books that either of these two will get. What more will they get? Which one is best?

7) a) Compare these searches in terms of width:

i) oil & pipelines & pollution

ii) oil v pipelines v pollution

iii) oil & (pipelines v pollution)

iv) (oil & pipelines) v pollution

b) Construct search trees for i) –iv). How do they compare to your answers to a)?

B MORE

8) Annotate each of the trees below to show which rule is used for each step. I have provided the annotations for (i), to demonstrate what should be done.

(i) A & ~(B ⊃ C)
A) &
~(B ⊃ C)) rule
B)mistake
~C) with ~⊃ rule

(ii) A & ~(A ⊃ B)
A
~(A ⊃ B)
~A
B
X

(iii) A v (B & C)
/ \
A (B & C)
B
C

(iv) A & (B v C)
/ \
A (B v C)
B
C

(v) A ⊃ B
A
/ \
~A B

(vi) A ⊃ B
B
/ \
A ~B

9) This chapter has focused on structures that are found both when searching for individuals with queries and when searching for models with sentences. The trees in the previous question, **8)**, can be understood as describing both kinds of search. (In accordance with their application to searching for models, they reappear in later chapters in the guise of patterns of deductive reasoning.) For (v) in **8)**, give a model where the unclosed branch describes all the individuals that satisfy the criteria of a pair of searches, and also give a pair of models where the atomic sentence on the unclosed branch is true on the model which makes the two sentences at the base of the tree true, but is false in the other model, which does not make both of the sentences at the base true.

10) For each unclosed branch of each of the trees below construct a model which makes all the sentences on that branch true

<u>**(Pa & Qa) v ~Pb**</u>

```
        /  \
Pa & Qa        ~Pb
Pa
Qa
```

<u>**(Am ⊃ Bn) & ~Bn**</u>

```
(Am ⊃ Bn )
        ~Bn
        /  \
    ~Am    Bn
             X
```

(Pm & Qn) v ~Ps

```
          /       \
Pm & Qn      ~Ps
Pm
Qn
```

(Aa ⊃ Bb) & Ba

```
    (Aa ⊃ Bb )
        Ba
       /   \
    ~Aa    Bb
```

11) below are some searches and for each of them a simpler but potentially wider search. In each case say whether the increased width would make the simpler search less useful, and why.

longer search	simpler but wider search
Find x: livingon**E**arth**x v** livingOn**V**enus**x**	**Find x:** livingOn**V**enus**x**
Find x: Flying**x & A**nimal**x**	**Find x: A**nimal**x**
Find x: Dog**x v C**at**x** **Find x: ~C**at**x**	**Find x: D**og**x**
Find_x: **L**ive-young-bearing**x v M**ammal**x** **Find x: ~M**ammal**x**	**Find x:** **L**ive-young-bearing**x v M**ammal**x**
Find x: Drugdeale**x** ⊃ d**A**ngerous**x** **Find x: D**rugdealer**x**	**Find x: D**angerous**x**
Find x: Bookstore**x** **Find x: C**anada**x** ⊃ B**Cx** **Find x:** B**Cx** ⊃ **V**ancouver**x**	**Find x: B**ookstore**x** **Find x: C**anada**x** ⊃ **V**ancouver**x**

12) Two queries can be *complementary*, in that one gets a result if and only if the other does not, or they can be *non-overlapping* in that no result is ever got by both. Which of the following pairs of queries are complementary and which are non-overlapping? (When in English we say "opposite" sometimes we mean complementary and sometimes non-overlapping.)

a) **Find x: G**reen **x** , **Find x: R**ed **x**

b) **Find x: G**reen **x** , **Find x: ~G**reen **x**

c) **Find x: G**reen **x** ⊃ **B**ird **x** , **Find x: G**reen **x & ~B**ird **x**

d) **Find x: G**reen ⊃ **B**ird **x** , **Find x: B**ird **x** ⊃ **G**reen **x**

13) You are searching in a database with full Boolean search, for people who have won the Nobel prize, but in Literature you are only interested in French authors. Which of the queries below gets what you want? Back up your answers with search trees.

a) Nobel prize & (French ⊃ Literature)

b) Nobel prize & (literature ⊃ French)

c) (Nobel prize & ~ literature) v (Nobel prize & literature & French)

d) Nobel prize & French & Literature

e) (Nobel prize & ~French) v (Nobel prize & literature & French)

14) You are searching on the internet for non-human intelligence. If you enter the search terms intelligence, -human , you know you will get too many hits, and most of them will be irrelevant. But you think that on earth intelligent life is likely to be among whales, dolphins, apes, corvids (crows, jays, and ravens), and parrots. How do you take account of these assumptions to make a manageable search?

15) We are interested in information about the Turkish author Orhan Pamuk, but we do not want interviews with him except for the one in which he discussed Turkey's refusal to acknowledge the Armenian genocide. So we want "Pamuk and if interview then genocide". **Find x: Px & (Ix ⊃ Gx)**. But we are using a search engine that does not do conditionals, so we consider various substitutes.

(a) **Find x: Px & Ix** (*Pamuk, interview* in Google terms)

(b) **Find x: Px & ~Ix** (*Pamuk, -interview*)

(c) **Find x: Px & Ix & ~Gx)** (*Pamuk, Interview, -genocide*)

For each of these say if it is too broad, in comparison with the conditional search, or too narrow, or both.

C: harder

16) Chapter 4 ignored queries with more than one variable such as

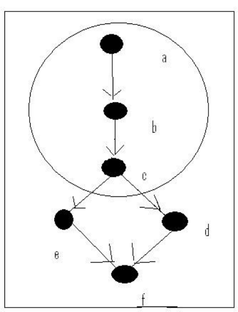
Find (x,y): Rxy, and models for sentences such as **Rab**. Discuss the complications of considering these and sketch a modification of section to handle them.

17) Here is a model presented as an arrow diagram to represent the relation **M** "is **M**ore popular than" plus a closed region to represent the attribute "is **N**asty".

(a) Verify that the search *in this model* (i) **Find x: Mxb ⊃ Nx** is wider than both (ii) **Find x: Mxb** and (iii) **Find x: Nx**.

(b) Modify the model so that (ii) is wider than (iii), and so that (iii) is wider than (ii).

(c) What does this show about whether either of (ii), and (iii) is intrinsically wider than the other, as discussed in section 5 of the chapter)? What about (iii) and (i)?



(The purpose of this exercise is to make you familiar with diagrams like this, which combine relations and attributes, and also to make you familiar with diagrams in which a relation puts the domain in an order. The order in this case leaves some individuals unordered with respect to one another.

(d) What general features would a relation have to have if every individual was ordered with respect to every other?)

18) (for mathematicians) prove that every unclosed branch of a tree with one sentence at its base determines a model for that sentence. (Hint: show that (a) if a rule generates a branch when it is applied to a sentence S then the models which make all the sentences introduced onto that branch by the rule also make S true, and (b) the atomic sentences on any unclosed branch determine a model.)

19) (for eager mathematicians) prove the same result for trees with a finite number of sentences at their base.

chapter five: truth assignments

In this chapter and the next two we study the way whole sentences can be combined using the Boolean connectives to make other sentences, when the result is true or false, and how this affects some important logical concepts. This is what is called propositional logic. We have already met the Boolean connectives AND, OR, IF, and NOT in their role of making complex queries from simpler ones. As a result much of what we see will look familiar. There is a basic difference, though. Queries are not true or false: they are instructions that can be carried out and which collect or get individuals from a domain. Their results are collections of these individuals. Sentences of natural languages such as English and propositions of artificial languages such as those we will discuss in the rest of this book are true or false, once their words are interpreted in a domain; they have truth values. We can describe the ways that Boolean connectives influence the truth values of propositions in very simple ways, which have consequences for issues about the logical relations between them. This is what is called propositional logic. It has a particularly simple notation and structure. It allows a simple way of writing down logical facts so they are easy to remember.

>> how fundamental is the line between statements and instructions? how is "find the green parrots" related to "I want the green parrots" and to "which are the green parrots?" how is "grass is green" related to "believe that grass is green" and to "is grass green or does 2+2=9?"

5:1 (of 9) sentences, propositions, and models

Statements such as "grass is green", "2+3=7", and "Earth is not the only planet with life" are true or false. The last of these is either true or false, but we do not know which. There are borderline cases, such as "yes, sure", or "she's the one", where we need to know what the words are being used to mean on a particular occasion in order to know whether they have a chance of truth or falsity. Statements contrast with commands, such as "Find all the citizens who have spoken against the government" or "Touch your nose and make a funny face". They also contrast with questions, such as "Who are the citizens who have dared to have their own opinions?" or "How many toes has the average platypus?" All three kinds of sentence have to have the right grammatical structure, and this can include Boolean connectives such as AND, OR, IF, and NOT.

>> does "earth is not the only planet with life" have some of the problems of "she's the one"? what about "2+3=7"?

We have been studying one particular kind of command, expressed by the **Find x** notation. It gives a simple and intuitive insight into some aspects of Boolean structure, and complex searches are of obvious practical interest. The interest in complicated searches will continue here and in future chapters. But the complications of search commands, queries, make it harder to describe some simple aspects of Boolean structure.

We will sidestep these problems by working with formal languages like those we have used for search commands. A clue to how to do this is given by a central fact of the previous chapter (Chapter 4). Search trees show that finding models that make sentences true is very similar to finding models with individuals which the criteria of queries are true of. So if we can focus on the structures, and to searches for models making sentences true and models in which we can find individuals, some things will be much simpler. Then instead of writing atomic propositions in the form Pa, representing cells in databases, we will denote them with the letters **p, q, r, s, t,** $\mathbf{p_1}$**,** $\mathbf{p_2}$**,...** . In practice we will not need $\mathbf{p_1}$**,** $\mathbf{p_2}$**,...** , but including them makes it clear that there are infinitely many atomic propositions, thus infinitely many propositions in all. All Boolean combinations of atomic propositions are molecular propositions of propositional calculus. So **p & q**, **p v q**, **r ⊃ ~s**, **~(p ⊃ (q & ~r))** are typical molecular propositions. Since there are infinitely many atomic propositions and many ways of joining them, there are infinitely many molecular propositions. The artificial language that uses these atomic and molecular propositions — so it is entirely made up of atomic propositions and Boolean combinations of them — is the language of propositional logic[13]. Its sentences are called propositions. I give a slightly more careful description of what counts as the language of propositional logic in section 3 below, and a precise definition in the appendix to this chapter..

5:2 (of 9) truth assignments and truth tables

The basic properties of propositional logic can be described in terms of *truth assignments*. A truth assignment is an assignment of a truth value, **T**rue or **F**alse, to every atomic proposition in the language. These truth values can then be applied to molecular propositions in accordance with the truth tables for their connectives. Truth tables were introduced in the previous chapter and are also discussed below. The previous chapter (Chapter 4) showed how we can construct truth assignments from models and how we can construct models from truth assignments. So talking about truth assignments is a simple way of talking about models, when only the structure given by Boolean connectives matters. In this way propositional logic is useful for discussing features that queries, simple object and attribute or relational sentences, and the quantified sentences that we will study later, have in common.

An important fact about Boolean connectives is that the truth value of a Boolean combination of propositions, one got from by joining them with Boolean connectives, is determined by truth values of its parts, as specified by a truth assignment. We say that molecular propositions that are Boolean combinations of atomic propositions are *truth*

13 It is sometimes called the propositional calculus. This can be confusing because people think that "calculus" must involve integrals and derivatives. But “calculus” is an old-fashioned word for a way of calculating, and comes from the Latin for a pebble. You can do arithmetic with pebbles, but you can also calculate truth values of complex propositions from truth values from truth values of simple ones. The old-fashioned phrase for what we now call calculus was "infinitesimal calculus", calculating with tiny quantities, as if sand instead of pebbles.

functions of them. This is important because it means that to tell what molecular propositions are true in a model we only need to know what atomic propositions are true in it. This does not mean that it is always obvious whether there is a truth assignment makes a molecular proposition true. This is also important, and we will return to it in chapter 7.)

Each of the Boolean connectives has a meaning in logic, which is slightly simpler than the meaning it has in everyday English. The special feature of the meanings of the connectives in logic is that they are truth functions. For example an AND sentence is true when (and only when) both of the sentences it joins are true. There is a simple principle like this for each of the connectives. Here they are:

A & B	is true if and only if	both A and B are true
A v B	is true if and only if	one of A, B is true
A ⊃ B	is true if and only if	A is false or B is true
~A	is true if and only if	A is not true

These are important. Learn them. They give the meanings in logic of AND, OR, IF, and NOT, written as **&, v, ⊃, ~**. These four words, either expressed in English or in symbols, are the basic *connectives* of logic. Their meanings as given by the truth tables are more focused than their meanings in ordinary English. They can be summed up with truth tables. (Some people find truth tables easier to work with and some people are more comfortable with rules like those in red in the box above. I find the rules less confusing when dealing with molecular propositions. Constructing a truth table for a complex proposition gives many opportunities for mistakes.) The truth tables for the connectives were given in the previous chapter (4), but I shall give them again.

AND/&

With AND the thing to remember is that it leaves out everything about time and causation. It just says that a conjunction — got by joining two *conjuncts* by & — is true when both of the conjuncts are true, and in no other cases. (In ordinary English "he robbed a bank and got sent to jail" is not surprising, while "he got sent to jail and robbed a bank" is rather puzzling. It's as if we often hear "and" as "and then" or "and so". But in logic if either of A & B and B & A is true the other is.)

The basic facts about conjunction, that a conjunction is true only when both contracts are true, is summed up in the truth table:

P	Q	P & Q
T	T	T
T	F	F
F	T	F
F	F	F

OR/v

With OR the thing to remember is that it is the *inclusive* meaning of "or". That is, a disjunction — made by joining two *disjuncts* by "or" — is true whenever at least one of the disjuncts is true (and in no other cases.) According to this a disjunction is true when both disjuncts are true. So "you can write the exam or you can write a paper" does not rule out the possibility that you can do both, on this meaning of "or". (I think that most uses of "or" in English are actually inclusive, but people usually find this a surprising claim, since they think of exclusive uses of "or" as in "for five dollars you can have soup or salad", mentioned in chapter one. It can be hard to persuade them that these are the exceptions.) At any rate the truth table is:

P	Q	P v Q
T	T	T
T	F	T
F	T	T
F	F	F

IF/ ⊃

With IF there are many differences between the meaning given by the truth table above and various things it can mean in everyday language. This is a big topic on which a lot has been written[14]. For our purposes it is enough to say that the material conditional, the meaning of "if" or ⊃ given by the truth table, is often a simple and manageable substitute for the very subtle and puzzling English word. An "if" sentence, or *conditional*, is made from joining an antecedent and a consequent: if A then B. (If it rains then the sidewalks will be wet; if she comes to the party then I'll leave.) To repeat, the first component("it rains", "she comes to the party") is called the *antecedent*, and the second ("the sidewalks will be wet", "I'll leave") is called the *consequent.* According to the meaning given by the truth table for ⊃ a conditional is true in all cases in which the antecedent is false and in all cases in which the consequent is true. What it does is just to exclude the cases in which the antecedent is true and the consequent is false. Like the meaning of AND this differs from the English meaning by ignoring all considerations about time and causation.

14For some psychology on the topic see J.Evans & D. Over *If,* Oxford University Press 2004*.* The definitive philosophical book is J Bennett , *A Philosophical a Guide to Conditionals*. Oxford University Press 2003.

The truth table for this material conditional is

P	Q	$P \supset Q$
T	T	T
T	F	F
F	T	T
F	F	T

>> this meaning of IF fits perfectly with the meaning we used in Boolean search, in **Find x: Ax ⊃ Ax** . say why.

NOT/~

This is the closest in meaning to the English. To a first approximation the truth table gives the meaning of the English "not" as well as the logician's **~** . **~**A is true when A is false, and vice versa. This is summed up in the truth table for ~:

P	~P
T	F
F	T

>> why did I say "to a first approximation?" can you think of ways we use "not" in English that do not fit easily with the truth table?

We can also give truth table definitions of connectives besides AND, OR, IF, and NOT.

Here is a truth table that defines two more connectives.

P	**Q**	**P ↔ Q**	**P | Q**
T	T	T	F
T	F	F	T
F	T	F	T
F	F	T	T

>> ↔ has a familiar expression in English, "if and only if" (in mathematics iff). what about |?

Are these all the truth functions now? See exercise 36.

5:3 (of 9) the language of propositional logic

Given atomic propositions **p**, **q**, **r**, **s**,. . we can make molecular propositions such as **p & q**, **p ⊃ q**, or **~(p & ~q) ⊃ (r v s)**. The **p**, **q**, **r** and so on can be any propositions at all. The simplest way to think of them is as cells in models-as-tables, that is, combinations of attributes or relations with individuals. So **~p** ⊃ **r** might be an abbreviation for "**~Ab ⊃ Rcd**", if **p** is the cell where **A** meets **b** and **r** is the cell where **R** meets **c** and **d**. On another occasion the same proposition might be an abbreviation for the same truth function of different cells in a different table. But the point of writing it in this notation is to focus on the way the truth value of the whole proposition depends the values of the parts.

The symbols of propositional logic are less like a real language than those of Boolean search are, or those of quantifier logic, which we study in part III of this book, are. We can use the language of search commands and that of quantifiers to communicate. I have said to a class "Find x: on your desk x and pen x" and after a moment everyone has reached out to wave a pen at me. The same is only true in a much more limited way for propositional logic. But to communicate with propositional logic we would have to explain the meaning of every atomic proposition (presumably in terms of individuals, attributes, and relations, though propositional logic does not specify them) one by one. Then we could join them together to say more complex things. But propositional logic is often used as an easy example of how one can state a language's grammatical structure simply and precisely. This is ironical given that propositional logic is not a very good candidate for language status, but it makes sense given that there are easily stated rules for combining Boolean connectives in a well-formed way. It is also true that these rules are a core part of the well-formedness rules for better candidates, such as Boolean queries and the propositions of quantifier logic.

>> it would be a foolhardy thinker who stated “something is a language if and only if it ... ”. but describe some ways in which search commands are more like spoken languages than the symbolism of propositional logic is. (you may want to come back to this question when you have seen more about propositional logic.)

In any case, we can state simple rules for what is a well-formed proposition of propositional logic. These describe how simple proposition-letters **p**, **q**, **r**, ... can be combined in allowable ways by **~**, **&**, **v**, and ⊃ to make more complex propositions. I am not going to state the rules here. I do in an appendix to this chapter. For now, here is a list of well-formed formulas. The patterns will be familiar to you because of your practice with search commands.

p	**~r**	**~~s**	**p & q**
q & q	**q v p**	**~p v**	**q**
~(p & q)	**~p v q**	**~(p & ~q)**	**(p & q) v u**
p ⊃ q	**p & q & u**	**(u ⊃ p) v u**	**(p v q) & (p v q)**
p v q v u	**p v q v u v q**	**(((p ⊃ q) ⊃ q)⊃ q)**	

On the other hand the following are not formulas. They are ill-formed.

p & q v u	needs brackets
(p & q))	orphan bracket
& p	**&** has to join two propositions
p ~ q	**~** applies to just one proposition

(It’s like when you’re writing a program and the compiler says “syntax error”; very annoying. Compilers were invented by Grace Hopper.)

5:4 (of 9) central connectives and outside-in

Every proposition has a *central* connective. For example in **p ⊃ (q v r)** the central connective is ⊃. It is a conditional that has a disjunction as its consequent. In **p v (q & r)** the central connective is **v** : it is a disjunction that has a conjunction as its second conjunct. On the other hand the central connective of **p & (q v r)** is **&** : it is a conjunction that has a disjunction as its second disjunct. Because there is always a central connective there is never any scope ambiguity in formulas of propositional logic. We always know whether the formula is a negation, a conjunction, a disjunction, or a conditional. And we know it for each of the formula's subformulas: the negated formula, conjuncts, disjuncts, antecedent or consequent, right down to atomic propositions. So (**p & ~r) ⊃ q** is a conditional whose antecedent is a conjunct whose second conjunct is a negation.

When you see a roposition you should first of all identify the central connective, and then the central connectives of its parts. In chapter 3, section 3, I gave a moral "look at the whole expression, not just the smallest pieces". We often do the opposite in everyday life, when understanding ordinary language. Someone says "There is a black cow in the field, and it has a brown spot, or perhaps there is a tree in the way" and we first think "cow, field, spot, tree: now what is being said about them?" This is beginning from the inside, and mathematical expressions are usually best understood from the outside. When we do this, many shortcuts to understanding and evaluating the expression often appear. Often we do not need to think about its the full content, down to the smallest parts. For example **(p ⊃ p) ⊃ q** is a conditional whose antecedent is another conditional **(p ⊃ p)**, and this antecedent says something that is always true. So with suitable content for **p** and **q** the whole thing might express "Suppose that when Mo is happy Mo is happy. Then Jo is sad." And when we think of it as a conditional with a trivial antecedent, it is not surprising that, as we will see later in this chapter, this is equivalent to "Jo is sad." We don't have to wonder when Mo is happy. In all the equivalencies between formulas discussed in section 6 of this chapter it will help to read them in an outside-in way. It will help with exercises 3, 4, 5 ,21, too.

>> but in **~(p & (q ⊃ r))** the **~** is not in the middle, so how can it be central?

>> in $(x+y)^2$ what is the central algebraic operation?

>> in **(p & (q ⊃ (r v s))** the conjunction is the central connective. but its second conjunct itself has a central connective, the conditional. does that make the conditional in some way central?

How do we tell what the central connective is? In the way we are writing our formulas there is only one way of putting a formula together in the way just described. For example the structure of **~((p v q) & (r v s))** can be given in any of the four equivalent ways below.

~ A B v v **p q r s**	NEGATION CONJUNCTION DISJUNCTION DISJUNCTION **p q r s**
~((p v q) & (r v s))	**~ p v q & r v s** | ______|_____ |_________________

So to tell what the central connective is, try to reconstruct the sentence according to any of these three ways: the central connective is the top or outermost connective in the reconstruction that works.

>> give one or another of these analyses for each of the five formulas in the second last column of the well formed formulas above.

>> these analyses suggest ways that in everyday speech we resolve scope ambiguities by pauses and intonation patterns. try saying the example out loud, perhaps substituting English sentences for p, q, r, s to see if you can make this fit the way you speak.

>> if you know some linguistics: what does this remind you of?

5:5 (of 9) evaluating complex propositions

We return to central connectives below. The rules for applying the Boolean connectives given by the truth tables determine the truth value of complex formulas in terms of the truth values of their parts. This can be done mechanically using truth tables or, more easily, the rules summarized in the box in the previous section, which I will repeat them now, in a slightly different formulation.

A conjunction (**&**) is true only when both conjuncts are true. It is false except when both conjuncts are true.

A disjunction (**v**) is true as long as at least one disjunct is true. It is true except when both disjuncts are false.

A conditional (⊃) is true when the antecedent is false and when the consequent is true. It is true except when the antecedent is true and the consequent is false.

Negation (**~**) turns true to false and false to true.

The underlined versions are good for quick mental evaluations of truth values, rather like automatic Boolean search routines or quick mental arithmetic.

Calculating truth values is a little like algebra. Suppose x is 2 and y is 3, what is $(x+y)^2$? Well, given these numbers $(x+y)^2 = (2+3)^2 = 5^2 = 25$. Similarly, suppose that **p** is True and q is False, what is the truth value of **p ⊃ (q v p)** ? Well, given these truth values we can calculate:

p ⊃ (q v p) = **T ⊃ (F v T)** = **T ⊃ T** [by the rule for **v**] = **T** [by the rule for ⊃]. (I am writing **T** for the truth value True and **F** for False, as we do in truth tables.)

q is true and **r** is false. Is **(q ⊃ r) v r** true or false?

(q ⊃ r) v r = **(T ⊃ F) v F** = **F v F = F**

q is false, and **r** and **s** are true. What is the truth value of **(r & s) ⊃ ~s** ?

(r & s) ⊃ ~s = **(F & T) ⊃ ~T** = **F ⊃ F = T**

>> put "[by the rule for ..]" notes in these calculations.

Working out the truth values of molecular formulas given the truth values of their atomic components is a good way of getting familiar with the Boolean connectives. There are questions in the exercises for this chapter giving more practice.

These calculations are made possible by the fact that every Boolean formula has a central connective, with the other connectives in its scope. For example the central connective of **p ⊃ (q v p)** is ⊃. **p ⊃ (q v p)** is a conditional with antecedent **p** and consequent **q v p** , so its truth value in the example above is the value of a conditional with a True antecedent and a consequent that has the value of **(F v T)** = **T**. The central connective in **p & (q ⊃ r)** is **&**. It is a conjunction of **p** and the conditional **q ⊃ r**. And the central connective of **~(p & (q ⊃ r))** is **~**: the whole formula tells us "it is not true that both p and the conditional from q to r".

The calculation of the truth value of a complex expression in terms of truth values of its atomic parts can be done fairly automatically. You should not have to think too much, and in fact you are likely to make more mistakes and find the process more confusing if you are too verbal about it. The pattern of central connectives should tell you what kind of a calculation you should set yourself to do without too much deliberate oversight. This is like the way that I advised carrying out certain commands in the first three chapters. And in fact it is a general characteristic of mathematical thinking: the verbal part of your mind tells some nonverbal part what to do and then lets it get on with the job. (I make some more remarks about this in chapter 9 section 7.)

5:6 (of 9) equivalent propositions, contradictions, tautologies

Sometimes two propositions have the same values on all lines of a truth table. Exactly the same truth assignments make them true. We then say that they are *equivalent*. Remember that A, B, and so on are placeholders that can represent any proposition.

For example, here is a basic connection between the conditional, disjunction, and negation: between **v** and ⊃ **:**

A v B is equivalent to ~A ⊃ B

To show that this is right we make a truth table for ~A ⊃ B .

truth table for ~A ⊃ B

A	B	~A	~A ⊃ B
T	T	F	T
T	F	F	T
F	T	T	T
F	F	T	F

We get this truth table as follows. The column for ~A follows the principle for negation just above: negation flips T to F and F to T. So it is just the column for A with T and F reversed. To get the column for ~A ⊃ B we think as follows: this is a conditional, so it is true except when the antecedent is true and the consequent is false. So it will have Ts except when we have T for its antecedent, ~A, and F for its consequent, B. That is the last row of the table, so we get the truth table for ~A ⊃ B as above, all Ts except for the last row.

Now compare this truth table to the truth table for **v**. To repeat:

A	B	A v B
T	T	T
T	F	T
F	T	T
F	F	F

The conditions under which ~A ⊃ B and A v B are true and false are exactly the same. If either is true so is the other; if either is false so is the other. A and B were standing for any propositions so for example **~p ⊃ q** is equivalent to **p v q** , and **~(q & r) ⊃ (r v s)** is equivalent to **(q & r) v (r v s)** .

Here is another equivalence:

A v B is equivalent to ~(~A & ~B)

To show that this is right we write out the truth table for ~(~A & ~B)

A	B	~A	~B	~A & ~B	~(~A & ~B)
T	T	F	F	F	T
T	F	F	T	F	T
F	T	T	F	F	T
F	F	T	T	T	F

The ~A and ~B columns are got by flipping the truth values in the A and B columns. Then the ~A & ~B column is got by the rule that as a conjunction it is true only when both of its conjuncts, in this case ~A and ~B, are true. Then the ~(~A & ~B) column is got by flipping the truth values of the ~A & ~B column.

The column for ~(~A & ~B) on the resulting truth table, above, is identical to the column for A v B in its truth table. So any formula ~(~ A & ~B) is equivalent to the corresponding A v B .

There are many such equivalences. Here is a list of some of them, including the two we have just seen.

A v B is equivalent to ~(~A & ~B) A & B is equivalent to ~(~A v ~B)	(de Morgan's laws)
A v B is equivalent to ~A ⊃ B	(definition of v in terms of ⊃)
~~ A is equivalent to A	(law of double negation)
A ⊃ B is equivalent to ~ (A & ~B)	(definition of ⊃ in terms of ~ and &)
A ⊃ B is equivalent to ~A v B	(definition of ⊃ in terms of ~ and v)
A ⊃ B is equivalent to ~B ⊃ ~A	(contraposition)
A & (B v C) is equivalent to (A&B) v (A & C)	(distribution of & over v)
A v (B & C) is equivalent to (A v B) & (A v C)	(distribution of v over &)

The first two of these relate AND and OR by means of NOT. They are collectively known as *de Morgan's laws.* All nine equivalences are worth learning. (And they all have names, if that helps.) Memorize them if you have to, but in fact they each make sense: a few moments reflection on each of them should convince you that it is true. While you are at it you might you might as well get into your heads variants on de Morgan's laws, which are sometimes useful.

~A v ~B is equivalent to ~(A & B)

~A & ~B is equivalent to ~(A v B)

(Note that in English we have a special word for ~A & ~B, or equivalently, ~(A v B) . We say "neither A nor B". Although circuit designers use the term "nand", we do not have a short word in regular use in English for ~A v ~B or equivalently ~(A & B) .)

>> is there a word for ~A v ~B in some language you know?

Propositions are equivalent when they have identical truth tables. Two particularly simple truth table patterns that propositions can share are when they are all T and when they are all F. Propositions of the first kind are called *tautologies*, and they are all equivalent. The following are among the many tautologies:

p v ~p , q ⊃ q , (p & q) ⊃ (p v q) , ~(p v q) ⊃ (~p & ~q) , (~p & ~q) ⊃ ~(p v q)

the very simplest is **p v ~p**. Its truth table is

p	**p v ~p**
T	**T**
F	**T**

As you can see by applying the rules for **v** and **~**. All the others are similar.

Propositions of the second kind are called *contradictions*. They also are all equivalent.

The following are among the many contradictions:

p & ~p , p & q & ~(p v q) , ~(q v ~q) , (p ⊃ q) ⊃ ~(p & ~p) , (p v ~p) ⊃ (q & ~q)

The very simplest is **p & ~p**. Its truth table is

p	**p & ~p.**
T	**T**
F	**T**

As you can see by applying the rules for **&** and **~**. All the others are similar.

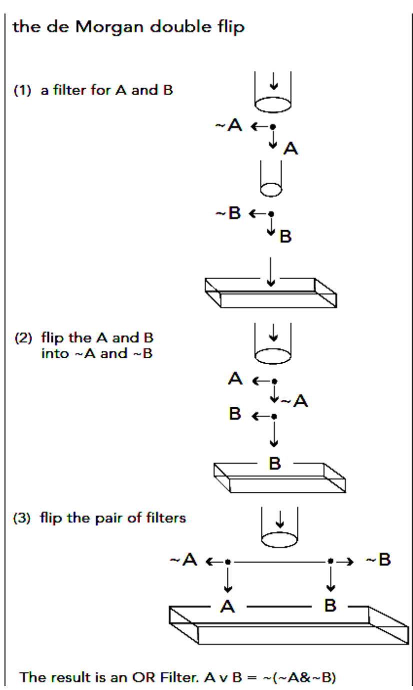
Intuitively, tautologies are propositions that assert something trivial: they cannot fail to be true. And contradictions are propositions that assert something completely impossible, there is no way that they can be true. In spite of the equivalence to something trivial tautologies can be complicated, because it can be far from obvious that a proposition is a tautology. For example it would take some effort to see that **(r ⊃ (s ⊃ p)) ⊃ (s ⊃ (r ⊃ p))** is true on all lines of its truth table. (And it would take some effort to see that **((p ⊃ (q&~q)) ⊃ ((r v~r) ⊃ p)** is false on all lines of its truth table.)

>> so since **p v ~p** and **(q & r) ⊃ q** are both tautologies, they are equivalent. but does "either it is raining or it is not" really say the same as "if Sam is happy and Mary is sad then Sam is happy”?

5:7 (of 9) more on de Morgan’s laws

“The negation of a conjunction is the disjunction of the conjuncts”, “the negation of a disjunction is the conjunction of the disjuncts”. These should feel intuitively right to you. If A & B is false, then they’re not both true, so one of them is false. at least one of A, B is false. And if A v B is false, then not even one is true, so they're both false. Think about these until they click. Somehow when we handle the symbols we get confused, though. We think ~(A & B) must be ~A & ~B, but this is wrong. (And similarly for ~(A v B) .) Another reason to be clear about the scopes of negations.

de Morgan's laws can make sense visually, if we picture filters the right way. Remember how AND amounts to filters in series while OR amounts to filters in parallel (pictures in chapter 2 section 3.) So what gets through the filter goes on down and what is blocked goes off to the side. Now think of negation as taking what gets through a filter and throwing it away, while letting through (collecting) what the filter blocks (throws away). Picture this as rotating the whole filter, so what had gone done now goes out, and what had gone out goes down. This puts parallel component filters in series and series component filters in parallel, and when we rotate these filters — negating them — we get the right effect. This can be pictured as follows:



Think about de Morgan's laws till they make sense, or use the pictures: whatever works for you. You you will have a good sense of the scope of negation: eventually they will seem simple and natural, and you cannot understand why they were once confusing.

5:8 (of 9) disjunctive normal form

One general way of finding equivalent formulas has many applications. It will allow us, for example, to design complex internet searches. Think about why the following two equivalences are true.

A ⊃ B is equivalent to (A & B) v (~A & B) v (~A & ~B)

A & (B v C) is equivalent to (A & B & C) v (A & B & ~C) v (A & ~B & C)

We can see that these are equivalent by considering the lines of the truth tables for the formulas on the left. The first is simpler and we can see the equivalence by considering its truth table, which by now is familiar:

A	B	A ⊃ B
T	T	T
T	F	F
F	T	T
F	F	T

Think of this as saying that for A ⊃ B to be true we have to have T on the first, third, or fourth line of the truth table. That is, either A is T and B is T, or A is F and B is T, or A is F and B is F. And when any one of these is T, so is A ⊃ B. But that is the same as saying that A ⊃ B is equivalent to (A & B) v (~A & B) v (~A & ~B) . The longer proposition just spells out which lines of the truth table make A ⊃ B true.

The same reasoning applies to A & (B v C) . The truth table is big, because it involves three propositions, so it takes 8 lines to give all the possibilities[15].

A	B	C	A & (B v C)
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	F
F	F	T	F
F	F	F	F

A & (B v C) is true on the first three lines, and only on them. That means that it is true if A is true and one of B, C is too. (Check the truth table to see that this is so.) And that means that it is equivalent to (A & B & C) v (A & B & ~C) v (A & ~B & C) .

>> we saw in section 4 that A & (B v C) is equivalent to (A & B) v (A & C). so (A & B) v (A & C) must be equivalent to (A & B & C) v (A & B & ~C) v (A & ~B & C) , since they are both equivalent to A & (B v C). check this with a truth table.

This line of thought is very general. It can be summed up as: any proposition is equivalent to the disjunction of a set of conjunctions describing all the truth table lines which make it true. Since these conjunctions consist of atomic letters and their negations, we can state the *disjunctive normal form theorem*: every Boolean formula is equivalent to a disjunction of conjunctions of atomic formulas and their negations.

>> (for mathematicians) I said "we can state" rather than "we have shown (or proved)". what more details need to be added to fill this out to a full proof? (See exercise36.)

The disjunctive normal form theorem should not be very surprising. It says essentially that every Boolean formula states "I am true if this is the case, or if this is the case, or .."

15 This is not a trivial matter. It is part of the reason that although the idea of a truth table is simple, and small ones are easy to handle, the task of doing them for arbitrarily large sentences is too demanding for efficient computer programs.

where the list runs through all the situations in which the formula is true. It is like saying that "C is a country in North America" is equivalent to "C is Canada or C is the USA or C is Mexico."

5:9 (of 9) Boogle

The disjunctive normal form theorem gives us a systematic way of doing Boolean search on many internet search engines. I'll end this chapter by explaining.

Begin by supposing, unrealistically, that our search engine can handle disjunctions and negations. Suppose, to give a semblance of reality, that we enter search terms separated by commas representing conjunction, optionally prefixed by -, or some other symbol representing negation, and with brackets separating OR strings. So "(cat, -white) OR (dog, -pitbull)" would be an instruction for the search

Find x: (Catx & ~Whitex) v (Dogx & ~Pitbullx).

This might be the search of someone looking for a pet to adopt, wanting a dog or a cat, with a phobia of white cats and a terror of pitbulls. You would have to have a very particular aim to do this search, but whatever your as long as what you want can be described using a Boolean combination of search terms, you can express it as a search using these terms. The disjunctive normal theorem of the previous section guarantees it: given a query **Search x: Cx**, where C is a Boolean combination of terms we can rewrite the **C** part as a disjunction of all the conjunctions of search terms and their negations which would by themselves find subsets of what we want. The "cats but not white or dogs but not pitbulls" search is an example.

>> "a disjunction of all the conjunctions of search terms and their negations which would by themselves find subsets of what we want." why is this an accurate wording?

>> why does " A is equivalent to B" tell us that **Find x: Ax** gets the same results a **Find x: Bx** ?

We can also do conditionals. Suppose we are doing research for a term paper in English and want 16th century plays but if they are in English they must not be by Shakespeare. The search we want is

Find x: Playx & Sixteenthcenturyx & (Englishx ⊃ ~Shakespearex)

The difficult part is the last conjunction, with its conditional. But we know now that it is equivalent to

Find x:

(English x & ~Shakespeare x) v (~English x & ~Shakespeare x) v (~English x & Shakespeare x)

This is not yet quite what we want. We want a query where all the conjunctions are in the scope of disjunctions, and this has a disjunction as a member of a conjoined list. But by applying the equivalence that

A & (B v C) is equivalent to **(A&B) v (A&C)**

or by applying the disjunctive normal form theorem to the whole query rather than its conditional part, we see that the whole query is equivalent to

Find x:

(Playx & Sixteenthcentury x & Englishx & ~Shakespeare x)

v (Playx & Sixteenthcentury x & ~English x & ~Shakespeare x)

v (Play x & Sixteenthcentury x & ~English x & Shakespeare x)

This is not a very complicated search. But we can simplify it by using our knowledge that Shakespeare wrote all his plays in English (and we are not interested in translations, say) so that the last disjunction will add nothing. So we have

Find x:

(Play x & Sixteenthcentury x & English x & ~Shakespeare x)

v (Play x & Sixteenthcentury x & ~English x & ~Shakespeare x)

And we can go to our imaginary search engine and enter

(Play , "Sixteenth century" , English , -Shakespeare)

OR (Play , "Sixteenth century" , -English , -Shakespeare)

or even more simply and without much loss

(Play , "Sixteenth century" , English , -Shakespeare)

OR (Play , "Sixteenth century" , -English)

The search engine was imaginary in two ways: it allows disjunctions and it allows negations. Search programs that are more specialized than general internet search engines will often do these, but our aim is to enable Boolean search on the big all-purpose sites. The disjunction problem is not severe. The recipe we have gives disjunctions a wider scope than conjunctions — we have (A AND B) OR (C AND D) but not (A OR B) AND (C OR D) — and that suggests an easy fix. We run each disjunction in a separate window. So to carry out the search for 16th century plays we open the search engine in two windows (or tabs). In one we enter

Play , "Sixteenth century" , English , -Shakespeare

and in the other we enter

Play , "Sixteenth century" ,-English , -Shakespeare

And then we combine the results. We are likely to do this by pasting, say, the first two screens of each into a single document.

Now for negation. As I explained in chapter three section 6, search engines often balk at negations because of the very wide searches they produce. The solution that usually works is not to search for NOT A, when there are many things that are not A, but to search for B & NOT A, where B is relatively small (on the scale of the internet) and most Bs are A. (The smaller the size of B, the larger the proportion of Bs in it one can safely search for.) The plays search already satisfies this to some extent, but we could improve it by replacing "NOT English by French OR Dutch OR German OR Spanish OR Italian" say (depending on the topic one is researching.) Then — skipping a few steps in a way that by now should be clear — we would open six windows and perform the following searches before combining them.

Play , "Sixteenth century" , English , -Shakespeare

Play , "Sixteenth century" , French

Play , "Sixteenth century" , Dutch,

Play , "Sixteenth century" , German

Play , "Sixteenth century" , Spanish

Play , "Sixteenth century" , Italian

>> can you fill in the skipped steps?

Different search engines will reject different negations, so one may have to experiment to find the combination of general category and negated sub-category (the A and B above) that works. But the general recipe is clear: (a) write the query as a disjunction of conjunctions and negations (b) reword the negations, introducing wider categories if necessary, so that they do not produce too many results, (c) run each disjunction in a separate search window, (d) combine the results.

>> using a regular search engine there is also the problem that e.g. “Shakespeare” will retrieve documents that contain the word “Shakespeare” even when they are not really about Shakespeare. describe ways of wording queries thatlessen this problem.

>> for sophisticated people like you these procedures are manageable. but the average Internet user is not going to be comfortable with them. describe extensions to familiar browsers and search engines that would make the process and easy. are there commercial possibilities here?

words used in this chapter which it would be a good idea to understand: atomic proposition, central connective, contradiction, de Morgan's laws, disjunctive normal form, equivalent propositions, model, molecular proposition, proposition, sentence, tautology, truth assignment, truth table, well-formed.

Appendix to chapter five: defining propositions of propositional logic

There are only eight rules.

1) **p**, **q**, **r**, **s**, **t** are atomic propositionss.

2) $\mathbf{p_n}$ for any numeral **n** written in standard decimal notation is an atomic proposition. (This allows us infinitely many atomic propositions. $\mathbf{p_1, p_2, p_3,}$ ... We will never run out, though in practice we never use more than a few. The clause about numerals in standard decimal notation is to prevent $\mathbf{p_{III}}$ or $\mathbf{p_{VIX}}$ from being atomic propositions.)

3) If X is a proposition then **~**X is a proposition.

4) If X and Y are propositions then **(**X **&** Y**)**, **(**X **v** Y**)**, **(**X ⊃ Y**)** are propositions.

5) If **(**X**)** is a proposition and is not part of any longer proposition then X is a proposition. (So both **p & q** and **(p & q)** are propositions. But **p & q v r** is not, though **p & (q v r)** is.)

6) If (X **&** Y) **&** Z is a proposition then **(**X **&** Y **&** Z**)** is a proposition.

7) If (X **v** Y) **v** Z is a proposition then **(**X **v** Y **v** Z**)** is a proposition.

8) These are all the propositions.

This is a recursive definition. It stipulates some simple cases and then defines more and more complex cases in terms of them. A recursive definition of "even number" might run:

a) 0 is an even number

b) if n is an even number then n+2 is an even number.

Applying b) to a) we get that 2 is even. Applying b) to this we get that 4 is even, and so on for 4, 6, 8, ...

Augustus de Morgan Frege as a young man Russell as an old man Grace Hopper,

Augustus de Morgan, 1806-1871, was a British mathematician. He was one of the founders of what is now the University of London, as an alternative to the universities of Oxford and Cambridge, which required that all their teachers be members of the Church of England.

Gottlob Frege (1848 – 1925) was one of the first to represent logical ideas in an artificial language.

Bertrand Russell [1](1872 – 1970) was one of the first to see the philosophical importance of ideas such as Frege’s.

Grace Hopper (1906– 1992) pioneered ways of transforming sentences of an artificial language into instructions for a computer.

exercises for chapter five

The **A** and **B** exercises for this chapter are divided into two parts, for sections 1 – 5 and 6 – 9 of the chapter, so that a course has the option of covering the chapter in two weeks. The **C** exercises are for the whole chapter.

PART I, secs 1 - 4

A – core

1) Which of the following are acceptable propositions of propositional logic (well-formed formulas)?

a) **(p & q) ⊃ r**
b) **q v ~q & r**
c) **r & q ⊃ p**
d) **p & ~q~**
e) **(p & r) & ~(q & s)**
f) **p & q & (r**
g) **~~(q ⊃ p)**
h) **q ~⊃ r**

2) What is the central connective in each of these?

a) **(p & q) ⊃ r**
b) **~ (r v ~s)**
c) **r & (q ⊃ p)**
d) **p & ~q**
e) **(r & q) & ~(r & p)**
f) **(q & p) v r**
g) **~(p ⊃ r)**
h) **p ⊃ (~q & s)**

3) Suppose that **p** and **q** are true, and **r** is false. What are the truth values of the following?

a) **p ⊃ p** b) **p & q & r** c) **(q vp) ⊃ p**

d) **(q v p) ⊃ r** e) **~(p v q)** f) **~p v q**

g) **p ⊃ (q ⊃ p)** h) **~p ⊃ (p ⊃ q)**

4) Assume that **r**, **s** are true and **p**, **q** are false. Which of these are true, which false and which ill-formed (and therefore neither true nor false)?

a) **r & s** b) **r & ~r** c) **r & p v q**

d) **r ⊃ s** e) **q ⊃ r** f) **p v (r & s)**

g) **p & (r v s)** h) **p ⊃ (q & r)** i) **(r & (r v s))**

j) **p ⊃ (q ⊃ ~r)** k) **p ⊃ q & s**

5) Write out the truth tables for

a) **p ⊃ ~q** b) **~p ⊃ q** c) **~(p ⊃ q)**.

d) **~p ⊃ (p ⊃ q)** e) **(p v q) & ~(p & q)**

f) **(p & q) v (~p & ~q)** g) **p ⊃ q) & (q ⊃ p)**

Notice how the first three differ. How do the last two compare?

6) a) In the following English sentences, the comma or connective could be replaced with a Boolean connective: AND, OR, IF, NOT. Which one is appropriate in each case?

it's a cat**,** it's a dog: I don't care
a step closer and I'll shoot
if he's a genius then I'm the Queen of Romania
no shoes**,** no shirt**,** no entry
there were many animals in the shelter: alligators**,** penguins**,** marmosets
either he pays his tab or we throw him out
use it or lose it

8) A Latin square is a grid of letters where each letter occurs exactly once in each row and once in each column. Sudoku are a special kind of Latin square. For example

C	B	A
A	C	B
B	A	C

is a Latin square. Often when some cells are left blank there are only a few ways of completing the square. If we had left the bottom row of the square above blank, there would have been only one way of completing it. For a slightly more difficult case consider

C	B	A
1	2	3
4	C	6
7	A	9

I have identified the blank cells with their "phone pad" numbering. How can this be completed? For our purposes the point is not finding the answer but describing the reasoning involved. That is emphasized by asking how many ways of completing it there are. Consider blank cell 4. It cannot be a C so it must be an A or a B.

(a) write out all the restrictions on cells 4, 5, 7, 9 using the notation "N_L" for "cell N has letter L", and symbols for Boolean connectives. These are of two kinds. There are those like the restriction of 4 to B or A just mentioned. There are also conditions that depend on the choices we make for filling in the blanks. For example

if 4 is A then 6 has to be B. (Why?) Write out both kinds.

This is more information than we need to solve the problem. In practice one would start with one condition and not think of others until they were needed. For example consider one of the two possibilities for 4, A. Putting A in 4 means putting B in 6, and thus putting C in 9, and B in 7. (Why?) If we instead put B in 4 then 6 would have to be A, since the row already has a B and a C. But it cannot be an A since its column already has an A. So that option is ruled out and we only have one way of completing the square.

(b) How many ways of completing

1	A	3
4	B	6
7	C	9

are there?

(c) (harder) Find a 3x3 Latin square with exactly 6 blank cells that can be completed in 4 ways.

9) Which of the three truth assignments below does this model give for **p**, **q**, **r**, when **p** is **Aa**, **q** is **Bb**, and **r** is **Cc**?l

	A	**B**	**C**
a	**YES**	**NO**	**YES**
b	**YES**	**NO**	**NO**
c	**YES**	**NO**	**NO**

the truth assignments:

$\mathcal{T}_1$: **p is true, q is true, r is false**

$\mathcal{T}_2$: **p is true, q is false, r is false**

$\mathcal{T}_3$: **0 is true, q is false, q is false**

10) In each of the four cases below give a truth assignment that makes *all* of the propositions in the left column true and the proposition in the right column false.

make these true	while making this false
p ⊃ q	**q ⊃ p**
p v (q & r), q	**r & (p v q)**
~ (p & q), r	**r ⊃ p**
p ⊃ (q ⊃ r), ~p	**~p ⊃ (q ⊃ r)**

11) Facts: Three friends have seen the weather forecast on TV, but they give different reports. Alfie says the forecast was for snow (and nothing else). Betty says the forecast was for fine weather (and nothing else). Gemma says the forecast was for rain (and nothing else). At least one of them is correct, and at least one of them is lying. Moreover, we know that if Gemma is lying then the forecast is for snow. (Her only motive for lying is to go skiing without you.) We know that if the forecast was for fine weather then it is not for rain. (Snow can be fine for skiing.) And if Gemma is not lying then Alfie is telling the truth. (Alfie only dares to lie when Gemma is in it too.)

What can we conclude about the weather forecast? Draw as many conclusions as possible, but express them as concisely as possible, ideally as a single sentence.

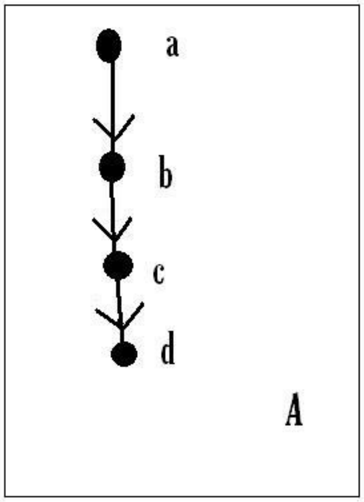
This is an exercise in two skills. The first is simplification, reducing the number of terms in which the facts and the conclusions we draw from them are expressed. The second is listing all the possibilities and eliminating those that do not fit the stated facts. This amounts to formulating the disjunctive normal form for these facts. So to do the exercise you must think first "how can I express it in terms of a small number of propositions", then "which combinations of these can be true given the facts", and then express these in the original terms.

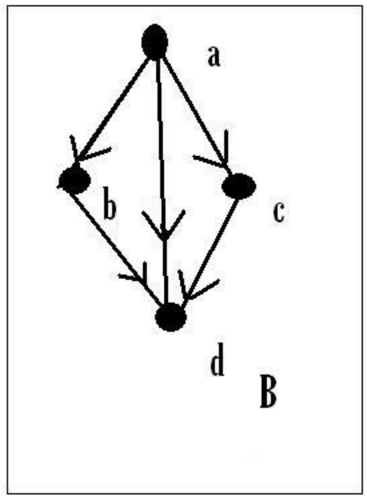
12) Below are arrow diagrams for two relations. Both arrange the individuals **a, b, c, d** in an order, and, using R as a symbol for both relations both give a truth-assignment for atomic propositions relating individuals with **R**.

(a) Which of the following are true in neither model, which true in both, and which in one but not the other?

Rab v Rba	**Rbb**	**(Rab & Rbd) ⊃ Rad**
(Rac & Rcd) ⊃ Rad	**Rad**	**Rab ⊃ Rba**
Rcd ⊃ ~Rdc		**Raa & Rbb & Rcc & Rdd**

(b) State informally (not in symbols) what the general differences between model A and model B are.





B – more

13) fill in the following truth tables

p	**q**	**~p**	**~q**	**p v q**	**p ⊃ q**	**~p & ~q**	**p v ~q**
T	T						
T	F						
F	T						
F	F						

14) **~(p & q & ~(s ⊃ r))** is equivalent to which one**s** of the following

a) **~p & ~ q & ~~(s ⊃ r)**

b) **~p & ~q & (s ⊃ r)**

c) **~p v ~ q v (s ⊃ r)**

d) **~p v ~ q v ~~(s ⊃ r)**

e) ~p **v ~ q v ~s v r**

15) Fill in these truth tables

p	q	~p & q	~(p v ~q)	~p v q
T	T			
T	F			
F	T			
F	F			

a) what does this tell us about the relation between **~p & q** and **~(p v ~q)** ?

b) what does this tell us about the relation between **~p & q** and **~p v q** ?

16) **a)** Suppose **p** and **q ⊃ s** are true, and **r** is false: which of

~(p & r) , **q ⊃ r** , **q & s** , **s ⊃ q**

are true, which are false, and which cannot be decided on this information?

b) Suppose **p** and **q ⊃ s** are false, and **r** is true: which of

~(p & r) , **q ⊃ r** , **q & s** , **s ⊃ q**

are true, which are false, and which cannot be decided on this information?

c) Why is it that the falsity of a conditional such as **q ⊃ s** gives us more information than its truth? (Does this hold for the normal English IF as well as for the ⊃ of logic?)

17) Find truth assignments that make

i) **p** and **p v q** true and **p & q** false

ii) **q** and **p ⊃ q** true and **p** false

iii) **p ⊃ q** true and **~p ⊃ ~q** false

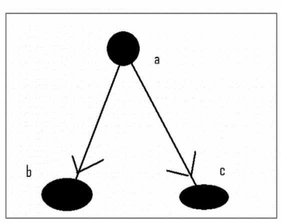
iv) "if you pay you'll be admitted" true and "if you are admitted then you paid" false. (Add a story to make it intuitively clear with the English IF that the one is true and the other false.)

18) The truth assignment 𝒯 makes **p** and **q** true and **r** false. Consider the model below

	A	**B**
a	**T**	**F**
b	**F**	**T**

What atomic propositions in this model can **p**, **q**, and **r** be in order that the model gives the same truth values as 𝒯? (For example, **p** cannot be **Ab**, because that would make **p** false.)

19) Using the arrow diagram below to give truth values to atomic propositions of the form **Rxy**, such as **Raa**, **Rab**, **Rbb**, choose atomic propositions to show that we can **(a)** make ~(A & B) true and ~A V ~B false, **(b)** make A ⊃ B true and **B ⊃ A** false, **(c)** make A ⊃ B true and ~A ⊃ ~B false.



PART II, sections 6 – 9.

A- core

20) For each in the left column say which on its row it is equivalent to.

p & q	**~(~p & ~q)**	**~p ⊃ q**	**~q ⊃ p**	**~p v q**	**~(~p v ~q)**	**~(p & ~q)**
p v q	**~(~p & ~q)**	**~p ⊃ q**	**~q ⊃ p**	**~p v q**	**~(~p v ~q)**	**~(p & ~q)**
p ⊃ q	**~(~p & ~q)**	**~p ⊃ q**	**~q ⊃ p**	**~p v q**	**~(~p v ~q)**	**~(p & ~q)**

21) Here are some English sentences. Which are equivalent?

a) Mo is sad and Bo is not sad.

b) Mo is not sad and Bo is not sad.

c) One of these is true: Mo is sad, Bo is sad.

d) One of these is false: Mo is sad, Bo is sad.

e) These are both false: Mo is sad, Bo is sad.

f) These are both true: Mo is sad, Bo is sad.

g) Mo but not Bo is sad.

h) Mo is sad and Bo is sad.

i) Mo is sad or Bo is sad.

j) Neither Mo nor Bo is sad.

22) Which of the following propositions is equivalent to which others? (Much of this is just reproducing what is in the chapter. But it is important to get these equivalences drilled into your head. It really helps not just to know them but to see how each makes sense. Some of them are new, not mentioned in the chapter, and may be surprising to you. Learn them, if they don't seem obvious.)

a) **p v q** b) **~(p v q)** c) **p ⊃ q**

d) **~(p & q)** e) **~p ⊃ q** f) **p & q**

g) **(~p & ~q)** h) **~(~p v ~q)** i) **~q ⊃ p**

j) **~(p & ~q)**	k) **~p v q**	l) **~~p**
m) **p**	n) **~p & ~q**	o) **~p v ~q**

23) Which of the propositions in $\mathcal{A}$ below are equivalent to which in $\mathcal{B}$?

$\mathcal{A}$

~p & q	**~p v q**
~(~p & ~q)	**~p v (q & r)**
~p & ~q &~r	**~p & (~q v ~r)**
~(p & ~q)	**~(p & q) & ~(r & ~s)**

$\mathcal{B}$

p & q	**~(p & ~q)**
~p v q	**~ (p v q v r)**
~(p & (~q v ~r)	**~(A v (B & C))**
~(p v (q & r)	**~ ((p & q) v (r & ~s))**
p v q	**~(p v ~q)**

24) (a) give truth tables for the following.

(i) **p ⊃ (q ⊃ p)**	(ii) **p ⊃ (p ⊃ q)**	(iii) **(p ⊃ p) ⊃ q**
(iv) **(p ⊃ q) & (q⊃ q)**	(v) **(p⊃ p) ⊃ (q⊃ q)**	(vi) **(p v ~p) ⊃ q**
(vii) **p & (p ⊃ q) & ~q**	(viii) **(p⊃ q) v ~(q⊃ p)**	(ix) **~q ⊃ q**
(x) **p & (p ⊃ q) & ~p**	(xi) **p ⊃ ~p**	(xiii) **(q & ~q)⊃ p**

(b) which are equivalent to which others?

(c) which are tautologies?

(d) which are contradictions?

(e) if we took ⊃ to be the "if" of everyday English some of these might seem rather implausible. That is, it would be far from obvious that the tautologies were always true, the contradictions always false, or that the equivalent propositions always had the same truth value. Which of these seem to you in this way implausible?

25) For each sentence in the left-most column say which sentences on its row it is equivalent to.

~(p & q)	**~p & ~q**	**p & ~q**	**p**	**~p v ~q**
~(p v q)	**~p & ~q**	**p & ~q**	**p**	**~p v ~q**
~(p ⊃ q)	**~p & ~q**	**p & ~q**	**p**	**~p v ~q**
~~p	**~p & ~q**	**p & ~q**	**p**	**~p v ~q**

25) There are other cases analogous to de Morgan's laws. Show with truth tables that

~p ⊃ ~q is not equivalent to **~(p ⊃ q)** but is equivalent to **q ⊃ r** .

On the other hand, show with truth tables that **p ↔ q** *is* equivalent to ~**p ↔ ~q** .

26) Although **~(p & q)** is NOT equivalent to **~p & ~q**, and **~(p v q)** is NOT equivalent to **~p v ~q** , when **~p & ~q** is true then **~(p & q)** is true, and when **~(p v q)** is true then **~p v ~q** is true.

a) Mark on a truth table all the rows where **~p & ~q** is true and check that **~(p & q)** is also true on them.

b) Mark on the truth table rows where **~(p & q)** is true but **~p & ~q** is not true.

c) Mark on a truth table all the rows where **~(p v q)** is true and check that **~p v ~q** is true on them, and

d) mark on the truth table rows where **~p v ~q** is true but **~(p v q)** is false.

(This question anticipates the idea of logical consequence, which is the topic of chapter six.)

26) What are the disjunctive normal forms for

p ↔ q , p & (q v r) , p & (q v (r & s)) , ((p ⊃ q) ⊃ r) ⊃ p ?

(I am using A ↔ B as an abbreviation for (A ⊃ B) & (B ⊃ A).)

(Some of these are chosen because they have formulas within the scope of formulas within the scope of formulas, but the DNF theorem shows that we can always find an equivalent formula with just conjunctions within the scope of disjunctions. Put this way, it is somewhat surprising.)

27) How would you rephrase these searches so that they are manageable on the internet?

(a) you want data on animals that are not cats, and their viral diseases

(b) you want data on cat viruses, except those associated with the flu

(c) you want data on viruses, but for viruses affecting cats you only want information on flu viruses

(d) you want data on flu viruses, but not on those affecting cats.

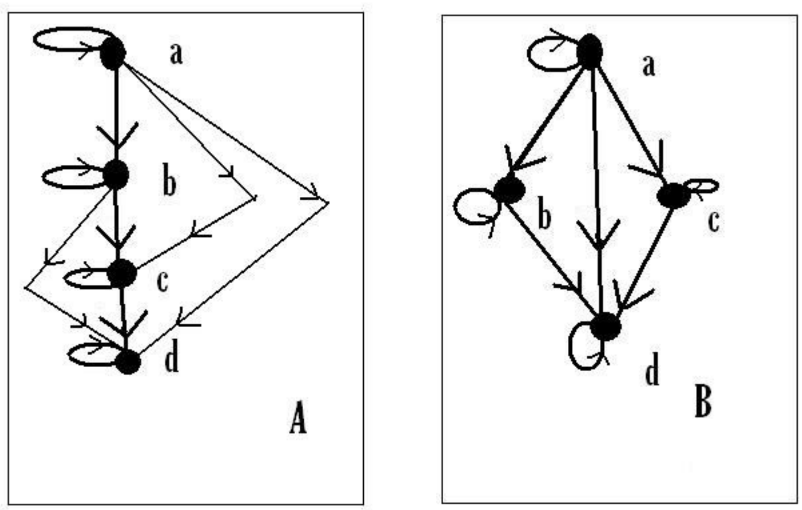
(e) you want data on viruses but not data that is both on cats and flu

28) **(a)** Which of the following are true in neither, both, or one of models A and B below. (If just one say which.)

Rab v Rba	**Rbb**	**(Rab & Rbd) ⊃ Rad**
(Rac & Rcd) ⊃ Rad	**Rad**	**Rab ⊃ Rba**
Rcd ⊃ ~Rdc	**(Rab v Rba) & (Rbc v Rcb) & (Rcd v Rdc)**	

(b) State informally (not in symbols) what the general differences between model A and model B are.

(This is a development of question 11, but the models are more complicated.)



29) You are searching for a tablet on the internet, using a search engine that will not allow you to put an AND or OR within the scope of another AND or OR. And it will not allow you to use IF at all. So all of the following are *not* allowed.

(LongBattery **OR Q**uickCharge**) AND (M**ovieCam **OR M**anyApps**)**

IF (NOT FreeData**) THEN B**igMemory

(IF NO USB **THEN S**Dcard**) AND EITHER P**retty **OR T**ough

Either (I-pad **AND O**nSale**) OR (NOT K**indle **AND A**ndroid**)**

Reword the queries so you can do them.

30) In which of the two models below is "Either Jo or Fred was married to both Thelma and Louise" true (taking it on its usual meaning)? In which is "Both Jo and Fred were married to either Thelma or Louise"?

(a)

was **M**arried to	**t**helma	**l**ouise
jo	**T**	**F**
fred	**F**	**T**

b)

Married to	**t**helma	**l**ouise
jo	**F**	**T**
fred	**T**	**T**

Can you make a model in which both are true?

C- harder

31) a) Show that A **&** B and C **v** D can be defined in terms of ⊃ and **v**, using equivalences like those of section 6 of this chapter. Show the same for ⊃ and **&** in terms of **v** and **~**. And ⊃ and **v** in terms of **&** and **~.**) So given any of the pairs of connectives ⊃ and **v** , **v** and **~**, or **&** and **~** we can define all of **~**, **&, v**, and ⊃.

b) Show that using any of these we can define any connective that can be given by a truth table.

c) Show that using the connective | mentioned in section 2, and defined by

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

We can define both members of each of these pairs, and therefore all truth functional connectives.

d) What Boolean connectives, definable by truth tables, are there that have not been mentioned? (So besides **~**, **&**, **v**, ⊃, **|**, and ↔.)

32) Here are two equivalences that most people do not find obvious

(p & q) ⊃ r is equivalent to **(p ⊃ r) v (q ⊃ r)**

(p v q) ⊃ r is equivalent to **(p ⊃ r) & (q ⊃ r)**

Give a couple of examples to make them not so surprising.

How might these relate to the and/or confusions that are common in everyday speech? (They are a more general form of de Morgan's laws, in that **~A** is equivalent to **A ⊃ (A & ~A)** .)

33) In this book as in all introductory logic, we assume that there are just two truth values, True and False. As an old Latin tag has it "tertium non datur" — there is no third truth value. This can seem wrong. Consider "He is an adult", said of a 17 year old. He is on the borderline between childhood and adulthood, so "yes and no" or maybe "not yes and not no" are possible, though confusing, answers. What would truth tables for a third truth value look like? How could one fill in the blanks for

A	**~A**
T	
F	
N	

where N is a third truth value, distinct from both truth and falsity, but "between" them, and **~** is plausible as "not"?

34) Prove by mathematical induction on the number of lines of the truth table that every formula is equivalent to one in disjunctive normal form.

35) Besides disjunctive normal form there is also conjunctive normal form, where a formula consists in a series of conjunctions, each of which is a disjunction of an atomic proposition or its negation. Prove that every formula is equivalent to one in conjunctive normal form. (This result seems to me less intuitively plausible than the disjunctive normal form theorem, though it is certainly true in propositional logic.)
(Hint: it follows from the disjunctive normal form theorem using the fact that
(A & B) v C is equivalent to (A v C) & (B v C).)

36) Prove by mathematical induction on the number of lines of the truth table that every formula is equivalent to one in disjunctive normal form.

37) **p v (q v ~p)** is a tautology, true on all lines of its truth table, and **p & (q & ~p)** is a contradiction, true on none of them. **p v q** is true on three of the four, and **p & q** on one of them. Is there some sense in which this determines which is more likely to be true than which? Is there some idea of probability where facts such as these determine which propositions are more probable than which others?

chapter six: logical consequence

6:1 (of 8) logical consequence

The related concepts of *logical consequence, deduction, and valid argument* are central to logic. They are also very easy to misunderstand. Section 4 below discusses why it is easy to misunderstand them and how they are related. We have already discussed the analog of logical consequence for queries in the form of the width of a query. One query is intrinsically wider than another when it will get at least as many results in any model/database. The discussion of search trees showed that this is in basic respects like the relation of width between sentences. Width between sentences, taken as searches for models, is called logical consequence: one (true or false assertive) sentence S is a logical consequence of a set of sentences $S_1,\ldots,S_n$, when S is true in all models which make all of $S_1,\ldots, S_n$ true. For example, "there is a mouse in the bathroom" is a logical consequence of the set {if there are mouse-droppings in the bathroom then there is a mouse in the bathroom, there are mouse droppings in the bathroom}. If a model makes both these *premises* true then it will have to make the conclusion true.

In this chapter all the examples will work with logical consequence that depends just on the Boolean connectives, so we can simplify models by taking them to be truth assignments. Some of the examples will use the propositions of propositional logic, and others will be English sentences with a loose indication of their truth assignments.

The mouse in the bathroom argument can be represented as a search tree like those we have already seen.

```
              droppings
              droppings ⊃ mouse
               droppings
                 /      \
    ~droppings         mouse
          X
```

The only unblocked branch has droppings and mouse. So, speaking intuitively and in a preliminary way, if we are searching for ways to make both the premises true then any model we find that does this will also make "there is a mouse in the bathroom" true. It is a logical consequence of them, parallel to the fact that if you searched for rooms with droppings with the constraint that you were only interested in mouse droppings, then you would only find rooms with mouse droppings.

Logical consequence has an intuitive connection with good reasoning. (But there are warnings later in this chapter about taking the connection to hastily.) We can see why it is appealing to make the connection with the same example. Assume that a person at any time knows some facts and there are many others that they do not know. So there is a range of possibilities — models of reality — that they think possible, and they would like to narrow this down to a smaller range. Suppose that someone has observed mouse droppings. So they can eliminate all the models which do not represent the kitchen as having mouse droppings. They also know that where there are parts droppings there are mice, so they can also eliminate models where there are droppings but no mouse. Put the two eliminations together and the only models that are left are models with mice. So reasoning in accordance with logical consequence in this case fits well with a sensible pattern of thinking.

S_1,…, S_n are the *assumptions* or *premises*, and S is the *conclusion*. I have written this to allow for any number of premises, but usually we will consider just one, two, or sometimes three premises. The transition from the premises to the conclusion is called an *argument*, and if S really is a logical consequence of S_1,…, S_n then the argument is called *logically valid*. This vocabulary of assumptions and valid argument can be misleading, as I will explain. So think in terms of the exact definition with models, or in terms of the looser thought that an argument is logically valid when its conclusion, thought of as as a query, will find all the models that its premises, taken together, will find.

>> describe cases where someone can *use* an argument, in this sense, although they are not *having* an argument with anyone, and cases where an argument can be valid although what is being argued is ridiculous.

In this chapter and the following two we investigate the relation of logical consequence in propositional logic, so we are interested in when one proposition is true under all truth assignments that make the propositions in a set of assumptions true. Sometimes two propositions will be logical consequences of each other, and then they are *logically equivalent*. We saw this already with logically equivalent propositions which are true on the same lines of truth tables, such as **p & q** and **~(~p v ~q)**, or with the equivalence of any proposition to its disjunctive normal form.

So remember these two basic definitions (which can apply to all sentences capable of being true or false, and not only to propositions of propositional logic.)

One sentence is a logical consequence of others when it is true in all models in which all of them are true.

Two sentences are logically equivalent when they are true in all the same models.

6:2 (of 8) hidden information games

There are many games in which one has to deduce what a situation is, given clues. In a typical game the player has to make a choice given partial information. If the player is lucky she can go on to make further choices, eventually either arriving at a winning or a losing situation. So the important question for the player is which situations are consistent with the information given. One example is *Minesweeper*, standard on most PCs (and available for Macintosh computers). In *Minesweeper* the player has to click on blank cells: if there is a mine present the player loses, but if the player can click on all the cells that do not have mines the player wins. Clues are provided in the form of numbers indicating how many mines there are in cells neighbouring a particular cell. So the agent's problem is to deduce the location of mines from these cells — to see when "mine in cell C" is a logical consequence of the available information — and to avoid those cells, while choosing from the possibilities consistent with the information that do not involve mines. Standard *Minesweeper* works on a two-dimensional array. (There is a three-dimensional version.) The important points can be illustrated with a 1-D version. The player is given an array, for example

1		1	

She knows that the second cell from the left has to have a mine, because there is 1 mine in a cell neighbouring the leftmost. So two arrays consistent with that 1 on the left are.

1	*	1	

1	*	1	*

We have not used all the information yet, though. In the original array there is a 1 in the second cell from the right, but this makes the second of these two models impossible. So the player is left with the first model. She can safely click on the blank cell without revealing a mine.

When we first play a game like this we reason by forming and eliminating models as I have been describing. This is slow, though. So as we get practice with one we make the thinking more automatic, taking advantage of the spatial presentation of the information to develop quick routines for getting to the same conclusions. It would be impossible to play quickly without doing this, and a good player develops a large range of short-cut routines. It is interesting to notice in one's own case the play between explicitly eliminating models and taking spatial short-cuts. I think it gives one some insight into one's own thinking.

>> granted that we search for models in these games, using available information, what should we consider as the premises of the arguments?

>> give examples from grade school arithmetic where you learn a procedure by thinking and later perform it in a quicker and less thoughtful way. when do you go back to thinking about it?

6:3 (of 8) counter-models

It might seem to be very hard to tell if one sentence is a logical consequence of some other sentences. After all, we have to tell if it is true in absolutely all models which make them true, and there is an enormous variety of such possible models. And in logic in general it can indeed be very hard. But in the special case where the sentences are constructed with Boolean connectives there are shortcuts. One is truth tables, as we have seen. The other is formal deductions, which we study in chapter 7 in the form of derivations. Truth tables are instances of the *semantic approach*, building on the connection between logical consequence and models. Deductions and derivations are instances of the *syntactic approach*, using considerations about how formulas are structured to get rules for making convincing arguments leading from premises to conclusions. Logicians are most satisfied when they can back up what they say about logical consequence with considerations of with both kinds.

We can often avoid considering all possible models, though, when we ask whether a sentence is *not* a logical consequence of some others. To show this all we have to do is to find just one model that makes the premises true and the conclusion false. When we do this we are looking for a *counter-model*, a model that shows that one sentence is not a logical consequence of some others. Finding counter models is a more intuitive business than proving logical consequence, and getting a feel for finding them is very useful in evaluating the validity of arguments. For one thing, we are searching for just one model, which is to make the premises true while making the conclusion force. This is simpler than considering all the models for the premises, and does not involve us in tricky thinking along the lines of "if the conclusion is true in all models from this particular set when the premises are true in them then we do not need to consider any further models." To see how it works, consider some examples.

Suppose we have the following premises:

Prue is rich or Quinn is lying
If Prue is rich then Quinn is lying
If Quinn is lying then Rick is happy

Write these as:

p v q
p ⊃ q
q ⊃ ~r

We have the following candidate conclusions; which ones really are logical consequences of the three premises?

Rick is happy
Prue is not rich
Rick is not happy

Write these as: **r, ~p, ~r**

Think about these for a while, until you feel clear about the question. Are any of these such that you can be sure of their truth just on the basis of the information found in the premises?

(Think, think, think, …. Pause right here before going on.)

Well, what about the first one? Could it be that one? One way to think this through is to tell yourself a story involving Prue and Quinn, trying to make it come out so that these premises are true but Rick is *not* happy. That isn't hard: tell the story so that Prue is not rich and Quinn, who says she is, is lying, so if Prue is rich Quinn is lying. Moreover the only way to make Rick happy would be for his aunt Prue to get a lot of money. But Prue stays poor, so Rick never gets to be happy. The premises are true — check that each of them is — but it is not true that Rick is happy. So it is not a logical consequence of these premises: they can be true in ways that do not make it true. On this story, moreover, Prue is not rich, so we also have a case where the premises are true and the "conclusion" that Prue is rich is false. So that is not a logical consequence either.

We would have a harder time telling a story in which the premises are true and the third candidate conclusion, that Rick is not happy is false. This might make us suspect that it cannot be done, and this conclusion is a logical consequence of the premises. In fact, it is. But trying to show this by exploring many stories would be time-consuming and inconclusive. But there is a better way, when we are dealing with sentences of propositional logic, to use truth tables. For three atomic propositions, **p**, **q**, **r** the truth table is:

p	**q**	**r**	**p v q**	**p ⊃ q**	**q ⊃ ~r**	**~p**	**~r**	
T	T	T	T	T	F	F	F	
T	T	F	T	T	T	F	T	*
T	F	T	T	F	T	F	F	
T	F	F	T	F	T	F	T	
F	T	T	T	T	F	T	F	
F	T	F	T	T	T	T	T	
F	F	T	F	T	T	T	F	
F	F	F	F	T	T	T	T	

The only column of this truth table that you may need to think about is the one for **q** ⊃ **~r** . The truth values in this column arise because the conditional is true except when the antecedent is true and the conclusion is false, and that is when **q** is T and **r** is also T, so just those two cells are F. Looking at the table we can identify the two rows where all the premises are true. (I have highlighted them.) On one of them (*), both **r** and **p** are false, so neither of these is a logical consequence of the premises. But on both of them ~r is true, so it is a logical consequence of these premises.

Using truth tables as a mechanical way of presenting the truth assignments that are generated by models, we see that for sentences of propositional logic, where we can ignore some features of models, we have an automatic way of testing whether a conclusion is a logical consequence of a set of premises. It has probably struck you that this truth table involves quite a lot of writing to come up with just two rows where all the premises are true. These arguments involved just three atomic propositions, and thus eight rows to capture all their possible combinations. If we had four atomic propositions there would be 16 rows, and if five then 32 rows. (if n then2^n.) Truth tables can become too large to be practical ways of testing arguments. The tree method of the next two chapters, which is a development of the search trees of chapter 4, can be seen as giving a shortcut. But truth tables do illustrate a pair of basic points: to show that a conclusion is not a logical consequence of some premises we find a way that the premises can be true but the conclusion false, and to show that a conclusion is a logical consequence of some premises we have to survey all the ways that the premises can be true and make sure that the conclusion is true on all of them.

>> what are the advantages and disadvantages of the storytelling method and the truth table method compared to each other?

There are two general ways of showing that a conclusion is or is not a logical consequence of some premises. One is what we call semantic, involving concepts such as truth and models. When we use truth tables we are thinking in semantic terms. The other is what we call syntactic, involving definite patterns of valid argument to make careful deductions of conclusions from premises. It is best when these two fit together. We want to know that our deductions never give us conclusions that are not true in all models of the premises. And we would like to have patterns of deduction to serve as backups or confirmation of considerations about models. It is much easier to see how the two sides fit together when we are dealing with a formal logical language than with every day spoken language. It is best to keep both the semantic and syntactic approach — models and careful argument — in mind, and to use each as a check on the other. It is important to get used to finding counter-models to claims that one sentence is a logical consequence of some premises. And it is important to begin getting a sense of when there is no counter-model because the sentence really is a logical consequence. Some of the exercises at the end of this chapter are there to give you more practice in this.

6:4 (of 8) consequence, deduction, and argument

When we reason we often move from premises, which we either believe or have assumed in order to see what follows from them, to conclusions, trying to use only information that is contained in the premises. Much of mathematics, philosophy, and law, consists of such reasoning. In practice we nearly always make use of more information than we explicitly state as premises, but the aim is still to stick as closely as we can to our stated assumptions. We state such reasoning in the form of *arguments*, in which we state our premises and then go from them to conclusions and then on to further conclusions, so that eventually we can say "Therefore P" where P may be "$e^{\pi i}$ = -1" or "God exists" or "my client is not guilty". An argument whose conclusion is a logical consequence of its premises is known as a *logically valid argument*. And studying logic is *supposed* to make people into sharp reasoners and persuasive arguers.

It is often not obvious whether a conclusion is a logical consequence of some premises, whether the argument from the premises to the conclusion is logically valid. In fact, it is often very far from obvious. So not seeing that a conclusion is a logical consequence, or as we often say that it follows, is not a sign of stupidity. It sometimes takes centuries for thinkers to see that something follows. Going beyond propositional logic, for example to the logic of quantifiers in part three, this is often because things get very subtle and complicated. But even in propositional logic consequence is often not obvious. For example, is **(pvq)** ⊃ **r** a consequence of **(p** ⊃ **r) v (q** ⊃ **r)**? I doubt that it is obvious to you, although in fact it is a consequence. In response, logicians have developed systems of *deduction*, which are step-by-step rules guaranteed to give only logical consequences.

Following the rules, we *deduce* conclusions from premises. The derivations of subsequent chapters have their origin in traditional systems of deduction. So if you stick to the rules you are safe. The technical term for this property of the rules is that they are *sound*. Rules can also be *complete*, meaning that they capture all the logical consequences can be described in their vocabulary. Completeness is a sophisticated topic and I shall barely discuss it. But it grapples with a very profound problem, that logical consequence is a richer and more complicated matter than we can easily grasp.

A logically valid argument does not have to have a true conclusion. Consider "the earth is flat, therefore the earth is flat", or "the earth is flat & fish swim, therefore the earth is flat", or "if grass is green then the earth is flat, grass is green, therefore the earth is flat". In each of these the conclusion is a logical consequence of the premise or premises: if the premises are true the conclusion has to be true. But the premises are not all true and neither is the conclusion.

A logically *in*valid argument, one where the conclusion is not a logical consequence of the premises, does not have to have a false conclusion. Consider "cats chase mice, therefore eagles fly", or "cats can fly, therefore eagles fly". In neither case is the conclusion a logical consequence of the premise, but in both of them the conclusion is true. So if you show that someone's argument for a conclusion is invalid you are not showing that the conclusion is false. The most you can be showing is that they have not given a good reason for believing it. There are exercises at the end of this chapter to reinforce this point.

It is important to realise that the conclusion of an argument that is not logically valid can be true. One reason is that the fact that though the premises may not show that a conclusion *must* be true, they can be evidence that it is true. For example if a common disease D has symptoms A, B, C and a very rare disease has the same symptoms, then though the argument "If D then A & B & C , A & B & C, therefore D" is not logically valid, A & B & C may give you reasons for expecting D. If you reason from symptoms of common diseases to diagnoses of the diseases you will sometimes get a false conclusion, but you will often be right.

>> (for philosophy students) relate this to Hume's problem about induction.

Be wary of the word “valid” here. We often use “valid” in conversation to mean “true”. (“You have a valid point there” = “What you say is true.”) But in logic we apply “valid” not to statements that can be true or false but to *arguments*, which are sequences of sentences. They cannot be true or false; they can just join sentences in a way that follows logical consequence or does not. Logicians also talk of “sound” arguments, which are valid arguments that have true premises. A sound argument will have a true conclusion, since the conclusion has to be true given that the premises are. The connections between truth, logical consequence, and good reasons for believing conclusions are pretty tricky.

This a very confusing topic. There is no clear evidence that learning about logical consequence makes people better at reasoning in general or at persuading others. In fact there is disturbing evidence that teaching students logic has very little impact on them, not even on their recognition of which arguments are deductively valid and which are invalid. Students learn to do well on the material used as examples, and then if they are tested with even slightly varied examples they often seem to be answering at random.

Psychologists are puzzled by this, and logic teachers find it too depressing to think about.

Here is my explanation of what is going on. When we reason or persuade we normally do so against a background of an enormous amount of information that we are taking for granted, or which both parties in a discussion accept. You and I may have deep disagreements about religion and politics and movies, but when we discuss these or any other topics we each assume, and assume that the other assumes, that Alberta is to the north of Texas, that grass is green in springtime, that the English language originated in England, that X is a terrible actor, and all sorts of other obvious information. And when you reason all by yourself you also assume all these things. We have discussions to persuade one another of little gaps in our shared assumptions, and we reason individually to fill in gaps in our individual knowledge. There are many such gaps: anyone can immediately produce a long list of questions to which their honest answer would be "I don't know". But they are much smaller than the amount of information that we take for granted. (We may not appreciate the amount of information that is taken for granted, because we usually do not bother to mention it, and we are usually not consciously aware that we are assuming it in our reasoning.) Given all this assumed information, we can reason from it to new conclusions and we usually do so carefully, sensibly, in ways that we would describe as reasonable and logical, and in ways that will be clear to the people we want to persuade.

But that's where the trouble begins. Since we think of this familiar business — filling in gaps in terms of an enormous assumed background — as logical reasoning, when we are asked if something is a logical consequence of some assumptions, or whether it follows logically, or whether an argument is logically valid, we answer in terms of what we would count as reasonable and systematic gap-filling reasoning. That is a natural reaction, given what is familiar and the way we normally speak. But *that is not what logical consequence, or a logically valid argument, is*. It is in a way the opposite. Not filling in gaps against a large assumed background, but given a small amount of information filling in a few of the remaining blank spaces assuming no background at all. That is not a familiar intellectual activity, for most of us. I think that the least confusing way to learn it is to treat it as something unfamiliar and unnatural, to be approached slowly and carefully, like some strange monster that has washed up on the shore.

I am skeptical of the idea that studying logic gives one an inside track to good reasoning or persuasive argument. That is one reason that I do not begin a logic course with a discussion of valid argument, but instead first explain basic ideas of logical form, in particular the ideas of Boolean connectives, and the relation between a sentence and a model/database. These ideas can be motivated well enough by their connections with searching, and by how they help us understand how complex language works, without needing to justify them with dubious psychology. And the very unnaturalness of thinking in terms of logical consequence is one of the reasons for studying it: it helps to acquire the skill of thinking not in terms of loose connections between large bodies of information but in terms of precise connections between small bodies of information.

>> what is your impression, from this point in the course, of the relation between the study of logic and the everyday quality that we describe as “being logical”.

All the same, there are connections between logical consequence and effective argument. One connection is that when one sentence is a logical consequence of some premises it is safe to argue from the premises to the conclusion: given that the premises are true you can be sure that the conclusion will be. Another connection is that some *fallacies*, some patterns of argument where the conclusion is *not* a logical consequence of the premises, can interfere with good reasoning and effective argument.

6:5 (of 8) an example: inferring disjunctions

This is a logically valid argument: A therefore A or B. (So is B therefore A or B.) So the following are, surprisingly, logically valid

> It is raining. Therefore either it is raining or there is life on Mars.
> Cats can fly. Therefore either I am living in the 21st century or cats can fly.

But these seem like very puzzling arguments, to many people. They do not seem like good ways of reasoning. The first seems to introduce something completely irrelevant to the premise, that may well be false. And the second seems to go from an obvious falsehood to something that is weird and silly but on reflection true. In fact, they are both logically valid *and* they are silly ways of reasoning. Some valid arguments are silly. No one would waste their time and mental energy thinking along these lines. BUT suppose that it is raining: then it is true that either it is raining or there is life on Mars. If there is life on Mars then "either rain here or life on Mars" is true because it is raining, and if there is no life on Mars then "either rain here or life on Mars" is true because it is raining. And suppose that hidden in a valley in the Andes there are flying cats. Then it is true that either I am living in the 21st century or there are flying cats (since both of the disjuncts are true). So both of these arguments meet the definition of logical consequence: if the premise is true then the conclusion is.

But there are arguments of the form A therefore A or B that it does make sense to use. Suppose we assume that if you live in Canada or the US then your greatest danger of flu is in the winter. Assume also that Bo lives in Canada. Then we can reason

> Bo lives in Canada. Therefore Bo lives in Canada or Bo lives in the US. If he lives in Canada or the US then his greatest danger of flu is in the winter. Therefore Bo's greatest danger of flu is in the winter

Or consider the following argument

> If an animal is a rabbit or a squirrel the vet will only see it on Thursday
> Smartie is a squirrel.
> (Therefore) Smartie is a rabbit or a squirrel
> Therefore: the vet will only see Smartie on Thursday.

So we do sometimes reason from A to A or B. In fact we do often. But although all such arguments are valid, many would be unhelpful time-wasting ways to reason. We do not try to formulate rules of logic so as to catch only the worthwhile arguments. That would make logic very complicated. But the price we pay for keeping it simple is that we have to include some arguments that at first sight seem silly, in fact, some that are just plain silly.

The validity of A therefore A or B makes sense when we see sentences as searches for models. To find all the models in which "it is raining" is true, you can instead use the wider search for all the models in which "either it is raining or there is life on Mars" is true. It is a search which will get you all the models you wanted, plus some more. This is often too many more to make it a practical way of searching, but we cannot know that just from logic. Similarly, reasoning from "A" to "A or B" will always give a true conclusion when the premise is true, but it is often not a useful or helpful true conclusion[16].

16 There is a branch of advanced logic called "relevance logic", which tries to describe patterns of deduction where there are always definite connections between premises and conclusions. It is formally interesting and philosophically controversial..

6:6 (of 8) five examples of valid argument

Here are five among the many patterns of valid argument. We will see more in later chapters. They are not always sensible ways to reason, as noted. Note how we write the patterns with the assumptions above a horizontal line and the conclusion below it. This is standard. The patterns have standard names, which I have written below them.

and-elimination	or-introduction	elimination of alternatives
A & B / **A**	**A** / **A v B**	**A v B**, **~A** / **B**

modus ponens	modus tollens
A, **A ⊃ B** / **B**	**A ⊃ B**, **~B** / **~A**

These should become as familiar to you as facts of simple arithmetic. And they are as simple and obvious, really. Consider the last two. *Modus ponens*[17] is valid because **A ⊃ B** is true only in three cases: when **A** and **B** are both true, when **A** is false and **B** is true, and when **A** and **B** are both false. (This is what the truth table for ⊃ says.) Of these **A** is true only in the first, when **B** is also true. *Modus tollens* is valid because the only one of these truth assignments making true in which **B** is false (so **~B** is true) is the last, when **A** and **B** are both false.

There are also frequently occurring patterns of invalid argument, mistakes people often make, and some of them have names. But I am not going to give them or their names. Years later you would just remember that you had been taught them, and think that they

17 The names *modus ponens* and *modus tollens* go back a long way, to Latin phrases meaning the way of laying down an argument and the way of refuting it.

were valid. When an invalid pattern appears, you are going to have to think about it and search for a counterexample. For example the argument "A v B therefore A" is invalid, because by making A false and B true we can make its premise, A v B, true while its conclusion, A is false. And this should make sense, since A v B just says that one of {A, B} is true, without telling which. There are exercises at the end of this chapter to help you recognise invalid arguments, as well as valid ones. They are all confined to very simple cases, because for more complicated cases it helps to have the resources of later chapters.

>> is it always a mistake to use an argument that is not valid?

>> is there a way in which these *are* facts of simple arithmetic?

6:7 (of 8) consequence for conditionals

We can make quite complicated chains of deduction using just ⊃ and ~. Moreover these correspond well to reasoning that we often find convincing with ordinary English IF and NOT. They are useful for getting the idea of a deductive argument and prepare the ground for the discussion of derivations in the following two chapters.

We can do a lot with just the two principles of modus ponens and modus tollens. Each has a search tree. (Search trees will be magically transformed and renamed as derivations in the following two chapters.) The two trees are:.

```
       A                              ~B
      A ⊃ B                          A ⊃ B
     ______                         ______
      /  \                           /  \
    ~A    B                      ~A    B
     X                                    X
  (modus ponens)                 (modus tollens)
```

These are search trees with several sentences at their base, as described in chapter 4, section 4. Notice how similar they are: one associates a contradiction with A, and the other with B. And this makes sense: intuitively only the possibility involving A's truth is compatible with the assumptions of *modus ponens*, and only the possibility involving A's falsity is compatible with the assumptions of *modus tollens*. Moreover each can be turned into a very natural argument in ordinary English. *Modus ponens* can be taken as saying "A is true and moreover if A is true then B is true, which leaves only two possibilities: either A is false or B is true. The first of these is ruled out because we are assuming that A is true, so that leaves only the second possibility, B." *Modus tollens* can be taken as saying "B is false and moreover if A is true then B is true, which leaves only two possibilities: either A is false or B is true. The second of these is ruled out because we are assuming that B is false, so that leaves only the second possibility, that A is false."

I have stated these two patterns of logical consequence in the abstract, using letters instead of sentences, to make it easier to see their general form. But it is also important to be able to recognize and construct examples in ordinary English where sooner or later you will have to make and evaluate arguments. Begin with "it will rain" for A and "the dam will burst" for B. We then get as premises "it will rain. If it will rain then the dam will burst", leading to what I hope is the obvious conclusion "the dam will burst", by notice ponens. In the same way, "the dam will not burst" and "if it will rain then the dam will burst" lead to the conclusion "it will not rain". (Because if it did rain the dam burst and we are assuming that the dam will not burst.)

Somewhat less obvious conclusions and come by putting the patterns together in chains. The simplest chain is just two instances of notice ponens stuck together. Putting it abstractly, and providing a tree, the pattern is:

```
     A
     A ⊃ B
 ____B ⊃ C___
     /    \
 ~A        B
  X       /  \
      ~B    C  (*)
       X
```

Notice how the branch at (*) is got by using the premise B ⊃ C. Extended chains of logical consequence like this will require us to "bring down" premises from the base of the tree. An instance of an argument with this pattern might be "It will rain. If it will rain then the dam will burst. If the dam will burst the village will be flooded. Therefore the village will be flooded.”

>> supply reasoning as in the rain/dam burst examples above, to show how this is intuitively correct.

>> I have worded these examples so that the English sentences best fit the versions with letters. but this does not make the English completely natural. improve it. it is also stimulating to ask what this suggests about features of English that are ignored in logic as we have studied it.

We can get more complicated arguments if we augment modus ponens and modus tollens with another principle, contraposition. Contraposition takes as premise the single sentence A ⊃ B and has the conclusion ~B ⊃ ~A. Assuming that if A is true then B is also true, then if B is not true, A cannot be true. If we assume that if it rains the dam bursts then we also know that if the dam does not burst there is no rain.

>> contraposition is similar to modus tollens. (how?) but if you try to make a tree for it as we did for the previous three principles, you will run into a problem. what is the problem? (this problem will be resolved in chapter 7.)

Very often when we can get a conclusion from a set of premises by using contraposition we can also get it by using modus tollens. But not always. Here is an example where contraposition is needed. I give a tree to make it clear that the conclusion is a logical consequence of the premises.

$$\begin{array}{ll} (\sim p \supset \sim q) \supset r & \\ \underline{q \supset p \qquad\qquad} & \\ \sim p \supset \sim q & \text{(by contraposition)} \\ r & \text{(by modus ponens)} \end{array}$$

Notice how in this one we apply the general principle of modus ponens to the particular case where A is **~p ⊃ ~q** and B is **r**.

An example of this case in ordinary English would be "if he will not work unless we have given him a raise then we should fire him; if he works then we have given him a raise. Therefore we should fire him." Notice how we use "A unless B" to say "if not B then not A". There are many quite complicated logical consequences involving the material conditional. In fact, we seem to use the conditional quite a lot to express complicated patterns of consequence. (This is a little surprising given how many meanings the English word "if" can express.) But to state the full range of them we will need the resources of the next two chapters.

6:8 (of 8) extra: AND inside OR, OR inside AND

We often express a conjunction where each conjunct is a disjunction, or a disjunction where each disjunct is a conjunction. This is not surprising given the significance of disjunctive and conjunctive normal forms, which we met in the previous chapter and will meet again. The idea we are expressing then has the form

(A v B) & (C v D), or (A & B) v (C & D).

Often the words we use hide this somewhat, though it is an example of the kind of symbol-like idiom I am encouraging. For example we say "Both Bo and Mo ski at either

Whistler or Vail", or "Either Bo or Mo skis at both Whistler and Vail." Or talking of these people and these places we might say "Both people ski at one of these places", or "One of them skis at both". The *both/either* one says "(Mo at W v Mo at B) & (Bo at W v Bo at B)": the central connective is AND. The *either/both* one says "(Mo at W & Mo at B) v (Bo at W & Bo at B)": the central connective is OR. (How can you tell? It's a semantic consideration. Ask yourself under what conditions it would be true, and then which central connective fits them best. That may sound vague and unhelpful, but there are rarely many choices and the differences between them are usually clear. When they are not the English sentence is usually vague or ambiguous.)

These are different. Suppose that Mo skis at Whistler and Bo skis at Vail. Then it is true that both of them ski at one of these places, but not true that either of them skis at both. This is worth getting clear in your mind — think about it for a moment— because while it is not vital at this stage it can prepare an idea that will be important in part III.

These idioms need a warning, though. I said that "Both Bo and Mo ski at either Whistler or Vail" means "Bo skis at either Whistler or Vail and Mo skis at either Whistler or Vail." And most often this is what these words will mean. But English is a subtle and unpredictable beast, and they could be used to express other things as well. On occasion they could mean "Bo skis at Whistler and Bo skis at Vail and Mo skis at Whistler and Mo skis at Vail", even though it might make more sense to express this with "Both Bo and Mo ski at both Whistler and Vail". We can even use "Both Bo and Mo ski at either Whistler or Vail" to mean "Either Bo skis at Whistler and Mo skis at Whistler or Bo skis at Vail and Mo skis at Vail." AND in the scope of OR rather than OR in the scope of AND. Imagine a conversation in which someone says “Sally and Sam both ski at the same one of those places. I wonder who else is like that?” And someone else replies “Bo and Mo both ski at Whistler or Vail”. Then we understand it as saying “Bo and Mo at Whistler or Bo and Mo at Vail”. (Rather than Bo at Whistler or Vail and Mo at Whistler or Vail.)

This can seem confusing just because one tries to let the English stand on its own. That’s not the best approach, though. Think directly in terms of the logic. You think semantically — what models would make the sentence true? — and when you meet the English sentences you take them as crude suggestions which have to be explained in clearer and more explicit terms. Then everything will fall into place.

>> can you think of variants on the “both/either” and “either/both” idioms that resist these ambiguities?

words used in this chapter that it would be a good idea to understand:

and-elimination, assumption, contraposition, counter model, deduction, elimination of alternatives, logical consequence, modus ponens, modus tollens, or-introduction, premise, semantic, syntactic, valid argument

exercises for chapter six

1) a) Make a model, in the form of a table, to show that "Bo is happy" is not a logical consequence of the premise set {Mo is rich or Mo is lying , If Mo is rich then Bo is happy}.

b) Make a truth table showing the same thing, and say how it is related to the model.

2) The arguments in (I) are all invalid. The "conclusions" are not logical consequences of the premises. The situations in (II) describe countermodels to them: they show how the premises can be true while the conclusions are false.

a) Which situations are countermodels to which invalid arguments?

(I)

(a) <u>Mo is a student</u>
Mo is a student and a musician

(b) <u>Mo is a student or a musician</u>
Mo is a student and a musician

(c) <u>Mo is a student or a musician</u>
Mo is a student

d) <u>If Mo is a student then Mo is a musician</u>
If Mo is a musician then Mo is a student

e) <u>If Mo is a student then Mo is a musician</u>
Mo is a musician

(II)

(i) Mo is a non-student non-musician
(ii) Mo is a non-student musician
(iii) Mo is a student non-musician.

b) symbolize the atomic sentences in the arguments above with letters and construct a truth table showing the combinations of truth and falsity that make the premises through and the conclusions false..

3) a) Given the table below, fill in the missing cells.

	plays**C**hess	plays**V**iolin	loves**P**hilosophy	**H**uman	**C v V**	**P ⊃ H**
lindsey	NO		YES		NO	YES
paris		NO	YES	YES	YES	YES
britney	NO	YES	YES			YES
avril	NO	NO	YES	NO		

b) Why is this a question about logical consequence?

4) Which of the arguments below are instances of modus ponens, which are instances of modus tollens, and which are instances of contraposition?

if Mo came the party Mary left
if Mary did not leave the party, Mo did not come

Mo is going to the party
If Mo goes the party, there will be a fight
There will be a fight

If Mo was at the party, there was a fight
There was no fight
Mo was not at the party

If no one comes to the party, Liu will feel awful
Liu will not feel awful
Someone will come to the party

if no one came to the party, Liu felt either awful or relieved .
If Liu felt neither awful nor relieved then someone came to the party

If any of Mary's friends come to the party, Mo will start a fight
Some of Mary's friends will come to the party
Mo will start a fight

5) Two of the sentences a) – d) below link the assumptions 1, 2, 3 to the conclusion 6.

Put them in the right order, by writing their letters in the appropriate places. (Warning:

two of them are distracters, that should be left out.)

1 if the Oilers lost Murray got drunk
2 if the Oilers did not lose Murray proposed to Sarah
<u>3 Murray did not propose to Sarah</u>
4
5
6 Murray got drunk

a) if Murray did not get drunk then the Oilers did not lose
b) If Murray did not propose to Sarah then Murray got drunk
c) if Murray did not get drunk then Murray proposed to Sarah
d) If Murray proposed to Sarah then Murray did not get drunk

6) Which of these are true? (They have not all been explicitly discussed in the text. Some may take a bit of thought.)

a) A sentence is a logical consequence of a set of sentences if it is true in all models that make all sentences in the set true.
b) A sentence is a logical consequence of a set of sentences if it is true in all models that make some sentences in the set true.
c) A sentence is a logical consequence of a set of sentences if it is true in some models that make all sentences in the set true.
d) The conclusion of a logically valid argument is true.
e) If the premises of a logically valid argument are true, the conclusion is true.
f) If an argument is not logically valid its conclusion is false.
g) One query Q1 is broader than another Q2 when any answer to Q1 is an answer to Q2.
h) One query Q1 is broader than another Q2 when any answer to Q2 is an answer to Q1
i) the negation of a conjunction is a conjunction of negations
j) the negation of a conjunction is a disjunction of negations

B - more

7) a) Fill in the empty cells in the model below to make the sentences in the A list true and the sentences in the B list false.

	wears a **H**at	wears a **D**ress	wears **P**ants
albert			
bertie			
charlotte			

A: true
Find x: Hx & Px has the answer {**c**}
the people wearing a hat and a dress are Charlotte and Albert
either Albert wears a hat or Bertie wears a dress
if Bertie wears a dress then Charlotte does not wear a hat

B: false
if Bertie wears a hat then Albert wears pants (treat this as **Hb ⊃ Pa**)
Either Charlotte does not wear a dress or Bertie wears pants but Albert does not.
(~**Dc v (Pb & ~Pa)**) remember: this is to come out False]

b) Why is this also a question about logical consequence?

8) Why not? Explain what is wrong with taking the conclusions of these arguments as logical consequences of the premises. (I have put in?'s to emphasize that the conclusion does not really follow logically.)

If she took the four pm ferry she arrived in time. She arrived in time. Therefore she took the four pm ferry?

He has a fever, purple spots, and sore joints. If he has awfularia he will have fever, purple spots, and sore joints. Therefore he has awfularia?

If the theory of evolution is true then when the environment changes animals will eventually adapt to those changes. When the environment changes animals do eventually adapt. Therefore the theory of evolution is true?

9) Which of these candidate conclusions are logical consequences of these premises?
Premises:
if there is a god, sinners will be punished

if there is a god, there will be prophets
if there is no god, all things are in space and time
sinners will be punished
there will be prophets
all things are in space and time
if there is only a devil, sinners and non-sinners will be punished
if there are delusions, there will be prophets

<u>candidate conclusions</u>:
there is a god
there is no god
there is only a devil
if there is something not in space and time, there is a god

10) Which of these candidate conclusions are logical consequences of these premises?

<u>premises</u>:
if Joe has driven all night he will be tired tomorrow
if Joe is tired tomorrow he will have an accident
Joe will be tired tomorrow
If Joe has been drinking all night he will be tired tomorrow
If Joe has been drinking all night he will have an accident

<u>candidate conclusions</u>:
Joe has driven all night
Joe is tired tomorrow
Joe will have an accident
Joe has been drinking all night
Joe has neither driven all night nor been drinking all night

11) What additional premises could be added to 9 to make the conclusion “there is a god” a logical consequence?

12) For which of the following arguments is the conclusion a logical consequence of the premises?

If you drink this poison you will die
If you don't drink this poison you will not die

If you take this medicine you will be saved
If you don't take this medicine you will not be saved

If we leave Afghanistan the Taliban will win
If we don't leave Afghanistan the Taliban will not win

If your muscles don't ache then you don't have the flu
If you have the flu then you will have a fever
If your muscles don't ache then you don't have a fever

If he took the medicine then he survived
He did not survive
He did not take the medicine

If he takes this medicine then if he stays in bed he will recover
He will stay in bed
He will not recover
He will not take the medicine

13) Here are three **in**valid arguments. Below are situations that may be counter-examples to them. Which are counterexamples to which arguments? Remember: a counterexample is a situation that makes the premise(s) true and the conclusion false

i) **If** you do **not** have a ticket **then** you will have to pay
If you have to pay **then** you do **not** have a ticket

ii) **If** you do **not** have a ticket **then** you will have to pay
You will **not** have to pay
You do **not** have a ticket

iii) **If** you have a ticket **then** you will **not** have to pay
If you **not** have a ticket **then** you will have to pay

Situations
a) Everyone has to pay. b) No one has to pay.
c) Two kinds of tickets for this special event: A – special ticket, enough to get in, B – regular ticket, needs a surcharge. You have B. Those with neither pay.
d) Two kinds of tickets: A and B as above. You have A. Those with neither pay.

14) Fill in the gaps in the argument below, so that it is logically valid and uses only the rules modus ponens, and modus tollens .

1	If it **R**ained the **D**am broke	
2	If the **S**luice was opened the **D**am did **not** break	
3	if there was a **H**urricane it **R**ained	
4	the **S**luice was opened	
5		[4,2 modus ponens]
6		[5,1,modus tollens]
7	there was no Hurricane	[6,3, modus tollens]

15) Which of the following are True, and which False?

a) the conclusion of a valid argument is always true
b) the conclusion of a valid argument is always false
c) the conclusion of an invalid argument is always false
d) the conclusion of an invalid argument is sometimes true
d) the conclusion of a valid argument with true premises is always true
e) the conclusion of a valid argument with false premises is sometimes true

16) Write S for 'Mo is a student', M for 'Mo is a musician', and A for '2+2=8'. Assume that S is true and M and A are false. Give an example of

a) a valid argument with true premises and a true conclusion
b) a valid argument with false premises and a false conclusion
c) a valid argument with false premises and a true conclusion

17) Facts:

- either Albert or Bertie or Charlotte drives a Toyota.

- either Albert drives a Porsche or Albert drives a Toyota or Albert drives a bicycle.

- either Albert or Bertie or Charlotte drives a Porsche and a Toyota and a bicycle

(**(Pa & Ta & Ba) v (Pb & Tb & Bb) v (Pc & Tc & Bc)**)

- either Bertie does not drive a Porsche or Bertie does not drive a Toyota or Bertie does not drive a bicycle

- either Albert or Bertie or Charlotte drives neither a Porsche nor a Toyota nor a bicycle.

((~Pa & ~Ta & ~Ba) v (~Pb & ~Tb & ~Bb) v (~Pc & ~Tc & ~Bc))

- Albert does not drive a Porsche and Charlotte does drive a bicycle.

- Only one person drives a Toyota

Who drives a Porsche, who drives a Toyota, and who drives a bicycle? That is, fill in the empty cells in the model below to fit the facts.

	drives a **P**orsche	drives a **T**oyota	drives a **B**icycle
albert			
bertie			
charlotte			

Hint: Which individual(s) drive all the vehicles? Which one(s) drive none of them?

Remark: this is a traditional logic puzzle, but one where, unusually, it can be tackled by looking directly for logical consequences of the given facts.

18) Given the stated facts of question 17**)** which of the following are true?

Bertie drives a Porsche and Charlotte does not drive a bicycle.
Albert drives a Toyota and Albert drives a bicycle.
Albert drives a Toyota or Albert drives a bicycle.

19) Below are twelve arguments. Which of them is a

(a) valid argument with true premise(s) and true conclusion
(b) valid argument with true premise(s) and false conclusion
(c) valid argument with false premise(s) and true conclusion
(d) valid argument with false premise(s) and false conclusion
(e) invalid argument with true premise(s) and true conclusion
(f) invalid argument with true premise(s) and false conclusion
(g) invalid argument with false premise(s) and false conclusion
(h) invalid argument with false premise(s) and true conclusion

(note: not all of these combinations may be possible, so some may apply to none of the arguments) p =Paris is in France q = Paris is in China t = Toronto is in Canada (the real cities and countries)

s = Stephen Hawking is a man h = Stephen Hakwing is human

(Suggestion: first think "is it valid?" then consider the truth values of the premise(s) and conclusion. By "false premise(s)" I mean that one or more is false. True and false in the real world.)

s ⊃ h **h⊃ s**	**p** **p v q**	**p v q** **p**	**p& q** **p**	**p & q** **q**	**p & t** **t**
h ⊃ s **s⊃ h**	**pv q** **q**	**t v ~q** **t**	**t v ~q** **~q**	**p, ~p v q** **q**	**~q, q v t** **t**

C – harder

20) Explain why modus tollens gives arguments that are deductively valid.

21) Explain why contraposition gives arguments that are deductively valid.

22) Describe a way of using disjunctive normal forms to show that one formula is a logical consequence of another. (How could you extend it to showing that a formula is a consequence of a set of formulas?)

23) There are infinitely many models. So how can we ever know that a conclusion is true in *all* models that make all of a set of premises true?

24) Give examples of situations where reasoning by modus ponens, and reasoning from A & B to A, are unhelpful ways to think.

25) Consider the assumption that there is a single enormous model such that a sentence of English is true if and only if some version of it in the vocabulary of logic is true in that model.

(a) Why should we take a long hard breath before making this assumption?

(b) Explain why if we accept the assumption then all the facts about the relation between logical consequence and truth are verified, in particular consequences of true premises are true, and false premises can have both true and false consequences.

chapter seven: Boolean derivations

7:1 (of 5) what is a derivation?

A derivation is a mechanical way of extracting — deriving — from a proposition its logical content, to determine various of its logical properties such as which other propositions are its consequences, whether it is consistent, whether it is a tautology, and so on. (See the next chapter, 8, for explanations of consistency, tautology, and contradiction.) There will not always be a mechanical procedure for determining any of these. But in the very special case of Boolean propositions where we are determining these properties in terms of truth assignments, they exist. This chapter explains one way of making such derivations, which is closely related to the search trees of chapter four. Traditionally, derivations have been explained as a kind of deduction, a mechanical way of deriving a proposition's consequences, and other logical properties of propositions have been discussed in terms of this. (Tautology and contradiction are discussed in the previous chapter, Chapter 6, and consistency is discussed in the next chapter, Chapter 8.) We will respect this tradition, to some extent, but we will also emphasize that there are other things that a derivation can tell us about a proposition[18].

We have a sentence of propositional logic, in short a proposition, and we are searching for truth assignments, lines of its truth table, which make it true. The search is will take the form of search trees, familiar from the past two chapters. It is worth being explicit, though, about the way we must carry out these searches if our object is to discover

18 Don't call them derivatives! That is from calculus, has no relation, and will annoy your prof.

logical consequences. So in this chapter the rules will be stated in clear and definite terms, important aspects will be noted, and the connection with logical consequence will be central. Suppose the proposition is of the form A & B. (Both of the conjuncts may be complex, but the central connective is **&**. For example **~(p & q) & (q ⊃ r**) is of the form A & B.) Then we know that all truth assignments that make it true make A true, and also that all truth assignments that make it true make B true. So both A and B are logical consequences of A & B. Searching for truth assignments that make A true will always get at least as many as searching for assignments that make A & B true. The same for B.

This motivates the first rule of making a derivation:

> **[& rule (conjunction)]** When you have a sentence A & B, you can write either A or B beneath it. When the derivation begins with A & B we put a horizontal line beneath it and then A, B.

```
 A & B
 -----
   A
   B
```

Note that I said that you *may* write A or write B beneath. Writing either, both, or none are all allowed. We will later see situations where this *may* changes to *must*. The options you have in making a derivation can be confusing, and even dismaying. Usually there are several derivations you can make for a given purpose, and it is not important which one you make. This is a topic to keep in mind in what follows.

For the second case suppose the sentence is A v B .Then the truth assignments that make it true break into two classes. There are those that make A true and those that make B true. Neither of these has to include all the assignments that make A v B true, but searching in parallel for A assignments and B assignments will get them all.

With a derivation using the & rule there is no new branch: the two conjuncts are listed on a continuation of the branch that has their conjunction. Branches of derivations are intended to represent general classes of truth assignments making the propositions at the base of the tree true. When there is only one branch the propositions on it are logical consequences of those above it, as in this case conjuncts are logical consequences of their conjunction.

We also have a rule for disjunction, which again is familiar from search trees.

> **[v rule (disjunction)]** When you have a sentence A v B , you may write both A and B beneath it in the form of two separate branches. When the derivation begins with A v B we put a horizontal line beneath it and then the A, B branches.

```
  A v B
 ______
 /    \
A      B
```

Since there is more than one branch, the intention is not to represent propositions on either branch as logical consequences of those above them. And clearly neither A nor B is a logical consequence of A v B. Each branch is meant to represent one class of truth assignments for the propositions at the base of the tree, and taken altogether they are meant to represent all those propositions truth assignments. We will return to what more precisely this means.

For the third case suppose the formula is A ⊃ B. Then again the truth assignments that make it true break into two classes. There are the assignments that make A false, and thus ~A true. And there are the assignments that make B true. Between them they include all the assignments that make A ⊃ B true though neither class gets them all by itself. So we are again dealing with a branching rule. (And a parallel search: if you want to catch all the truth assignments for the conditional you will explore both branches.)

> **[⊃ rule (conditional)]** When you have a sentence A ⊃ B, you may write both ~A and B beneath it in the form of two separate branches. When the derivation begins with A ⊃ B we put a horizontal line beneath it and then the ~A, B branches.

```
   A ⊃ B
   ‾‾‾‾‾
   /   \
 ~A     B
```

Now we have a rule for each of the Boolean connectives we are using except for ~ . Instead of a single rule for ~ we have three, depending on the proposition that is negated. The reason that we need three rooms instead of one is that there is not really anything much that all the truth assignments that make a proposition force, and thus its negation true, have in common. Instead, the patterns are different for each Boolean connective. For conjunction and disjunction amount to de Morgan's law, as the rules **~&** and **~v** suggest. For the conditional they amount to the important feature of its truth table, that conditional is true except when the antecedent is true and the conclusion is false. (There is room for a fourth rule saying what happens when we apply negation to negation. But it turns out that this is not needed given the other three.). The three rules are below. I will just state them in terms of the diagrams that you should understand by now.

[~& rule]

```
 ~(A & B)
 ‾‾‾‾‾‾‾‾
   /   \
 ~A     ~B
```

[~v rule] ~(A v B)
~A
~B

[~⊃ rule] ~(A ⊃ B
A
~B

>> give the motivation for each of these three, in terms of searching for truth assignments

7:2 (of 5) examples

These rules break sentences of propositional logic down into their parts. This will give us a manageable way of answering a number of important *semantic* questions about them. That is, questions about the conditions under which they are true and false. These questions could also be answered by using truth tables, but truth tables get very unwieldy as the sentences become larger, while derivations can handle quite large sentences. Derivations appeal to *syntactic* properties of sentences, that is, features of their grammatical structure rather than what they mean. In principle, one could answer syntactic questions about a sentence without knowing that **&** means AND, **v** means OR, and so on, though for normal human beings this would not be a natural way of proceeding.

>> "semantic" originally meant "about signs". what is the connection with the present use of the word, which is standard in logic, philosophy and linguistics?

>> can we answer grammatical questions about English sentences without knowing what they mean? consider "Horton hatches a who."

To see how this works we need some practice. Here are some examples, with a couple of further ideas. First a short and simple one, followed by some comments.

(a)

1	**~(p v q)**	
2	**~p & ~q**	(1: ~v)
3	**~p**	(2: &)

First comment: I have put line numbers on the left and explanations on the right of which rules are used. Derivations do not need to have these, but sometimes they make it clearer what is going on. (Like a REM note in a program.)

Second: The rules apply because **~(p v q)** is an instance of the general pattern ~(A v B) and **~p & ~q** is an instance of the general pattern A & B. A sentence can be an instance of several patterns. For example, **~(p v q)** is also an instance of ~A.

Third: We could have made other derivations starting with **~(p v q)**. For example we could have had **~q** at line 3. One reason for doing the derivation this way would be if you had been asked to show that **~p** is a logical consequence of **~(p v q)**. The derivation does show this because the only truth assignments that make **~(p v q)** true are those that make the lines that follow it true. This is because the derivation does not branch.

Now another derivation that is very similar. We are asked to show that

~((pvq) v r) ⊨ **~p** . One answer is this

(b)

1	**~((p v q) v r)**	
2	**~(p v q) & ~r**	(1: ~v)
3	**~(p v q)**	(2: &)
4	**~p & ~q**	(3: ~v)
5	**~p**	(4: &)

This time we have applied the rule **~v** twice. The first time ~(A v B) is **~((p v q) v r** —so A is **(p v q)** and B is **r** — and the second time ~(A v B) is **~p v ~q** — so A is **p** and B is **q**.

Time for a branching derivation. Suppose we are asked to show that **p, p ⊃ q** ⊨ **q** . We can respond with this:

```
(c) 1          p
    2          p ⊃ q
    3           p
              /   \
    4       ~p   5  q             (2: ⊃)
             X
```

There are three new features here: there is more than one premise, the derivation branches and one has an **X** . (I have also repeated the **p** on line 3. This is not really necessary.) The branching at line 3 follows the ⊃ rule. Because of the branch, we should hesitate to consider the propositions on each branch as consequences of the premises. (Together the branches will catch the truth assignments for the premises they are derived from, but each branch need not catch all of them.) BUT on the left branch we have both a **p** and a **~p**. As with search trees, a branch that has any formula and its negation — both on the same branch, so here on the left branch but not on the right one — is said to *close*, and we mark it with an **X**. No truth assignment can make the formulas on such a branch true: it would have to make both the formula and its negation true, which is impossible. So the only truth assignments that make 1 and 2 true are ones that make the formulas on the right branch true. And that shows that **p, p ⊃ q** ⊨ **q**.

7:3 (of 5) derivations and logical consequence

We write "**~p** is a logical consequence of **~(p v q)**" as "**~(p v q)** ⊨ **~p** ". (⊨ is often called a *turnstile*. The point of putting ⊨ in colour is so you don't think it is part of a sentence of propositional logic. It is a relation between such sentences.) A proposition P is a logical consequence of a set of assumptions of propositional logic, $\{A_1,...,A_n\}$, $\{A_1,...,A_n\}$ ⊨ P , when there is a derivation with $A_1,\ldots,A_n$ at its base where A occurs on every branch that does not close (has no atomic proposition and its negation, so no X). A is a logical consequence of $A_1,\ldots,A_n$ under these conditions because every truth assignment that makes all these assumptions true is described by one branch or another of the derivation. So if A is on every branch it is made true by every such truth assignment, so it is a logical consequence of the assumptions. I have not presented a formal mathematical proof of this fact, but you can persuade yourself that is true by noticing that for each of our rules every truth assignment that makes all the assumptions true makes one branch or another of the resulting tree true, so sentences that are found on every branch are made true by all those assignments. And when the derivation has broken the molecular proposition down into its smallest atomic components, the branches give truth assignments to those atomic propositions and thus to the molecular proposition. (The derivation may reveal the conclusion on every branch before the premises are fully broken down into their parts. But then we could continue the branches until they were fully broken down. This is the idea behind C-derivations, discussed in section 5 of this chapter.) In systems of logic beyond propositional logic things are not this simple.

There are usually many derivations that will show that one sentence is a logical consequence of some premises.othe These questions often have many right answers. When you make a derivation you should first be clear about what it is meant to show. At this stage we are showing that a formula is a logical consequence of given others, so we need to be clear about what consequence we are aiming at. Later we will show other facts with derivations. This is an inescapable fact about argument and proof. There are many ways of getting from premises to any of their consequences. Every mathematical theorem has in principle infinitely many proofs, though in fact we get bored with proving the same result in yet another way.[19] And when asked to give an argument or proof from given premises to a given conclusion it is often not at all obvious what to do, as every student of mathematics or philosophy, and every lawyer whose client has a very weak case, knows. In this course we are more interested in understanding what logical consequence is, and how we show it, than we are in making un-obvious arguments. So it will always be possible to make the derivation that is asked for by blundering ahead blindly and mechanically, though the result may be longer than necessary. Later in the chapter I explain how to do this. But it is important to see that in the rest of life this is usually not so.

> >> what is the difference between a valid argument and a proof? "proof" is not such a very precise term, so think of several things we often mean when we say we have proved something.

There is an important principle here: <u>to show that a conclusion is a logical consequence of some premises, we can make a derivation from those premises where the consequence is found on all branches of the derivation except those that close</u>.

It is worth repeating here that not all the formulas that appear on a branch are consequences of the premises. But the derivation as a whole can show that a formula is a consequence, when it is found on all the branches that do not close. This is evident from the v rule. Neither **p** nor **q** is a consequence of **p v q**, but **p** is a consequence of **p v q**, **~q** since the **q** branch closes. The best way to think of a derivation is as going through all the possibilities given the premises. (Often when we think we are not trying to

19 Elisha Loomis' *The Pythagorean Proposition,* published by the National Council of Teachers of Mathematics contains 370 proofs of the Pythagorean theorem on right angled triangles.

demonstrate something we think is true, but are exploring what we do not yet know to be false.)

The consequence can appear on branches that do close, as long as it is found on all un-closed branches. For example we could show that **p v (p & ~p)** ⊨ **p** with this derivation

```
(d) 1    p v (p & ~p)
          /       \
2        p       5  p & ~p       [1: v]
3                6    p
4                7   ~p
                      X
```

In this derivation **p** is found on the only branch that does not close, and is also found on a branch that does close. The derivation still shows that **p** is a consequence of the premise. And this makes sense as it is like "either it will rain (as the reliable forecast says) or it will simultaneously rain and not rain (as crazy Joe says). If the first alternative, then it will rain. And we can ignore the second alternative as it contradicts itself."

>> we could also have stopped the derivation at line 6. it would still have shown that **p** follows from the premise. and it would still have made sense in a familiar way. modify the crazy Joe argument so that it fits this shorter derivation.

Suppose we are asked to show that **f ⊃ t, ~f ⊃ t** ⊨ **t** . We make this derivation

```
(e)         1          f ⊃ t
            2         ~f ⊃ t
                      /     \
            3       ~f      4  t          (1: ⊃)
            5       /  \
            6  ~~f      t                 (2: ⊃)
                X
```

>> this derivation corresponds to reasoning we use very frequently. describe it in everyday language.

This was a simple derivation, but it raises an important issue. The ⊃ rule was used to branch **~f ⊃ t** into a **~~f** branch and a **t** branch, the first of which was inconsistent with a

~f on the same branch that we had already derived. That left us with only branches with **t**, so we can conclude that **t** is a logical consequence of our premises: it is true any way we try to make the premises true. But, students often worry, why didn't we also apply the ⊃ rule to premise 2 on the branch with the **t** as well as the branch with the **~f**? Part of the answer is that we could have, but that it would have made no difference. We would have got the tree below.

(f)

```
1                 f ⊃ t
2                ~f ⊃ t
               /        \
3            ~f          t              (1: ⊃ )
            /  \        /  \
4        ~~f    t    ~~f    t           (2: ⊃)
          X
```

Again all the branches without **X**s have **t**. (The branches on the right had **t** even before their last branching.) So this tree shows exactly the same logical relationships as the simpler one. They are both acceptable derivations, though not every derivation from these two premises will show **t** to be a consequence of them. (Keep reading, if you want a mechanical way of finding the ones that do.) But you may still worry, how do we know that if we keep applying the rules to an unclosed branch (no **X** — yet) we will not find a contradiction? Isn't there a natural stopping point?

The answer is that we can often go on deriving manically, if we want, making the tree bigger and bigger. (This becomes even more of a possibility when we introduce the rule **EM** below. But there is a natural stopping point, as we will see.) And there is always the possibility that a hidden contradiction will turn up. But if it does it does, and this does not deny the fact that **t** is on all the unclosed branches. All models for the premises still make **t** true. One way of seeing this is to consider explicitly contradictory premises. First a one-step derivation

(g) 1 **p & ~p**

p (&: 1)

In this derivation the only un-closed branch has **p** on it. And that is reasonable: if **p & ~p** is true then **p** is true, because whenever A & B is true, A is true. (But **p & ~p** isn't true, ever.) We can continue the derivation with another line.

```
1    p & ~p
       p            (&: 1)
      ~p            (&: 1)
       X
```

Now the only branch has closed. So the two-step derivation does not show that **p & ~p** ⊨ **p**. But the one-step derivation does.

>> "whenever A & B is true, A is true. (But **p & ~p** isn't true, ever.)" isn't this a bit puzzling?

Two more.

(h) **~(h v s), m ⊃ s** ⊨ **~m**

```
1      ~(h v s)
2        m ⊃ s
3      ~h & ~s              (1: ~v)
          ~s                (3: &)
         /  \
       ~m    s              (2: ⊃)
             X
```

(i) **(p v q) ⊃ (r v s) , p, ~s ⊨ r**

```
1                  (p v q) ⊃ (r v s)
2                          p
3          ________________~s________________
              /                            \
4          ~(pvq)        5                  r v s          (1: ⊃ )
6          ~p & ~q       (4:~v)   7       /     \
8          ~p            (6: &)           r       s        (5: v)
            X                                     X
```

Making short tidy derivations is a knack that comes with practice. But making formally correct derivations is a completely mechanical business, once you know the rules. The exercises following this section and at the end of the chapter are meant to give you more practice and to make the rules automatic.

Making derivations is a quasi-mathematical activity in this way: there is a purpose and meaning to the activity but a lot of it is best done automatically though with intelligent oversight. What you do is think generally about the reasons why the conclusion is plausibly taken as a consequence of the premises, and then you let this guide you in making a formerly correct derivation, which you do just by reflectively following the rules but choosing them so they fit your general idea. Then after you have done this you check with a more conscious kind of care that you have really followed the rules. If you have, then the derivation backs up your earlier intuition that the consequence does follow and that you have understood why it does. You need both a feeling for why the rules are as they are and a facility with applying them automatically. Examples (b) (d), (h), (i) are good for imagining how you would mobilize the skills. Many of the exercises at the end of the chapter are meant to develop a little more awareness of this process.

AS YOU GO EXERCISES 1

You could continue reading the next section. But I strongly urge you to wait until you have worked through this short exercise section. It will convince you that making derivations to show simple cases of logical consequence is not hard, and that the strategic thinking involved is familiar. There are more and more varied exercises at the end of the chapter.

1) Begin with one that we have seen before: a derivation to show that

f ⊃ t, ~f ⊃ t |= **t** . Since **t** is the conclusion we are aiming at, we want to make a tree starting from the two premises where all branches close except ones with **t**. In the main text this was done by applying the ⊃ rule to the first premise first, but now we are going to do it by first applying the ⊃ rule to the second premise. So we begin

1 **f ⊃ t**
2 **~f ⊃ t**

and we get our next line from line 2. First, what is its central connective? (Yes, we have to be clear about these.) 2 is a conditional whose antecedent is a negation. So we need the ⊃ rule, which means branching. So we get this

1 **f ⊃ t**
2 **~f ⊃ t**
**/ **
3 __ 4 __ (2: ⊃) ??

What goes on lines 3 and 4 in accordance with the ⊃ rule? Write it in.

Remember that we want unclosed branches with **t**: branches with **t** where there is no **~t**, or any other A and ~A on the same branch. Do we have any now? Yes. And so leave them as they are and apply a rule using premise 1 to the remaining branch. (What is the central connective of 1? So that's the rule we need.) So we get this

```
1        f ⊃ t
2       ~f ⊃ t________
        /           \
3       __       4   ____   (2: ⊃)  you've filled in these two blanks
       /  \
5  __   ___          now fill in these two and add an X to any branch that closes.
```

Compare the derivation you have made to the one in the answers following the exercises at the end of the chapter.

2) Now one needing a bit of strategy. The problem is to show that

(p & q) v (r & ~q) , q ⊨ **p** So you want to make a tree starting with these premises and with **p** on one or more branches, which you hope will not be closed. How are you going to get to this **p**? Which of the premises has a chance of providing it? [Pause and think before going on.]

Now begin the derivation with that premise. It's **(p & q) v (r & ~q)** , because the other does not even contain **p.** Since it is a disjunction there is going to be a branching, and one of the branches should close and the other should have **p**. Which are they? You will get this

```
1  (p & q) v (r & ~q)
2_ q________________
        /            \
3 _____        4 _____      [v: 1]      - the move that gives you a chance
                                          of p while the other branch closes.
5  p           6 ______                 - something you can get from 4
                                          that negates the unused premise.
                   X
```

Compare the derivation you have made to the one in the answers following the exercises at the end of the chapter.

One more, but it asks for two derivations. The premises **(p v q) &(~p ⊃ s) , ~p** have both **q** and **s** as consequences. Make derivations showing

(a) **(p v q) &(~p ⊃ s), ~p** ⊨ **q** and

(b) **(p v q) &(~p ⊃ s), ~p** ⊨ **s**

The complication is that the strategy is different for (a) and (b).

(a) **(p v q) & (~p ⊃ s)**
~p ______________ - we want a branch with **q,** and all others to close. which premise offers the best chance of this? Central connective? use the rule for that.

____________ - the result is this. Central connective? Use the rule for that.

**/ **

____ **q**

X

(b)

(p v q) & (~p ⊃ s)
~p ____________ - we want a branch with **s,** and all others to close. Which premise offers the best chance of this? Central connective? Use the rule for that.

__________ The result is this. Central connective? Use the rule for that.

**/ **

_____ **s** What rule do we use now to get **s** on the right and an X on the left?

X

4) Now do these two without any hints.

a) **~(p ⊃ q) , q v r** ⊨ **r**

b) **p ⊃ (q ⊃ r) , p & q** ⊨ **r**

7:4 (of 5) the rule of excluded middle

We need one more rule. (There is a way of avoiding it, though, for showing ⊨ **,** as explained in section 4.) The rule is "Excluded Middle"

```
      A                          (EM).
   ______
    /   \
   B     ~B
```

At first this rule may seem rather puzzling. What has B got to do with A? But note that it is a branching rule. It does not say that when you have A you have B, but that the models for A can be divided into two kinds, those which satisfy B, and those which do not. (This is a version of what is called "the law of excluded middle"— there's nothing in the middle between truth and falsehood[20].)

> >> what is the connection between the rule EM and the idea that there is nothing between truth and falsehood?

We need EM because the rules stated so far only allow us to show that conclusions are logical consequences of premises when they are simpler than them. The derivation works by taking the premises into pieces. But often a conclusion is got by putting pieces of the premises together again. For example we might want to show that **p, q** ⊨ **p & q** , or that **p** ⊨ **p v q** . We could make rules that stick pieces of the premises together, but instead we can do everything with EM. Consider some examples.

p, q ⊨ p & q

```
1                 p
2                 q
      _______________________
            /                \
3     ~(p & q)        4      p & q          (EM)
5     ~p v ~q  (3:~&)
      /          \
7   ~p        9  ~q          (5: v)
11   p  (1)  13   q          (2)
     X            X
```

At lines 11 and 13 I have repeated lines 1 and 2. This is not really necessary, but makes it easier to see why the branches close. To get the feel of this derivation think of it as saying " **p** is true, and so is **q**. Either **p & q** is true or it isn't. If we suppose it is not then

20 There was a tax collecting procedure in 15th century England, involving questions from which there was no escape, called "Morton's Fork". Personally, I think of the rule of excluded middle as Morton's Fork.

we get an impossible situation. So it must be true."

The rule of EM gets around the problem about complex conclusions because we do not have to find a way of building up to more complex propositions using rules whose nature is to break down to simpler ones. Instead, we can break down the negation of the conclusion, and show that its simpler parts contradict the premises. This will work because the truth assignments break into two classes, those that make the conclusion true and those that make its negation true: if all of the ones that make the negation true are ruled out by the premises, then all the remaining truth assignments make the conclusion true. Thus it is a logical consequence of the premises.

p |= pvq

1 **p**
/ \
2 **pvq** | 3 **~(pvq)** (EM)
4 **~p&~q** (3: ~v)
5 **~p** (4: &)
6 **p** (1)
X

p |= ~~p

1 **p**
/ \
2 **~p** | 4 **~~p** (EM)
3 **p** (1)
X

Note how in all of these we make a branching using EM and then show how one of these branches closes, leaving only the other one. Thinking which branches to create can involve some careful plotting of how you intend to get to your intended result. Note also how in the first of these two we have a branching within a branching. So both of the sub-branches have to close for that whole branch to close, leaving the remaining unclosed branch as the conclusion. To repeat, we will see an automatic way of doing this. But logic teachers hope that learning to do it by thinking ahead may make people better at the strategies of real-life thinking. At any rate, it is practice in thinking of a series of steps that leads to an end, like playing chess, writing programs, or playing games like Tetris.

>> why might someone hope that derivations give better practice than these alternatives? might they be wrong?

As a last example let us prove **~p v ~q** ⊨ **~(p & q)** the reverse of the rule **(~&)**. This is another part of de Morgan's laws: the negation of a conjunction is a disjunction of negations, and the negation of a disjunction is a conjunction of negations.

```
          1       ______~p v ~q____________________
                      /                          \
                   ~p                            ~q                (1:v)
                 /      \                      /        \
       p & q       ~(p & q)  (EM) p & q          ~(p&q)     (EM)
         p                          q
         X                          X
```

Note how this derivation involves two branches, both of which have sub-branches, and the conclusion is found on all the non-closed sub-branches. Note too that in it EM is applied after the v rule. It is tidier this way, but we could have applied them in the other order (see exercise **7**.)

Most of the content of this course is easy once you get used to it. This is especially true of the content of this chapter and the next. But it is essential that you do the exercises and check the answers, and discuss any difficulties you have in class.

AS YOU GO EXERCISES 2

Convince yourself that the material in this chapter is not difficult by doing the following derivations. They all need EM, because the conclusion is as complex as the premise.

5) a) **~p v q** ⊨ **p ⊃ q**

Given that we need EM, the easiest (not the only) way to start is as follows

1 **~p v q**
**/ **
2 **p ⊃ q** 3 **~(p ⊃ q)** **(EM)**

(Either the conclusion is true or it isn't. If it isn't, we expect some contradiction, given the premise.)

You do not need to do any work on the left branch. If all branches on the right close, then we have succeeded. Central connective on the right? What rule?

4_____ Write in what you get using this rule. No contradiction yet, but we haven't

/ ** used the premise. It's a disjunction, so the **v rule gives branches.

____ ____ Write in what the v-rule gives from the premise.

Now we need a contradiction on each of these branches. Look at line 4. Use the rule for its central connective, once for each branch.

____ _____ Write in the results of this rule, so that there is an **X** on each branch.

Note: it would have been neater to look ahead at line 4, and put in two lines anticipating the need for contradictions further down, but this requires more strategic sense. I will give both solutions in the answers.

b) **p ⊃ q** ⊨ **~p v q**

The technique is the same as in a). Make an EM branch at the beginning, into

~p v q and **~(~p v q)**, use the only rule that applies to the latter, use the premise to make two branches, and find a contradiction in each.

c) **p** |= **~~p**

This looks simple but there's a trick to it. We want to branch to a formula and its negation, where one is **~~p** and the other contradicts **p**. The naïve way is to do an EM branch with **~~p** and **~~~p** and get a contradiction from the latter. But branching to **p** and **~p** is simpler, shorter, and satisfies the EM rule. Write it out.

7:5 (of 5) other things you can show with a derivation

Suppose that the task is not to show that one formula is a logical consequence of others, but just to make derivations, the bigger the better. But there are two restrictions. You begin the derivations with just one formula, and the next line of a branch must always be a part of that beginning formula. (This sounds like it opens up confusingly many possibilities. But just wait: soon it leads to something mechanical.)

>> how can a derivation use atomic formulas that are not there at the start?

So for example

```
  p & (q ⊃ (r & p))
  -----------------
          p
     (q ⊃ (r & p)
       /       \
     ~q        r & p
```

is good, but

```
  p & (q ⊃ (r & p))
  -----------------
          p
     (q ⊃ (r & p)
       /       \
   (r v p)   ~(r v p)     (EM)
```

is not even part of one, since things start getting bigger again on the last line.

And this one is not what we are trying to make either:

```
         p & (q ⊃ (r & p))
               p
          (q ⊃ (r & p)
           /          \
          s            ~s          (EM)
```

for although the last line contains formulas that are smaller than the one above it, they are not parts of any previous formulas.

If we make derivations along these lines, they will always break the starting formula down into parts, and ultimately into its atomic parts. These can't be broken down any further, so the process will stop. Some atomic formulas may get overlooked though, so now add one more requirement: each branch that does not close must contain every atomic formula found in the starting formula or its negation. So now our derivations

- begin with just one proposition
- end when we have a derivation tree where each branch that does not close contains every atomic formula found in the starting proposition, or its negation

Call derivations that satisfy these conditions C-derivations (C for complete). They have some interesting features. Consider some examples, and what they show about their starting formulas. A very simple example will suggest a short-cut that helps with more complicated ones.

```
            p ⊃ q
          /         \
1       ~p            q
        /  \        /  \
2      q   ~q      p   ~p   (EM)
```

Notice how EM is used on line 2 to make every branch contain one of **p** or **~p** and one of **q** or **~q**. But the vital thing to notice is the four branches: **~p—q** , **~p—~q**,

p—q , and **~p—q** . These represent all the combinations of **p**, **q**, and their negations that make **p ⊃ q** true. (The combination of **~p** and **q** occurs twice: this method will often lead to duplicates like this.) In the previous chapter (ch 6) we discussed disjunctive normal forms, and C-derivations give a simple way of getting them: **p ⊃ q** is equivalent to **(~p & q) v (~p & ~q) v (p & q)** . This is simply the disjunction of the branches of the derivation, conjoining the atomic formulas and their negations on each and leaving out the repeated disjunct. The equivalence makes sense given that the formulas on each branch are true in all truth-assignments that make the formulas above them on the same branch true and that when a branch splits into two all truth assignments that make the formula before the split true also make one of the formulas on the two splits true. (This is just a hint how you might prove the equivalence carefully, a large part of what is known as the completeness of propositional logic, but it should be intuitively plausible. Exercise **13** at the end of the chapter asks for a real proof.)

We can now say how to make C-derivations mindlessly, no strategy needed. These can be unwieldy monsters, though, and while you can do them without looking ahead it can be easy to get lost in the details half way through. (There is a basic trade-off in our thinking here.) Start with the premise proposition, and use the &, V, ⊃, and **~** rules whenever they apply. When you have done this as much as you can, apply EM for any atomic proposition appearing in that premise on any branch on which that atomic proposition does not appear. The result will often be a large ugly mess, but it will have an unclosed branch for every truth assignment that makes the premise true. It will be a truth table in disguise. By the end of the next chapter you should understand how these mechanical derivations can show many things about Boolean propositions and the connections between them. But I recommend trying to make derivations by thinking ahead. It will teach you more about reasoning.

Now for some more interesting examples. I discuss three, which illustrate some important points.

A
```
                 p ⊃ (q ⊃ r)
              /                    \
          ~p                      q ⊃ r
        /     \                 /         \
      q        ~q            ~q             r
     / \      / \          /    \        /     \
    r   ~r   r   ~r       p     ~p      p       ~p
                         / \   / \     / \     / \
                        r  ~r r  ~r   q  ~q   q  ~q
```

This shows that the truth-assignments that make **p ⊃ (q ⊃ r)** true are **~p—q—r**, **~p—q—~r** , **p—~q—r** , **~p—~q—~r** ,**p—~q—r, p—~q—~r**, **p—q—r**, and **p—~q—r** . (I am using the "**—**" as a way of picturing the branch while giving the atomic propositions that the truth assignment makes true.)

>> what truth-assignment is *not* on this list? check that **p ⊃ (q ⊃ r)** is false on it.
>> when would we use an English sentence corresponding to **p ⊃ (q ⊃ r)** ? what words would we be likely actually to use?

B
```
                 p ⊃ (r ⊃ q )
              /                 \
          ~p                      r ⊃ q
        /     \                 /         \
      q        ~q            ~r             q
     / \      / \           /  \          /    \
    r  ~r    r   ~r        p    ~p       p      ~p
                          / \   / \     / \     / \
                         q  ~q q  ~q   r  ~r   r  ~r
```

This shows that the truth-assignments that make **p ⊃ (r ⊃ q)** true are **~p—q—r** ,

~**p—q—~r** , ~**p—~q—r, ~p—~q—~r** , **p—q—~r**, **~p—q—~r, p—q—r,** and **p—q—~r**. If we compare this to A we find some overlap: both **p ⊃ (q ⊃ r)** and **p ⊃ (r ⊃ q)**are true on the assignments **p—q—r** , **p—~q—r** , **p—~q~r** ,~**p—q—r**, **~p—q—~r**, and ~**p—~q—r**. But the assignment **p—q—~r** makes **p ⊃ (r ⊃ q)** is true while **p ⊃ (q ⊃ r)** is false, and **p—r—~q** makes **p ⊃ (q ⊃ r)** true while **p ⊃ (r ⊃ q)** false.

> >> find corresponding English sentences for which **p ⊃ (r ⊃ q)** is true while **p ⊃ (q ⊃ r)** is false, and for which **p ⊃ (q ⊃ r)** is true while **p ⊃ (r ⊃ q)** is false.

Propositions like **p ⊃ (q ⊃ r)** (from A) and **p ⊃ (r ⊃ q)** (from B) are said to be *logically independent*: either can be true while the other is false. Contrast this with the relation between **p ⊃ (q ⊃ r)**, and **(p v q) ⊃ r** . Here is a C-derivation for **(p v q) ⊃ r**:

```
          (p v q) ⊃ r
        /              \
    ~(p v q)            r
    ~p & ~q          /      \
    ~p              p        ~p        (EM)
    ~q            /  \     /  \
    /  \         q  ~q   q  ~q         (EM)
   r    ~r                             (EM)
```

(p v q) ⊃ **r** is true in **~p—~q—r**, **~p—~q—~r**, **p—q—r**, **p—~q—r**, and **~p—q—r**. This list is included in the list for **p ⊃ (q ⊃r)**. That means that if **p ⊃(q ⊃ r)** is true under a truth assignment then **(p v q)** ⊃ **r** is also true. In other words **p ⊃ (q ⊃ r)** ⊨ **(p v q) ⊃ r** . This gives us another way to show that A ⊨ B : we can give C-derivations of A and of B and note that the branches of A are among those of B . In the next chapter we will see other ways of showing ⊨ ; there are many. (If you find it surprising that

p ⊃ (q ⊃ r) ⊨ **(p v q) ⊃ r** it may help to know that **p ⊃ (q ⊃ r)** is equivalent to **(p & q) ⊃r)**. **S**ee also exercise **10** of this chapter.

Now compare **(p v q)** ⊃ **r** to **p & ~r** . Here is a C-derivation for **p & ~r**.

C

<u>p & ~r</u>
p
~r
**/ **
q ~q

So the truth-assignments for **p & ~r** , using the atomic formulas **p**, **q**, **r** and their negations, are **p—q—~r** and **p—~q—~r**. These do not overlap with the truth assignments for **(p v q)** ⊃ **r** , but they do not include all the truth assignments that make **(p v q)** ⊃ **r** false. As a result they cannot both be true, but they can both be false, for example when **p** is false **q** is true and **r** is false. Pairs like this are called contraries (think of lions and camels: nothing can be both a lion and a camel, but many things are neither.) Contraries are different from contradictories, where one is true if and only if the other is false (lions and non-lions).

>> what was the point of the proviso "using the atomic formulas **p**, **q**, **r** and their negations"?

We can show other things using C-derivations. When the branches for a proposition include all the truth-assignments for its atomic propositions and their negations, it is a tautology, true on all lines of its truth table. When all branches close, it is a contradiction. (In everyday English "tautology" means "triviality", but philosophers are interested in propositions that are true in all models because they can safely be assumed in any context.) There are exercises at the end of this chapter on telling contraries from contradictories, recognising tautologies and contradictions, and in general on getting familiar with derivations and what they can show.

words used in this chapter that it would be a good idea to understand:
branch, C-derivation, closed branch, conditional rule (⊃), consistent, contradiction, deduction, conjunction rule (&), derivation, disjunction rule (v), excluded middle(EM), negation rule**s**, tautology, turnstile (⊨).

exercises for chapter seven

Because of the as-you-go exercises in the chapter, there is no division into Core and More. However there is repetition between the exercises. 6 covers the same ground as 5, and if either of 1 or 2 is no problem, the other is unlikely to be. But there is a section of harder exercises at the end.

1) Below are four derivations. Mark each as follows:

- whenever a branch closes with an **X** indicate (e.g. by drawing lines) what two previous lines on that branch contradict one another
- at each line write in the rule that is used to get it and the lines it is got from
- if the derivation shows that a conclusion follows from the premises circle the conclusion.

```
  p ⊃ q
  p v r
  ~r
  ______
  /    \
~p      q
/ \    / \
p  r   p  r
   X      X
```

```
(p v q) ⊃ (r v s)
        p
       ~s
_____________________
      /        \
  ~(pvq)      r v s
 ~p & ~q      /   \
   ~p        r     s
   X               X
```

~(p & (q ⊃ r))
~p ⊃ q
~p v ~(q ⊃r)
**/ **
~p ~(q ⊃ r)
/ \ q & ~ r
~~p q q
X

p ⊃ q
q ⊃ r
p & s
p
s
**/ **
~p q
X **/ **
~q r
X

2) _ Fill in the gaps in the derivations below

A
1 **p v q**
2 **~(r v p)**
3 **~r & ~p** [ ]
[3,. &]
[3, &]
**/ **
p [1, v]
X

B
1 **~ p ⊃ ~q**
2 **q**
/ \
[1, ⊃]
X

C
1 **p v q**
2 **r v s**
3 **~(p & r)**
4 ~ q
5 [3, ~&]
/ \
[1, v]
/ \ X
[5, v]
X / \
[2, v]
X

D
1 **p ⊃ q**
2 **~((p & r) ⊃ q)**
3 **(p&r) & ~q** [2, ~ ⊃]
4 [3, &]
5 [3, &]
[4, &]
[4, &]
/ \
[1, ⊃]
X X

E
1 **p**
/ \
~~p [EM]
X

F
1 **p ⊃ q**
/ \
p ⊃ (q v r) **~(p ⊃ (q v r))** [EM]
2 [~ ⊃]
3 [2, &]
4 **~(q v r)** [2, &]
5 [4, ~v]
[5, &]
/ \
[1, ⊃]
X X

3) Make derivations to show the following. None of them need the rule EM.

(a) **(p v q) v r , ~p, ~q** ⊨ **r**

(b) **p v (q v r) , ~(q v r)** ⊨ **~p**
There are two natural ways of doing this derivation: try to find both.
Give an interpretation of this in terms of an everyday argument that someone might actually make. In fact, do this for all of these.

(c) **p ⊃ q, q ⊃ r, p** ⊨ **r**

(d) **p v q , p ⊃ (s&~s) ⊨ q**
Think of this as a version of "one of these two things is true, but if the second were true, we'd have a contradiction, so .... ".

(e) **p ⊃ q, q ⊃ ~p ⊨ ~p**
This won't seem surprising to you once you've thought of an everyday example.

(f) **p ⊃ q, p ⊃ ~q ⊨ ~p**
Very similar to (e).

5) Derivations that need the rule EM. They go from the simple to the moderately complicated.

a) **~p v q ⊨ p ⊃ q**

b) **p ⊃ q ⊨ p v q**

c) **~(p ⊃ q) ⊨ q ⊃ p**

(this and the next three are properties of ⊃ that can seem surprising, if you are thinking in terms of the English "if".)

d) **p ⊨ ~p ⊃ p**

e) **p ⊨ ~p ⊃ q**

f) **~(p⊃ q) ⊨ q ⊃ p**

g) **p ⊃ (q ⊃ r) ⊨ q ⊃(p ⊃ r)**

h) **p ⊃ (q ⊃ r) ⊨ (p & q) ⊃ r**

i) **(p v q) ⊃ r ⊨ (p ⊃ r) v (q ⊃ r)**

j) **(p v q) & r ⊨ (p & r) v (q & r)**

k) **(p & q) & (p ⊃ (q ⊃ r)) ⊨ r**

6) make derivations to show the following:

p & q, ~(p & r) ⊨ ~r

p v q , p ⊃ r , q ⊃ r ⊨ r

p ⊃ q , q ⊃ r , ~r ⊨ ~p

7) Make a derivation to show **~p v ~q** $\models$ **~(p & q)** in which, unlike the one in the chapter the **v** rule is used before **EM**.

8) Facts: one of three roads, A, B, C leads to the treasure. If you take A you will pass a pit of cobras and giant spiders. If you take B you will go through a forest of wolves and bears. If you take C you will either get to a mansion inhabited by enraged zombies or you will be stopped by a choir of angels singing songs of warning. If you do not meet any spiders then you have been cursed by your mother. If you do not meet any bears then you have been cursed by your father. If you do not meet any angels then you will have been cursed by our true love. You will not find the treasure if you have not been cursed.

Which road leads to the treasure? Solve this problem by representing the linked possibilities with letters, representing the facts as Boolean formulas, and making derivations to show which are really possible. The method is as important as the answer

9) a) make C-derivations for the following. (They all have just two atomic propositions, so they will not get very big.)

i) **(p ⊃ (q ⊃ p))** ii) **(p ⊃ q) & (p & ~q)** iii) **(p ⊃ q) v (q ⊃ p)**

iv**) (p &~p) ⊃ q** v) **(p ⊃ q) & (q ⊃ p)**

(b) these C-derivations show that some of the formulas in (a) are tautologies, some contradictions, and some neither. Which is which?

10) Here are four pairs of formulas. One pair are contradictories (if either is true the other is false), one pair are contraries (they cannot both be true, but they could both be false), one pair are logically equivalent, and one is a logical consequence of the other. Show, with C-derivations, which is which.

(i) **p & q** , **p & ~q**
(ii) **p & q** , **p ⊃ ~q**
(iii) **p & q** , **p ⊃ q**
(iv) **p ⊃ (q ⊃ r)** , **(p & q) ⊃ r**

11) We call attributes contrary and contradictory, too: contrary when they cannot both apply to something, and contradictory when if one applies the other does not. Which of the following pairs are contraries, and which contradictories?

(a) green, red (b) even, odd (c) vertebrate, invertebrate

(d) vegetarian, carnivorous (e) democracy, dictatorship

Harder

12) (a) make derivations showing that

(i) **(p & q) ⊃ r** is equivalent to **(p ⊃ r) v (q ⊃ r)**

(ii). **(p v q) ⊃ r** is equivalent to **(p ⊃ r) & (q ⊃ r)**

(b) one of these is intuitively rather surprising. Which and why?

(c) do these equivalences suggest an explanation of why people often confuse conjunction and disjunction, and a very small change in context makes one rather than the other more natural?

13) Describe how to make a derivation that goes on forever.

14) (for mathematicians) Prove by induction on the construction of well-formed propositions that the branches of a C-derivation for any proposition give its disjunctive normal form.

15) (for philosophers) Show with a derivation that **p & ~p ⊨ q .** Anything is a logical consequence of a contradiction. (Do you see why can be given this interpretation?) In what ways is this believable and in what ways implausible? Does it help to distinguish between logical consequence and reasonable ways of arguing?

SOLUTIONS TO THE AS YOU GO EXERCISES

All of these have other solutions, too. If you are unsure whether your solution is correct, ask.

1)

```
 1        f ⊃ t
 2       ~f ⊃ t
        /       \
 3  ~ ~f    4    t        (2: ⊃)
    /   \
  ~f     t
   X
```

2)

```
 1          (p & q) v (r & ~q)
 2            q
           /                  \
 3         p & q          4  r & ~q          [v: 1]
 5           p            6  ~q
                              X
```

3)
(a)

```
 1          (p v q) & (~p ⊃ s)
 2           ~p
            p v q                      [1: &]
            /    \
```

```
        p          q
        X

(b)  1     (p v q) & (~p ⊃ s)
     2            ~p
           ─────────────────────
              ~p ⊃ s                  [1: &]
             /          \
        ~~p               s
         X

4) a) 1          ~(p ⊃ q)
      2           q v r
                 ─────────
                 p & ~q          [1: ~⊃ ]
                 ~q
                 /   \
                q      r         [2: v]
                X
```

b) . one way

```
1 p ⊃ (q ⊃ r)
2  p & q
      p              [2: &]
      q              [2: &]
   /        \
~p           q ⊃ r   [1: ⊃]
 X           /  \
         ~q      r
          X
```

another way

```
1 p ⊃ (q ⊃ r)
2  p & q
 /         \
~p          q ⊃ r   [1: ⊃]
 p  [2: &]  /      \
X          ~q       r
            q              [2: &]
            X
```

5) a)

```
    1  ~p v q
        /     \
p ⊃ q      ~(p ⊃ q)      (EM)
           p & ~q
           p
           q
         /     \
       ~p       q
        X       X
```

b)

```
1        p ⊃ q
        /      \
~p vq          ~(~p v q)          (EM)
               ~~p & ~q           [~v]
               ~~p                              [&]
               ~q                               [&]
               /   \
             ~p     q                   [1: ⊃]
              X     X
```

c)

```
    p
/       \
~p      ~~p           [EM]
X
```

chapter eight: direct and indirect argument

8:1 (of 5) getting arguments from other arguments

The main topic of this chapter is how we can simplify arguments, the discovery that a conclusion is a logical consequence of premises, by including our information the fact that a type of argument is valid. There will be many examples of this. A related topic is the contrast between arguments that branch, as many of the derivations do, and arguments that proceed in a step-by-step linear fashion.

First to rehearse the rules consider or reconsider the four basic consequence relations centering on IF. I give derivations showing that the rules do produce logical consequences. Parts of these derivations are marked in blue; this will be relevant soon.

```
  p
 p ⊃ q
_______
 /    \
~p     q
X
```

p, p ⊃ q ⊨ q
(modus ponens)

```
  ~q
 p ⊃ q
_______
 /    \
~p     q
       X
```

~**q, p ⊃ q ⊨** ~**p**
(modus tollens)

```
  p ⊃ q
  q ⊃ r
_________
  /    \
p ⊃ r   ~(p ⊃ r)     (EM)
         p & ~r
          p
          ~r
        /    \
       ~p     q
       X     / \
           ~q   r
            X   X
```

p ⊃ q , q ⊃ r ⊨ p ⊃ r
(transitivity of IF)

```
        p ⊃ q
 _________________
  /               \
~q ⊃ ~p          ~(~q ⊃ ~p)
                 ~q & ~~p
                 ~q
                 /   \
               ~p     q
                X     X
```

p ⊃ q ⊨ ~q ⊃ ~p
(contraposition)

These are basic uses of the derivation rules. If you are even a little bit uncertain of them you should ask a question in class. You should be able to say which rule is used on every line. They all have two premises and the branches that lead to their conclusion are short. If we used them as basic rules instead of the rules that we stated in the previous chapter we could have had a system where derivations do not branch. But that would have brought other complications. One small complication is that any such rule will need two premises to derive one conclusion, while the rules in the previous chapter each had a single premise as input.

>> give examples of persuasive everyday arguments fitting the patterns of *modus ponens*, *modus tollens*, transitivity of IF, and contraposition

>> one of these rules can be got from the others. which?

>> give an example of a valid rule involving two premises and v, but no branching

>> these derivations involve branching. so why could the r*ules* be part of a system where derivations do not branch? (if you do not see why, you should ask.)

Of these, the derivation of the transitivity of IF was the longest. Here is another approach to it, with a C-derivation but one that uses three premises. Note that the part in blue is the same as in the earlier derivation.

```
 p ⊃ q
 q ⊃ r
~(p ⊃ r)
------------
 p & ~r
   p
  ~r
 /     \
~p      q
x      /  \
      q    r
     x     x
```

All the branches close. That means that there are no truth assignments that make all three premises t rue. Now — this is a subtle piece of reasoning of a kind that occurs several times in this chapter — consider a truth assignment that makes the first two premises true, call it $\mathcal{T}$. There is no truth assignment that makes all three premises true, as the derivation shows. So $\mathcal{T}$ in particular does not make all three true, but it does make the first two true. Therefore it must make the third premise false. But to make that first premise, **~(p ⊃ r)** , false, it must make **p ⊃ r** true. So any truth assignment that makes **p ⊃ q** and **q ⊃ r** true must also make **p ⊃ r** true.

The conclusion of this reasoning is not very surprising. But it illustrates an important principle:

> If there is no truth assignment making all of a set of propositions true, then the negation of any one of them is a logical consequence of the others.

>> but what if the "others" are inconsistent all by themselves? does this make a problem?

Compare the derivation from the three premises in which all branches close to the derivation of the negation of the third premise from the other two. You'll see that the biggest sub-tree is the same in both of them. You can transform either derivation into the other just by shifting this part of the three-premise derivation to the EM part of the two-premise one. This illustrates another important principle:

> A derivation with premises A_1, A_2, ..., A_n in which all branches close can be transformed mechanically into a derivation with premises A_1, A_2, ..., A_{n-1} in which only branches with $\sim A_n$ do not close.

.>> take some particular everyday argument that assuming some things are true, some other proposition has to be false. turn it into an argument that if we made those assumptions and also that this proposition is true, then we would get contradictory conclusions.

8:2 (of 5) linear and branching arguments

The examples in the previous section focused on reasoning with conditionals to prepare for a contrast between linear arguments, that go straight from premises to conclusions, and branching or indirect arguments that go from premises to several possibilities, some of which are then ruled out by other premises. In the history of philosophy and logic, most attention has been paid to linear reasoning. In fact it has often been assumed that all reasoning has to be like this. (Mathematical proofs are usually presented as linear sequences, and of course when we speak one sentence follows another, even if in making sense of it we reconstructed as a branching structure. I suspect that some philosophers have liked the image of the domineering intellectual, who forces you to a destination prepared in advance along a carefully mapped-out path. No deviations permitted.) Begin with some examples of linear arguments, and derivations that correspond to them.

LINEAR VERSION

1) if it rains the dam will burst
2) if the dam bursts the village will flood
3) if the village floods people will drown
4) (just seen the forecast) it will rain
5) the dam will burst (3, 1: *modus ponens*)
6) the village will flood (5, 2: *modus ponens*)
7) people will drown (6, 3: *modus ponens*)

BRANCHING VERSION

```
 r ⊃ b
 b ⊃ f
 f ⊃ d
 r
 ----
  /    \
~r      b
 X     /  \
     ~b    f
      X   / \
        ~f   d
         X
```

>> make a shorter argument from these premises to this conclusion using transitivity of IF together with *modus ponens*

LINEAR VERSION

BRANCHING VERSION

1) if it rained the dam burst
2) if the dam burst the village flooded
3) if the village flooded people drowned
4) (just read the news) <u>no one drowned</u>
5) the village did not flood (4, 3: *modus tollens*)
6) the dam did not burst (5, 2: *modus tollens*)
7) it did not rain

```
          r ⊃ b
          b ⊃ f
          f ⊃ d
        ___~d____
         /       \
       ~f         d
      /   \       X
    ~b     f
    / \    X
  ~r   b
       X
```

From these examples we can see that arguments that proceed directly without branching from premises to conclusions can be transformed into branching arguments, as long as enough of the branches can be eliminated (closed). And we can see a tendency for linear arguments to use rules that draw on more than one premise.

\>> why just "a tendency"? are there exceptions?

These transformations make sense of some patterns of argument that we frequently use. In the previous section we saw how we can transform an argument that some premises have no model into an argument that the negation of one of them follows from the others. These were both branching derivations in the examples, but note how the basic principle applies to examples such as the following.

1) if it rained the dam burst
2) if the dam burst the village flooded
3) if the village flooded people drowned
4) no one drowned
<u>5) (suppose for the sake of argument that) it rained</u>
6) the village did not flood (4, 3: *modus tollens*)
7) the dam did not burst (5, 2: *modus tollens*)
8) it did not rain
but (8) contradicts (5) therefore
9) it did not rain

This argument has two parts. In the first part we show that 1)—5) lead to a contradiction, and then in the second part we build on this to show that 1)—4) lead to the negation of 5), that is, 8). But why is this a good way of arguing? Why does it give true conclusions when the premises are true? The reason is given by the principle above: when a set of premises has no model, any one of them is a consequence of the others.

This is called an argument (or proof) by contradiction, or sometimes a *reductio ad absurdum.* (There are a lot of Latin phrases in logic. Logic has changed enormously in the past two hundred years, but it has a long history and so some of the terminology goes way back.) We often use arguments like this. One reason is that we can often describe the features some individual or situation would have to have if some assumption were true, so if we can then show that these features are impossible we can conclude that the assumption is false. This is a particularly appealing move in mathematics, where we can say "suppose there is an x such that ... " and then derive a contradiction, proving that there is no such x. The most famous such argument is the proof that there is no greatest prime number, which begins by supposing that there is a n such that n is prime and no number greater than n is also prime. We then prove that n!+1 must also be prime, contradicting the assumption. A simpler argument shows that there is no greatest integer. Suppose that numbers got no bigger than some n. But then n+1 (or 2n, or n!) would all be bigger than n, so we would have both **p** (n is the biggest) and **~p** (some numbers are bigger than n.)

>> but the contradiction cannot follow from that greatest number assumption alone. we have to assume also that, for example, 2n is always greater that n. does this open up a loophole in the proof, so there could be a greatest integer after all?

Arguments by contradiction are *indirect arguments*. They show that a conclusion is a consequence of some premises not by deriving it directly from them but by showing that a different argument leads from related premises to a related conclusion. We use indirect arguments a lot. Sometimes they make it easier for us to get an intuitive grip on what is being argued for and how the argument supports it. And sometimes they make the argument simpler, for example by avoiding too many branchings. They can allow a linear step-by-step argument to do the work of a more confusing branching argument.

8:3 (of 5) other indirect arguments

Argument by contradiction is not the only form of indirect argument. In this section I describe three more. As it happens, we can associate each with a Boolean connective: argument by contradiction with NOT, and the argument forms described below with NOT, AND, OR, and IF, each one working because of an important feature of the relation of logical consequence.

With IF we can use the method of *conditional argument* (sometimes called *conditional proof*.) Informal conditional arguments like the following are very common.

> Suppose that the earth's population continues to increase and that we manage to feed everyone for the next century. Each kg of meat needs 16 kg of animal feed, so it will not be efficient, or even possible, to do this by feeding everyone meat. So supposing that there is not mass starvation, we won't be eating much meat.

The "supposing" in the conclusion is a disguised conditional: if not starvation then not meat. The argument is an argument, not a proof: its assumptions could be challenged. But now the important point is that it starts with some assumptions, including the assumption that there is not going to be starvation, gets to a temporary conclusion that we will consume little meat, and then moves to the weaker conclusion that IF there is no starvation there will not be much meat. This final conclusion does not need the assumption that there will not be starvation, for it just asserts that assuming, rightly or wrongly, that there will be no starvation then we should not expect much meat. The conditional conclusion might be what we mean to persuade the audience of. But it might also be that we intended to argue for the stronger, non-conditional conclusion, but the audience would not accept the assumption of non-starvation. Then we can back-track and say “well, at any rate you’ll accept that *if* there is no starvation then we’ll be eating a lot less meat.”

As with an argument by contradiction, conditional argument links both to a fact about logical consequence and a way in which derivations can be transformed. The fact is this:

> If a conclusion C is a logical consequence of some premises then the conditional with one of them as antecedent and C as consequent is a logical consequence of the remaining premises.

If $A_1, A_2, \ldots, A_n \models C$, then $A_1, A_2, \ldots, A_{n-1} \models A_n \supset C$

To see that this fact is so, suppose that C is true in all truth assignments in which A_1, $A_2, \ldots, A_n$ are true. Now consider a truth assignment in which $A_1, A_2, \ldots, A_{n-1}$ are true. Either A_n is true in this truth assignment or it is not. If it is then all of $A_1, A_2, \ldots, A_n$ are true in the truth assignment and so $A_n \supset C$ is also true. If it is not then $A_n \supset C$ is true since A_n is false. Either way, $A_n \supset C$ is true in the truth assignment.

To see the transformation, consider an example. Here is a derivation of **r** from **p & q** and **p ⊃ (q ⊃ r)** .

```
   p & q
 p ⊃ (q ⊃ r)
    p
    q
  /       \
~p         q ⊃ r
 X          /  \
          ~q     r
           X
```

And here is a derivation of (**p & q**) ⊃ **r** from **p ⊃ (q ⊃ r)** alone.

```
              p ⊃ (q ⊃ r )
             /            \
(p & q) ⊃ r                ~((p & q) ⊃ r)          (EM)
                             (p & q)& ~r
                                  p & q
                                  p
                                  q
                                  ~r
                                /      \
                              ~p        q ⊃ r
                               X       /   \
                                     ~q      r
                                      X      X
```

Notice how the second derivation adapts the first one by using EM so that all further branches close from the negation of the conditional we want to derive. We can always do this to get a derivation of a conditional conclusion from one fewer premise. Exercises **4)** and **5)** give more examples of this transformation, and after doing them you should be able to write out the transformed derivation whenever the technique applies.

>> this transformation suggests that conditional arguments can be seen as a special case of argument by contradiction. say more about this.

The next form of indirect argument is associated with OR. I will call it disjunctive argument. Like conditional argument it allows us to weaken the premises of an argument, so we can back off a bit, not assuming so much. But unlike both argument by contradiction and conditional argument it builds on *two* arguments and combines them into one. The general fact is this.

disjunctive argument:

> If C is a logical consequence of some premises plus the additional premise A, and C is also a logical consequence of the same premises plus B, then C is a logical consequence of these premises plus A v B .
>
> If $A_1, A_2, \ldots, A_n, A \models C$ and $A_1, A_2, \ldots, A_n, B \models C$
> then $A_1, A_2, \ldots, A_n, A \vee B \models C$

One time we use this style of indirect argument is when we know we can get a conclusion from either of two assumptions, but we are not sure which of these assumptions is correct. So we retreat to "one of them must be right, and as long as either is the conclusion holds." For example

> Bea plans to come to campus tomorrow, and if she does then she will drop by and tell you what she thinks of your plan. So be prepared to be roundly criticised by a fierce lady. There has been a lot of flu around lately, which would keep her at home. But supposing she is ill and stays home, she will still have internet access, so a blistering email is sure to find you. Either way, you are going to be the target of sharp criticism.

Disjunctive argument can also be described in terms of a way derivations can be transformed. And again I am going to give an example that should let you see the general pattern. We start with two derivations.

```
      p ⊃ r                              p ⊃ r
      q ⊃ r                              q ⊃ r
      r ⊃ s                              r ⊃ s
      ____p____                          __q____
     /   \                                 /   \
  ~p          r                          ~q     r
   X     /  \                             X    /   \
       ~r    s                               ~r      s
        X                                     X
```

p ⊃ r, **q ⊃ r**, **r ⊃ s**, **p** ⊨ **s**

p ⊃ r, **q ⊃ r**, **r ⊃ s**, **q** ⊨ **s**

We can combine these two into a single derivation, combining the two third premises **p** and **q** into a single premise **p v q**.

```
              p ⊃ r
              q ⊃ r
              r ⊃ s
             _p v q_
             /      \
           p          q
        /    \      /    \
      ~p      r   ~q       r
       X    /  \   X     /   \
          ~r    s       r  s
           X            X
```

p ⊃ r, **q ⊃ r**, **r ⊃ s**, **p v q** ⊨ **s**

It is clear what we have done. In the combined derivation we have used the disjunction **p v q** with the OR rule to branch into two sub-trees, each of which is the same as one of the two original derivations and each of which has **s** on all its non-closed branches. (All the same, you should make sure that you see how each step of all three is in accord with the derivation rules.)

>> disjunctive argument can, in a way, do the work of our original branching OR rule, without branching. explain.

One more indirect argument form, again one that takes two logical consequences and combines them. I shall call it conjunctive argument. The idea is that when we can conclude C from some premises and also conclude D from the same premises, then we can conclude C & D from them. (This is hardly surprising.) For example we might argue

> I'm sure you are going to lose your job. You took the boss's car keys away from him when he was too drunk to drive. He hates people seeing his failings and while he cannot fire you for that there are many other reasons he can invent. I expect your house is going to be repossessed, too. The boss is a director of the bank that holds the mortgage, and his vindictiveness knows no bounds. So this time next month you will be jobless and homeless.

Stating the principle more formally, it is

conjunctive argument:

If $A_1, A_2, \ldots, A_n \models C$ and $A_1, A_2, \ldots, A_n \models D$

then $A_1, A_2, \ldots, A_n \models C \,\&\, D$

For an example of how to combine two derivations in accordance with the principle, consider these two

~ p

**/ **

p ⊃ q **~(p ⊃ q)** **(EM)**

p & ~q

p

X

~p ⊨ p ⊃ q

~

~ p

**/ **

r ⊃ ~p **~(r ⊃ ~p)(EM)**

r & ~~p

~ ~p

X

~p ⊨ r ⊃ ~p

These can be combined to get **~p ⊨ (p ⊃ q) & (r ⊃ ~p)**

```
                ~p
          /            \
( p ⊃ q) & (r ⊃ ~p )     ~((p ⊃ q) & (r ⊃ ~p ))
                          ~(p ⊃ q) v ~(r ⊃ ~p )
                             /             \
                          ~(p ⊃ q)        ~(r ⊃ ~p )
                           p & ~q           r & ~~p
                             p                ~~p
                             X                 X
```

As with the others, comparing the parts with the same colors should show you how the combining works. And as with the others there are relevant exercises at the end of the chapter. Note that conjunctive argument and conditional argument give ways of getting conclusions that are not atomic. And note that argument by contradiction, conjunctive argument, and conditional argument introduce a more complicated conclusion, while disjunctive argument introduces a more complicated premise.)

Indirect arguments are got by combining direct arguments. All my examples have been of combining complete and correct derivations. It is worth pointing out, though, that we often combine direct arguments that are much less formal and precise into indirect arguments. For example we can reason very informally that an assumption leads to a ridiculous conclusion, and then say "so it has to be wrong". Or we can have loose non-deductive reasons for believing that each of two assumptions would support a conclusion, and then say "so if we assume just that one of these is right, we can get this answer. We don’t have to decide which one it is.”

8:4 (of 5) consistency and independence

We have seen how to use derivations to show that a conclusion is a logical consequence of a set of sentences, and how to show that a set of sentences is inconsistent, having no model. These are closely related, since a conclusion is a logical consequence of premises if the set of the premises and the negation of the consequence, is inconsistent.

>> why the negation of the conclusion?

>> the discussion has been in terms of truth assignments. but it is full models rather than truth assignments that are most relevant to whether a real English sentence is true or false. would the considerations of this chapter transferred smoothly to a vocabulary of models?

There are other logical properties of sets of sentences, and derivations can give us information about them. The most basic is *consistency*, sharing a truth assignment. For example the set {**p ⊃ q, p ⊃ ~q }** is, perhaps surprisingly, consistent. If we make a C-derivation for it we find this

```
1                          p ⊃ q
2                          p ⊃ ~q
                     _______________
                    /               \
                 ~p                   q                      (1: ⊃)
               /     \             /     \
            ~p         ~q        ~p        ~q                (2: ⊃)
           /  \       /  \      /  \      /   \
          q   ~q     q   ~q    q   ~q    p     ~p            (EM)
                                    X    X      X
```

The unclosed branches give three truth-distributions, one in which **p** is false and **q** is true, and one in which **p** is false and **q** is false, and one in which both are false. Both formulas are true in all three of these.

>> but there are four unclosed branches. so why are there only three truth distributions?

Surprising combinations of sentences can be logically consistent, especially when there is an improbable but logically possible situation in which they are all true. So, sticking with the same example, "If I like someone I'll always give them a present" and "If I like someone I'll never give them a present" can both be true as long as I don't like anyone. There are many examples in the history of science and philosophy where combinations that were thought to be too unlikely to consider, or even contradictory, turned out to be possible or even true. Examples are curved space, time with no beginning, voting situations where everyone's preference is weighted the same but no one gets what they want, or mental decisions that always accompany action but do not cause it.

Another property of sets of sentences is *independence*. One sentence is independent of others when there is at least one model which makes both it and the others true and at least one model which makes it false and the others true. The truth value of the sentence is independent of them. An example that is important in the history of mathematics is the independence of the parallel postulate (that through any point there is exactly one line parallel to a given line) from the other axioms of Euclidean geometry. Geometers spent a lot of time trying to prove that the parallel postulate was a logical consequence of the others, until they found models in which all the other axioms are true and there are many or no lines through some points parallel to some lines. This led to the study of non-Euclidean geometries, which according to Einstein's theory of General Relativity describe the true structure of space-time. (A simple example of such a model is the surface of the earth, where all the longitudes going through the poles are parallel. This gives a model in three dimensions of a two dimensional non-Euclidean geometry.)

The example of non-Euclidean geometry made the strategy of showing that one sentence is independent of others, by giving a model for them in which it is not true, important in mathematics and elsewhere in science. And it stimulated the related ideas of a model and of formal derivation in logic. Independence and consistency are closely related ideas, since consistency, the existence of a model, is associated with not being able to derive a contradiction from the sentences, and independence is associated with not being able to derive one sentence from the others. A different way of putting it is that one sentence S* is independent of others S_1, ..., S_n when the sets {S_1, ..., S_n, S*} and {S_1, ..., S_n, ~S*} are both consistent. You can assert or deny S* while asserting the others without contradicting yourself.

>> why are these different ways of saying the same thing?

Here are two examples that show the independence of one Boolean proposition from others. First, trivially to get the idea across, **p** is independent of the disjunction **p v q**. The two truth distributions that (a) assign T to both p and q , (b) assign F to **p** and T to **q** both make the disjunction true, but one makes **p** true and the other makes **p** false. For a more interesting example consider the independence of **r v q** from {**p ⊃ q**, **r ⊃ p**}. (It is more interesting because all the propositions are molecular, and there are atomic propositions in common between the independent proposition and the set it is independent of.) Independence here can be shown with a pair of C-derivations.

For one direction of independence:

```
                 r v q
                 p ⊃ q
  _______________r ⊃ p_________
          /                  \
      r                         q
    /    \                  /       \
  ~p       q             ~p          q
 /   \    /  \          /  \       /   \
~r    p  ~r   p       ~r     p    p     ~ p
X     X  X            X      X   / \    /  \
                                r  ~r  r   ~r
```

This gives us three truth distributions, **p— q—r**, **p—q—~r**, and ~ **p—q—~r,** where all three propositions are true.

In the other direction:

```
              ~(r v q)
              p ⊃ q
          ____r ⊃ p____
              ~r & ~q
                ~r
                ~q
              /     \
          ~p          q
        /     \       X
    ~r          p
                X
```

This gives us just one truth distribution, **~p—~q—~r,** where the disjunction is false and the other two are true. But one in each direction is all you need to show independence.

8:5 (of 5) consistency and problem-solving

With truth-distributions for Boolean propositions, there is a simple connection between derivations and truth distributions. Propositional logic is *complete,* meaning that whenever one sentence is a logical consequence of a set of others there is a derivation showing this fact. Moreover all the models that are needed to show consequence and independence can be represented as truth-distributions over finite numbers of atomic propositions. In fact, Boolean logic would be unchanged if all models had finite domains of individuals. As we move beyond Boolean logic to more complex logical systems this tidy situation becomes steadily more tangled. quantifier logic, the topic of the final three chapters, is also complete, though it forces us to consider infinite models, and showing that one formula follows from others is far from automatic. In this book we cannot go far enough really to engage with these issues, but I shall try to show how even Boolean logic hints at them.

The central point, though we do not have the resources to state it carefully here, is that consistency (and thus independence) is a more difficult topic than logical consequence. It is often harder to show. And we can learn more interesting things about a proposition by knowing what others it is consistent with than by knowing what it follows from. Boolean logic gives the barest hint of this fact.

The hint goes like this[21]. Often we can show that one proposition is a logical consequence of another without showing that either is consistent. But finding a model for either takes more work. Here is an example. It deliberately uses complicated-looking propositions, but they are not quite as horrible as they seem. Consider the set of propositions A

A: **p , p ⊃ (q v ~r) , q ⊃ (t v s) , ~r ⊃ (t v s) , t ⊃ ~p , s ⊃ ~t ,**
~s ⊃ (~q & r) , ~t ⊃ (~r & q)

or the single proposition **a** that is the conjunction of all eight of these. It is trivial that **p** is a logical consequence of the set A. After all, it is the first member (or the first conjunct of **a**.) But it is not so trivial that **A** or **a** is consistent. It has a whiff of inconsistency in that it resembles the set {**p , p ⊃ q , q ⊃ r ,r ⊃ ~p**}**,** which is definitely inconsistent, since in three steps of *modus ponens* we can deduce **q**, **r**, **~p** from it, contradicting **p**. To check with a truth table would need a table with 32 rows, and a C-derivation would have many

21 This is a simplification. The full situation is very complicated, and not everything about it is understood, even though propositional logic is a very simple system. There is a big literature, mostly in computer science. My aim is just to give you a flavour of distinctions that are starker in more complicated logical systems.

branches. In fact A is consistent. It is made true by the truth assignment in which **p**, **q**, and **s** are true and **t**, **r** are false. Once we try this truth assignment it is easy to see that it makes A and **a** true, but working through all the truth assignments for the five atomic propositions would be a lot more work. In general, checking whether a derivation is correct is easy, knowing whether one formula follows from another is harder, and knowing whether one formula does *not* follow from another is hardest of all. In more complex systems of logic these comparisons become even more stark. (I chose the particular set A because it is trivial that **p** is a consequence, confusing-looking enough that you would be unsure whether it is consistent, and easy to see that it is verified by the truth assignment. It is not very special; there are many other possible examples.)

A metaphor may help. It is as if we were trying to push complicated shapes through a complicated hole. (A kind of filtering.) The hole is the outline of the Americas, say. We consider shapes to see if they will fit. Some obviously will or will not, and sometimes we can see a reason why the shape will not fit through the hole. (The indentations of the Gulf of California or the Bay of Fundy, get in the way perhaps.) But to know that a shape definitely will fit through the hole you will have to check many details. (Perhaps there is a clever way of tilting it so one of its bumps fits through Cape Breton, though it looks like it gets blocked.) This is a lot harder than noticing one mis-match that will prevent the shapes from fitting. Knowing that a shape will not fit is like knowing that a proposition is inconsistent or that a conclusion follows. Knowing that a shape will fit is like knowing that a proposition is consistent or that one proposition does not follow from others: typically more difficult.

>> why is knowing that a proposition is inconsistent like knowing that a conclusion does not follow?

There is a connection with issues about direct and indirect arguments that is worth mentioning. The theme is that it is in general harder to show that a proposition is consistent than that it has a logical consequence. (Though showing that a consequence follows can be hard, and showing consistency can *sometimes* be easy.) How are we to measure and compare complexity of derivations? When derivations do not have many or very long branches, simple length makes sense as a measure. But when as in the derivations we are studying there are many branches, things are more complicated. A C-derivation that showed consistency might do so with a very short branch although other branches were long. (A proposition can have simple models and also complicated ones.) The appropriate measure might be the total length of all branches down to the point where one ends and defines a truth assignment. A derivation system in which derivations do not branch will not usually give information about whether propositions are consistent. If a proposition is not consistent one can sometimes go on deriving consequences from it for a long time without deriving an explicit contradiction of the form P & ~P. In richer logical systems than Boolean logic, where models can be infinite, there is a correspondingly greater danger that one's assumptions are inconsistent but one has never come across a contradiction. Notice that we cannot always show that A is consistent by showing that ~A is inconsistent, since often we have a sentence, A, for which neither A nor ~A is inconsistent. Examples are **p**, **q**, **p & q**, **p v q**, **~(p & q)**, and many others: all of these have models, but their negations also have models.

>> in what areas of philosophy, or elsewhere, do you think it most likely that there are hidden contradictions in our assumptions that we have not yet discovered?

This is one reason why many intellectual and practical problems are hard. We are often faced with a set of conditions (assumptions, facts, constraints) that have many solutions, so we cannot get an answer by deducing from them that some particular solution must be right. Instead we want to know whether some possible solutions are consistent with the assumptions. We want to know whether some suspect could have committed the crime, or whether some configuration of space-time is compatible with general relativity. But even when we have found a candidate solution, knowing for sure that it is consistent with the given conditions is often very hard.

words used in this chapter that it would be a good idea to understand: branching argument, argument by contradiction, conditional argument (proof), conjunctive argument, disjunctive argument, independent propositions, linear argument, reductio ad absurdum.

exercises for chapter eight

A- core

1) Which of the arguments below is an instance of disjunctive argument , which conditional argument, and which argument by contradiction?

(i) Suppose someone was taking pictures when we had that nude swim. They will be all over the Internet next week. Then your father will never offer you that public relations job. We will need another source of income. So if someone was taking pictures we will be on the job market soon.

(ii) There is a rumour that the barn was made of paper painted to look like wood. We saw the barn last week standing a quarter-mile from the farmhouse. It certainly seemed solid. And in fact there had been a tornado right there a month before our visit. A paper barn would have been blown to smithereens. So it could not have been a painted paper barn.

(iii) That might have been Sciocchezzo Corleone disguised as a nun who just walked by. The only other possibility is that it was a real nun. But if it was the question still arises how she got past the police line. So we had better ask the supposedly alert and honest officers down there some probing questions.

2) Make your own examples of disjunctive argument, and-introduction, condition argument, and argument by contradiction, on topics as different as you can from

question 1).

3) Are there typical situations where we use these forms of argument? Think of reasoning from uncertain information and dealing with situations where the existence of particular kinds of objects cannot be directly verified. Be as specific as you can about how particular situations encourage particular forms of argument. If there are such correlations, are they consistent with reformulating indirect arguments in direct form?

4) (a) make two derivations showing that **p & r** $\models$ **r** and that **~t, t v r** $\models$ **r**, and combine them to make a derivation showing that **(p & r), ~t, t v r** $\models$ **r**

(b) Make a derivation showing that **~q, p & ~p** $\models$ **q**, and modify it into a derivation showing that **~q** $\models$ **(p & ~p)** ⊃ **q** .

(c) Make a derivation showing that {**p, p** ⊃ **q, q** ⊃ **~p** } is contradictory, and modify it into a derivation showing that **p, ~q p** $\models$ **~(p** ⊃ **q)**.

5) Each of these derivations can be used to show several things. Below each derivation are some possibilities. Say which ones are shown by each derivation.

```
1            p v (q v r)
2                 ~p
3                 ~q
        ____________________
        /                  \
       p                 (q v r)
       X                 /      \
                        q        r
                        X
           p v (q v r) , ~p, ~q ⊨ r
  { p v (q v r) , ~p, ~q , r } is inconsistent
```

```
1                 p ⊃ q
               /          \
~q ⊃ ~p                 ~(~q ⊃~p)   (EM)
                         ~q & ~~p
                             ~q
                            ~~p
                           /    \
                         ~p      q
                          X      X
                    p ⊃ q is a tautology
                    p ⊃ q ⊨ ~q ⊃ ~p
                ~q ⊃ ~p is a contradiction
```

```
1             p
2          ~(p & q)
3          p ⊃ (q v r)
           ~p v ~ q
            /     \
     ~p             ~q
     X             /  \
                 ~p    q v r
                 X     /    \
                      q      r
                      X
~(p & q) , p ⊃ (q v r) ⊨ p ⊃ r
p, ~(p & q) , p ⊃ (q v r) ⊨ r
```

```
1             ~((p ⊃ (q & r ))
2                  q & r
              ______________
               p & ~(q & r)
               ~(q & r)
               ~q v ~r
                /    \
              ~q      ~r
               X       X
  { ~( (p ⊃ (q & r ) ), q & r } is inconsistent
     ~( (p ⊃ (q & r ) ), q & r ⊨ ~q v ~r
         ~( (p ⊃ (q & r ) ) ⊨ ~(q & r)
```

```
1          ~(p ⊃ q ) & ~p
2          ~(p ⊃ q)
3          ~p
4          p & ~q
5           p
            X
~(p ⊃ q ) & ~p ⊨ p
~(p ⊃ q ) & ~p is a contradiction
~~(p ⊃ q ) & ~p is a tautology
{ ~(p ⊃ q ) & ~p } is inconsistent
```

```
(p ⊃ r) & (q ⊃ r)           (p ⊃ r) & (q ⊃ r)
       p                            q
   __________                 __________
     p ⊃ r                        q ⊃ r
     /   \                        /   \
   ~p     r                     ~q     r
    X                                  X
                 (p ⊃ r) & (q ⊃ r) ⊨ r
           (p ⊃ r) & (q ⊃ r), p v q ⊨ r
```

6) Show that the following sets of sentences are inconsistent

i) **{p &q, ~p & r }**

ii) **{ p ⊃ r, q ⊃ s, p v q, ~(r v s) }**
iii) **{ ~(p&(q v r)), s, s ⊃ p, s ⊃ r }**

B-more

7) a) Show that the following formulas are tautologies, either by constructing a C-derivation where it is true on all branches, or by constructing a derivation from its negation on which all branches close.

i) **p ⊃ p**
ii) **~(p & ~p)**
iii) **p ⊃ (~p ⊃ q)**
iv) **~(p v q) ⊃ ~p**
v) **~(p & ~q) ⊃ (p ⊃ q)**
vi) **(~p v q) ⊃ (p ⊃ q)**
vii) **(p ⊃ q) ⊃ (~p v q)**

b) Why are these two equivalent: (i) constructing a C-derivation where a formula is true on all branches, and (ii) constructing a derivation from its negation on which all branches close.

8) (a) Show that each of these sets of formulas is consistent, by finding a truth assignment that makes all of its members true.

i) { **p ⊃ q**, **p ⊃ ~q** , **~p** }
ii) { **p ⊃ q** , **~(q ⊃ p)** }
iii) { **p & q** , **p v ~q** }
iv) { **~(p & ~q**) }
v) {**(p v ~q) & (q ⊃ p)** }

(b) iv) and v) consist of just one proposition. Show that its negation is also consistent.

9) Here are four informal arguments in ordinary language:

✍ It may rain next week, or it may Snow. And it may stay warm, or it may get cold. Or it may even be warm and then cold later in the week. Great-Aunt Sophia will have to leave her apartment some time, and she may use her walker or she may try to walk unaided. Consider this possibility: Snow followed by warming and then cold. That means ice, and without her walker her balance is very bad. Assume now that she does go out unaided under those conditions. She will surely end up in hospital. So if we have that weather pattern and she ignores her walker you will have an aunt in the emergency room.

B It may rain next week, or it may snow. And it may stay warm, or it may get cold, or it may even be warm and then cold later in the week. Great-Aunt Sophia will have to leave her apartment some time, and she may use her walker or she may try to walk unaided. Consider this possibility: Bow followed by warming and then cold. That means ice, and without her walker her balance is very bad. Of course, she knows that, and she will not do anything that will land her in hospital. Suppose she did go to the store on ice without her walker. There would inevitably be a fall. But we know she won't allow that to happen. So she will not even try to leave her apartment under those conditions without her walker.

C It may rain next week, or it may Bow. And it may stay warm, or it may get cold, or it may even be warm and then cold later in the week. Great-Aunt Sophia will have to leave her apartment some time, and she may use her walker or she may try to walk unaided. Actually the forecast is for Bow followed by warming and then cold. Suppose it's the walker. She will take so long to get to the store that she'll be frozen. Suppose it's on foot. She will fall. Either way you'll have a sick Aunt to take care of. So be prepared to visit her next week.

D It may rain next week, or it may snow. And it may stay warm, or it may get cold, or it may even be warm and then cold later in the week. Great-Aunt Sophia will have to leave her apartment some time, and she may use her walker or she may try to walk unaided. Or she may take a taxi, even though she is very careful with money. The walker option means slow progress and cold feet, which she dislikes even more than spending money. The on foot option means a great risk of falling, and she won't take that risk. So I guess for once we'll find Sophia in a taxi.

One of these arguments is an elimination of alternatives, one is a conditional argument (conditional proof), one a disjunctive argument, and one is a proof by contradiction (reductio ad absurdum). Which is which?

15) Using the following lettering of the sentences

h = it is warm early in the week
c = it is cold later in the week
w = Sophia uses her walker
f = she goes unaided on foot
t = she takes a taxi
r = she is f**r**ozen
a = she has a f**a**lling **a**ccident
i = she is ill, for example in hospital

consider these four representations of crucial parts of the arguments in 9). For each, state which argument it fits best.

h & c , w v f, (w & c) ⊃ r , (f & h & c) ⊃ a , r ⊃ i , a ⊃ i therefore **i**

ii) **w v f v t , w ⊃ r , f ⊃ i, ~r, ~i** therefore **t**

iii) **(h & c & ~ w)** ⊨ **i** therefore **(h & c & ~ w) ⊃ i**

iv) **~w ⊃ f , f ⊃ i, ~i** therefore **w**

16) Below are three everyday arguments in ordinary English. Which is a *reductio* (proof by contradiction), which an elimination of alternatives, and which a conditional proof?

i) I wonder if Heidi is in town. If she is in town she's staying with Lee. This is confusing, so let's suppose she is in town. If she's staying with Lee they will have gone to the hardware grill. So, supposing she's in town, they've gone to the hardware grill. They haven't been seen there. But they're not invisible, so that's impossible. The guess must have been wrong: she is not in town.

ii) Haven't seen Luke for days. I guess he is either in the bar or in jail. It's a pretty vile brew they serve down there; you can be sure that if he is in the bar we won't see him tomorrow. And you don't get out of jail easy in this town, so if he is in jail we won't see him tomorrow. Sure sounds like we're not gonna see Luke tomorrow.

iii) He won't get an A in this course by acing the exam. In fact the only way he can get an A is by joining the Prof's weird little church: no join, no A. Perhaps he won't join. But suppose we learn that he got an A. That will mean he joined. So if he gets that A he must have joined the church.

7) (a) make a C-derivation showing that **~(p ⊃ q) & ~(q ⊃ p))** is a contradiction, and then turn it into a derivation showing that **(p ⊃ q) v (q ⊃ p)** is a tautology.

(b) These are truths about the material conditional, ⊃ ,that are intuitively wrong for many meanings of the English word "if". Can you give examples of situations where we would deny both "if A then B" and "if B then A".

(b) can you formulate a general principle that includes (a) as a special case?

C – harder

20) Consider strengthened versions of the principles linking indirect and direct argument.

$\mathbf{P_1,\ldots,}$ **and** $\mathbf{P_{n-1}, P_n} \models \mathbf{X}$ (where X is a contradiction) if and only if $\mathbf{P_1,\ldots, P_{n-1}} \models \mathbf{\sim P_n}$

$\mathbf{P_1,\ldots, P_{n-1}, P_n} \models \mathbf{C}$ if and only if $\mathbf{P_1,\ldots, P_{n-1}} \models \mathbf{P_n \supset C}$

$\mathbf{A} \models \mathbf{C}$ $\mathbf{B} \models \mathbf{C}$ if and only if $\mathbf{A \vee B} \models \mathbf{C}$

Show that all three of these are true.

21) A ⊃ B and A ⊨ B are similar, though they are different in that the first says that if A is true then B is true while the second says that if A is true then by logic alone B must also be true. Which of the following, which are correct for all A and B, are no longer correct if we replace ⊃ with ⊨ ?

A ⊃ (A v B) , (A ⊃ B) v (B ⊃ A) , (A & ~A) ⊃ B , A ⊃ (~A ⊃ B) ,

A ⊃ B is true whenever A is false

(Philosophers and logicians have often confused ⊃ and ⊨.)

22) Express the disjunctive normal form theorem as follows

DNF: Every Boolean formula F generates a C-derivation in which F is true on all and only the branches of the derivation.

And express the completeness of propositional logic as follows

C: There is a mechanical procedure which shows when one Boolean formula is a logical consequence of another.

Show that C follows from DNF.

23) We can use some indirect arguments as substitutes for others. Instead of using argument by contradiction we could use the equivalence between A ⊃ (B & ~B) and ~A together with conditional argument. Instead of using conditional argument we can use the equivalence between ~(A ⊃ B) and A & ~B together with argument by contradiction. Say how this can be done.

24) (for philosophers and mathematicians) Read an article on non-constructive existence proofs (such as https://en.wikipedia.org/wiki/Existence_theorem, or most textbooks on the philosophy of mathematics). What is the connection with arguments by contradiction? Does the fact that these proofs have attracted some controversy suggest that there is something dubious about arguments by contradiction? (This is relevant to a full answer to question **3**. But a completely full answer would take a book.)

chapter nine: quantifiers and variables

9:1 (of 8) quantifiers

From now on this book focuses on a new topic, another core area of logic, quantifiers. Many of the ideas are similar to ideas we have already seen, and a lot of what we have already done should have prepared you for them. The really striking difference between the logic of quantifiers and the logic of Boolean connectives that we have studied so far concerns the difference between logical symbols and everyday language. The logic of quantifiers can only be explained in terms of language that is very different from the language we use in everyday communication, at least on the surface. This makes the topic harder: the use of symbols is not just for abbreviation but to express a deeply different approach to language.

Quantifiers are used to say how many individuals in a domain have an attribute or enter into a relation. Logic has traditionally focused on the extreme cases of all, none, or at least one individual, but when we say that for example precisely thirty six of the ostriches have laid eggs we are using a quantifier. The contrast between the way these are handled in logic and in every day language can be illustrated by saying that in languages such as English "all the ostriches laid eggs" is like "Olga and Olivia laid eggs" referring to these two individuals as we would to a single one in "Olivia laid an egg". But in logic we express the same thing with language along the lines of "Take any one of the animals. If she is an Ostrich then she laid an egg." The use of a variable, "she" in this example, is essential. Logic usually uses symbols for the variables and the operators that bind them, similar to our use of variables such as x and y and the **Find** operator. But the contrast is the same whether we use variables as in mathematics or more familiar pronouns, and the main purpose of this chapter is to make this contrast clear and explain its significance.

The most-studied quantifiers are the words "all" and "some", though many other words, such as *a few*, *six*, *many*, *at least three*, *no*, and many others have closely related grammatical and logical features. Words such as *everything, and something,* are closely related, as we shall see.

>> think: what do all these have in common?

It is important to see that there are very different ways of expressing similar ideas to these. Consider, for example, the word *always.* Suppose we say *"Monkeys always have tails (but apes usually do not)."* The most usual meaning for this would be the same as "All monkeys have tails (and most apes do not)". Then, the word "always" is very similar to "all" (and "usually" is similar to "most"). But it also might mean "at all times monkeys have tails (but at many times apes do not)", or even "at all times all monkeys have tails (but at many times some apes do not)". It does have the advantage, though, that "always" does not seem to name some individual or individual with a tail, as "all monkeys" can. This also illustrates a point that we will see often in this chapter and the following ones: quantifier words in ordinary language usually have many meanings. They can mean different things individually, as when "always" can mean "all instances of what we are discussing" and can mean "at all times". And they can have different effects when combined, as when we mix "all" and "some" in "all the ostriches chased some monkey" which, as we will see in the next chapter, can mean that each ostrich had a monkey that it chased or that some unfortunate monkey was the target for all the ostriches. We can make similar points about *sometimes*, *often*, *invariably*, *occasionally*, and other such words.

"All" and "always", in these sentence, are both expressing a quantifier in that they are both describing something that is true of the whole domain in question. They are saying *how many* individuals satisfy a predicate or are connected by a relation. There is a link here with the Boolean connectives and and or. Suppose that we are discussing a domain of three monkeys, **a**lf, **b**rigitte, and **c**aspar, and we say "all these monkeys have tails", or "monkeys of this kind always have tails". Then we are saying something very similar to "Alf has a tail & Brigitte has a tail & Caspar has a tail". Similarly, "some of the monkeys have tails" or "these monkeys sometimes have tails" is similar to "Alf has a tail v Brigitte has a tail v Caspar has a tail". This close connection with conjunction and disjunction is one reason that logicians have paid more attention to "all" and "some" than to other quantifiers. (There are languages that use the same word for "all" and "and", as if for "Alf and Brigitte have tails" we said "Alf and Brigitte: they all have tails.")

(I was being careful in saying that the meanings of "all" and "some" are similar to conjunction and disjunction, respectively, not that they were the same. There are important differences, which we will return to.)

>> what about "every cat occasionally sleeps". how do you think the words "every" and "occasionally" fit together?

>> in the sentence "monkeys always hide when it rains" the word "always" is much less close to the word "all". why is this?

There are three ways of representing quantifiers in most of the world's languages. Many languages use all three to some extent. The first two have already been mentioned: quantifier words that tend to crop up where names do, such as "everything", "some", "a few", and so on, and adverbs such as "usually", "sometimes", and so on. The third way is to use a pronoun with a general meaning. For example, we might say "when one goes to a party one should bring a present", where the word "one" refers to anyone going to a party. Or we might say "if they're friendly to me, I'll be friendly to them", where the context is such that this means that I'll be friendly to anyone who is friendly to me. Or, "who dares wins". (This use of "who" is not very common in English but it is often used in other languages, for example Chinese.) Often we mix these together for the sake of clarity, as when we say "if they're friendly to me, I'll be friendly to them, whoever they are" or "if they're friendly to me, I'll always be friendly to them". I shall use all three ways when trying to explain how quantifiers work in logic.

>> suppose there is a movie called "all the stars". how would you explain the two meanings that "we can see all the stars tonight" can have?

A central aim of quantifier logic is to find a way of expressing quantifiers that is less tied to context than the ways we have in ordinary language. We want the meanings of quantifiers in logic to be more fixed and constant than those in spoken languages, and this leads us to novel ways of expressing ourselves. To get a sense of quite how slippery quantifier words are in spoken languages, consider the word "a", one of the most slippery of them all, and consider what we have to say to make its meaning clear in different contexts.

The indefinite article "a", has one of the widest range of meanings of all words in English. "Some" and "all" are very clear and regular in comparison. Contrast two English sentences: "a frightened cat is a dangerous animal", and "a cat is in the garden." "a" is saying something different in the two sentences. In the first, it is saying something about any or all cats, that if they are frightened they are dangerous. In the second, it is saying something about the garden, that it contains at least one cat. The difference can be seen in models with, say, four individuals. Consider a model with four individuals, where three of them have the attribute of being cats, and all of them may have attributes of being frightened, being dangerous, and being in the garden. To see if the first sentence is true — the frightened cats one — we have to check all three cats and see that each of them is dangerous. We cannot stop after finding one dangerous cat. We have to check all three cats. But to see if the second is true we can stop as soon as we find just one cat that is in the garden. Or to put it in terms that link to earlier ideas, and bring out the general pattern, the first one requires that a search,

Find x: IF Cat**x THEN D**angerous**x**
(**Find x: Cx** ⊃ **Dx**)

get all four objects in the domain, which we can check by considering just the three cats, while the second one requires that a search,

Find x: Cat**x AND G**arden**x**
(**Find x: Cx & Gx**)

get just one object. These are different searches and we are considering different things about them.

>> why can we just consider the three cats, to see that the IF search gets all four objects?

>> can you think of a situation where someone might say "a cat is in the garden" and be discussing all the cats?

This introduces a central theme: the language of quantifiers aims to describe what it is about a model that makes some sentences true in it and other sentences false. To continue with this theme, consider the difference between "Mo hates all butterflies" and "Mo hates only butterflies" (Or "Mo only hates butterflies".) These are clear sentences, though someone could get confused about the difference between them. But if we take a step back we can bring out the contrast by saying that a model makes "Mo hates all butterflies" when for any object **o** in the domain if **o** is a butterfly then Mo hates **o**. The search **Find x: IF B**utterfly**x THEN H**ate**mx** gets the whole domain. But a model makes "Mo hates only butterflies" true when for any object **o** in the domain if Mo hates **o** then **o** is a butterfly. The search **Find x: IF H**ate**mx THEN B**utterfly**x** gets the whole domain. The contrast comes out when we treat the two sentences as requiring conditionals to be true, and they are conditionals that are backwards from one another. It is similar to the contrast between "he'll come if you pay him", and "he'll come only if you pay him". The first says **IF Pay THEN Come** — when you've paid him he is sure to come, while the second says **IF Come THEN Pay** — when he comes you can be sure that he's been paid. (Think about this contrast for a moment.) Quantifiers are variable-binding operators, like **Find x:** or the summation operator.

>> what about "Mo hates butterflies", without the "all" or the "only". does it usually mean he hates butterflies or he hates only butterflies? can you think of a situation in which it is used to communicate the other, less usual, meaning?

>> why did I use the longer phrase "used to communicate the other meaning" instead of just "means"?

>> I described "he'll come only if you pay him" in terms of **IF Come THEN Pay**. But "if he comes then you pay" seems a strange way of saying the same thing. can you change it just a little so that it is more natural? what does this suggest about the English *if/then*? (there are languages which do not have separate words for "all" and "only", so hearers have to figure out which makes most sense in the circumstances.)

9:2 (of 8) complexity considerations

Quantifier logic just is a lot more complicated than Boolean logic. (The complexity is not really apparent until we consider multiple quantifiers, which appear in the next chapter.) One sign is that simple truth assignments are no longer adequate and we must consider fully specified models. Another sign is that the rules for getting the consequences of a sentence are more subtle and allow much more complicated derivations. Quantifiers open up a great range of expressive power.

This makes quantifier logic a challenge to understand fully. It looks fairly simple but it can express very complicated thoughts. Among the things it can express are axioms for set theory, which can describe the most complicated structures we can imagine.

I conjecture that this is one of the reasons why quantifiers in spoken languages are so sensitive to context. Context-free language for them would be very hard to learn. Perhaps children would not completely acquire their native languages until they were teenagers. Yet many quantified expressions are essential for everyday communication. Instead, we allow quantifier expressions to take many meanings, and we trust ourselves to figure out what exactly is being communicated depending to a large part on the details of the situation surrounding the speaker and the hearer.

9:3 (of 8) two tasks for pronouns

Quantifiers can be used to join simple attributes and relations into complex ones. They allow us to construct much more complicated attributes and relations than Boolean connectives do. This is the main topic of the rest of this chapter. However, there is also another very important feature of quantifiers that is also connected with the variety of meanings that quantifiers can have in everyday language.

We do not need the word “only” if we have the word “all”. To express “only” with “all”, we need to use IF plus pronouns such as “he”, “she”, “it”. ("Only fools eat raw pufferfish" = "if it is a raw pufferfish and you eat it then you are a fool.") But the pronouns are doing a very special job, that is particularly important when the topic is quantifiers. For example we can explain the difference between "Mo likes all cats" and "Mo only likes cats" by rephrasing the first as "if it is a cat then Mo likes it" and the second as "if Mo likes it then it is a cat". To do this we need the pronoun "it". The pronoun is not just saving us breath, as it would in "Finny was a goldfish. It lived in a silver bowl", where we could replace it with a repeated use of the name: "Finny was a goldfish. Finny lived in a silver bowl." Instead, it is doing two things. It is representing the claim as general "whatever it is, if it is a cat ...". And it is carrying this generality from one part of the claim to another ".. then Mo likes it." This is clearest when we separate the two functions. "Consider anything. If it is a cat then Mo likes it." If there were only three individuals in our domain, a, b, c, this would amount to "if *a* is a cat then Mo likes *a* & if *b* is a cat then Mo likes *b* & if *c* is a cat then Mo likes *c*" . So the "it" is not substituting for any one name but referring to all the individuals in the domain.

9:4 (of 8) universal and existential quantifiers 1: simple attributes

Suppose we have a model with a domain of individuals D, to which an attribute **P** applies. Then we can define the *universal quantifier,* ∀ and the *existential quantifier* ∃ by

∀**x Px** is true in the model if and only if every individual in D has **P**.

∃**x Px** is true in the model if and only if at least one individual in D has **P**.

The same definitions apply when we use any other variable, so that ∀**y Py** , ∀**z Pz ,** and so on, are true when every member of D has **P**, that is, when the search **Find x: Px** gets every member of D, as do the searches **Find y: Py**, **Find z: Pz**, and so on. And ∃**y Py**, ∃**z Pz**, and so on, are true when one or more members of D have **P**, that is, when these searches do not fail by getting the empty set.

Another way of linking these two quantifiers with queries is to say that ∀**x Px** is true in a model when the search **Find x: Px** gets the same results as the search **Find x: Px v ~Px**, and ∃**x Px** is true in a model when there is something, **a**, in the domain of the model such that {**a**} is a subset of the results of **Find x: Px**. These are trivial variants on what I said earlier, but they prepare something that will be useful later. A mangled English version of them would be for ∀**x Px** "whatever you pick it will be among the Ps", and for ∃**x Px "**you can choose something so that it is among the **P**s", or "if you choose suitably it will be among the **P**s".

These definitions can be stated more precisely, but the best way to understand these quantifiers is to consider a number of examples. In the rest of this section we consider progressively more complicated examples. By the end of this chapter you will have a good-enough sense of what is and is not a sentence of quantifier logic. Rough rules go as follows: we take open sentences made with atomic attributes or relation symbols and then either join them with Boolean connectives, following the rules for propositional logic, or precede them with quantifiers. This allows some sentences that we don't really have much use for, such as ∃**x Aa** ("there is someone such that Albert is angry"), but then similar things happen in propositional logic, where we can write things like **p v p v p v p**. ("Smoking is forbidden, or smoking is forbidden, but then perhaps smoking is forbidden, or even smoking is forbidden". I had a teacher who used to say things like this. Strangely, he was a very good teacher.)

The following are examples of well-formed sentences of quantifier logic

∀x (Dx ⊃ Lxx) **∃x (Dx & Lxx)** **∀x ∃y (Cy & Hyx)**
∀x ((Dx & Lxa) ⊃ Lax) **∃x (Dx & ∀y ((Dy & Lyy) ⊃ Lxy))**

The following are examples of ill-formed (badly-formed) sentences

∀x &(Dx ⊃ Lxx) The ∀**x** is not before a well-formed sentence according to propositional logic

∃x (Dx)& Lxx) brackets problem

∀x (Dx & Lxa) ⊃ The whole sentence is not well-formed according to propositional logic, since there is no consequent for the conditional

And the following are well-formed sentences that we will have no use for, but it would be too much trouble to rule them out.

∀x Fx & Rxy **Pa & ∃a Pa** **∀x (Fx ⊃ ∃x Gx)**

>> do you think you can state, fully and completely, what is and is not a sentence of English, or any other spoken language?

This formal (artificial) language is what we will study as quantifier logic. All its sentences are used to make assertions about the world or describing particular models. But quantifiers can also be used to express questions, as in "are all the children in bed?" or to give commands, as in "put a few of the bigger cakes away for tomorrow". In particular, quantifiers can be used in search commands, just as Boolean connectives can be. We can say "find all the prisoners who have been arrested exactly 3 times", or "find all the numbers less than 100 that have three prime factors". As with Boolean connectives, this allows a greater range of helpful examples. So I shall sometimes mention searches involving quantifiers in what follows.

9:5 (of 8) examples

Begin to get a feel for ∀ and ∃ by considering sentences of logic using them one at a time with attributes (not relations in this section, except for some of the examples involving search) and relating them to natural and mangled English sentences. One purpose of the mangled English is to help you see the point of repeating the variable between the quantifier and the following criterion. Another purpose is to prevent you thinking that there is a mechanical way of translating between English and quantifier logic. In fact, you have to take each English sentence and ask "what is it really saying?" and then say the same in terms of logic. The mangled English is meant to help you do this. Which mangled English versions seem to you ok, and which ones seem weird and un-natural, may depend in part on your native language and how quantification is handled in it.

To give the examples a kind of unity, I will use predicates naming kinds of animals, in an obvious kind of code. In giving the examples I do not often refer explicitly to models with domains of whose individuals the attributes are true or false. You can supply these if you want.

∃x (Bx & Ax)
There is a bear that is an aardvark.
Some bear is an aardvark
There is something such that it is a bear and it is an aardvark
There is something, let's call it Xeno. Xeno is a bear and Xeno is an aardvark•
(Why do the bear and aardvark signs overlap? Because of that one in the middle cage. We call him Xeno. Xeno is a bear and Xeno is an aardvark.)
You can search for "both bear and aardvark" and find one

∃x Bx & ∃x Ax
Something is a bear and something is an aardvark
There are bears and there are aardvarks
You can search for bears and find one, and you can search for aardvarks and find one

The contrast between this last sentence and the previous one is important. The rewriting in terms of search makes it clear: in the one case you are told that one search for a conjunction will succeed, and in the second case you are told that a conjunction of two searches will succeed. If you then try to say this without talking of searches you will find you need some version of quantifiers plus variables.

>> describe a model — a domain with an attribute — in which
∃x (Bx & Ax) is true and **∃x Bx & ∃x Ax** is false. can you do it the other way round? why?

A simple example of a quantified query may also help here. We have a model consisting of three birds in a row. The only relation in the domain is **L**, **Lxy** holds when **x** is to the left of **y**. The query **Find x: ∃y Lxy & ∃y Lyx** gets one individual, the middle bird. But the query **Find x: ∃y (Lxy & Lyx)** gets nothing, since no bird is both to the right and to the left of any other.

∀x ~Ux
Everything is not a unicorn
Take anything, whatever **it** may be. **it** is not a unicorn•
We can say of anything: that thing is not a unicorn.

∀x (Bx ⊃ Mx)
Everything is such that if it is a bear then it is a mammal
All bears are mammals
For everything: if that thing is a bear then that thing is a mammal
Take anything. suppose it is a bear. then it is a mammal•
Bears are always mammals
What things are mammals? Bears, among others, all of them•
If it's a bear then it's always a mammal

>> describe this in terms of search

Note how we use the conditional ⊃ to say that all things of one kind are of another kind. The universal quantifier ∀ and the conditional ⊃ fit together very well to do this. **Bx ⊃ Mx** will be true unless **Bx** is true and **Mx** is false. So if we say that for everything in a model **Bx ⊃ Mx**, we are saying that for no individual **x** in the model **Bx** and not **Mx**, which is just to say that everything that is a **B** is an **M**. The last English paraphrase above, which uses both the pronoun and the adverb devices, is really a lot like the logic. It is just a matter of slight rearrangement:

If it's a bear then it's always a mammal
If it is a bear then it is a mammal, always
Always: if it is a bear then it is a mammal
Universally: if it is a bear then it is a mammal
Whatever it is: if it is a bear then it is a mammal
For all x: if x is a B then x is a M
∀**x (Bx ⊃ Mx)**

By itself this is quite straightforward, but it can seem surprising that “All As are Bs” is ∀**x (Ax ⊃ Bx)** while “Some As are Bs” is ∃**x (Ax & Bx)**. “All” needs a conditional where “some” needs a conjunction. There are two ways you can deal with this.

(a) You can just learn something that will often be useful, as it is absolutely basic in relating symbolic logic to English:

“All As are Bs”is ∀**x (Ax ⊃ Bx)** while “Some As are Bs” is
∃**x (Ax & Bx)** .

(b) Alternatively, you can see that this difference between “all” and “some” makes sense. When we say, for example, “all bears are mammals” we are saying “ignore everything that is not a bear, and everything you have left is a mammal”. And that is just what the ⊃ in the middle of ∀**x (Bx ⊃ Mx)** does: together with the ∀ it says “everything has the property that ignoring non-**B**s, when something is a **B** it is an **M**”. If you make a picture of a typical model in which all bears are mammals, this will be clear right away. Even without this way of putting it, it probably makes sense to you that all bears are mammals if and only if anything you choose is, if a bear, a mammal. On the other hand, when we say “some bear is a mammal” we are saying “you can find one thing — more are not needed — which is both a bear and a mammal.” And that is just what the **&** in the middle of ∃**x (Bx & Mx)** does: together with the ∃ it says “you can find something with the property that it is both a **B** and a **M**”. Even without this way of putting, it probably makes sense to you that some bears are mammals if and only if you can choose something that is a bear *and* also a mammal.

Note that although it is not very natural English, we could express the thought that some Albanians are geniuses as "sometimes she is Albanian and she is a genius", which parallels the sequence of versions of "if it's a bear then it is always a mammal" above. It would take a very particular context to make this the natural way to express the thought, though.

>> find such a context

An important consequence of thinking of “all” this way is that it makes, for example, “all the messages are spam” true if there are only two messages, and they are spam, though we might instead say in normal English “both messages are spam”. And if there is just one message, and it is spam, “all the messages are spam” will be true. In fact, if there are *no* messages then “all the messages are spam” will be true. For take any individual **i**. Since “**i** is a message” is false, given the meaning of “if” in logic, “if **i** is a message then **i** is spam” is true. So, to vary the example, in logic “all unicorns can fly” is true, since there are no unicorns. (More carefully: in any model that does not include unicorns, and this includes models whose domain is a subset of the real world and where there are attributes corresponding to the English words “unicorn” and “fly”, the logic version of “all unicorns fly” is true.)

>> this is one of several ways in which "all" and "some" are unusual quantifiers. "Most cats are black" is not true in exactly the same models as "for most individuals x: if x is a cat then x is black". do you see why? (see exercises 13 and 14 of ch 11.) does this show that we should not be misled by the special features of these quantifiers, or that we should value them for being better behaved than others?

>> why might the cautious formulation in the parenthesis at the end of the paragraph above be needed?

>> most non-logicians would not think that "all unicorns can fly" is true just because

there are no unicorns. might this be because most people do not understand IF as ⊃ or because they understand ALL differently (or both)? (or because they are confused, perhaps.)

Now some more examples. In each case I give the symbolic logic and then a series of paraphrases into various kinds of English.

∃x (Bx & ~(Sx v Ax))
Some bears are not seals or aardvarks
There is something that is a bear and it is not the case that it is a seal or it is an aardvark
It can be found: true that **it** is a bear. false that **it** is a seal or **it** is an aardvark ▪

∃x (Bx & ~Sx & ~Ax)
Some bears are neither seals nor aardvarks
There is something that is a bear and is not a seal and is not an aardvark
You can find at least **one** individual: that **one** is a bear and that **one** is not a seal and that **one** is not an aardvark ▪
It can happen that a bear is neither a seal nor an aardvark

∃x (Px & Mx)
Some penguins are mammals
There is something that is a penguin and is a mammal
It's a penguin and it's a mammal. yes, you can find such things

(This example is here to contrast with the next one. But note how in the 'it' version the part expressing existence, "you can find.." comes at the end, while in logic it comes at the beginning.)

>> why is the sentence easier to understand this way?

>> don't learn your biology from logic books!

∃x Px & ∃x Mx
Something is a penguin and something is a mammal
There is something that is a penguin and there is something that is a mammal
NOTE that this sentence is different from the previous sentence. It is true in the real world, and the previous one is false. The difference is one of scope: in this sentence the quantifiers are within the scope of the conjunction, and in the previous one the

conjunction is within the scope of the quantifier.

>> find a similar example using the scope of OR rather than AND

∀x Px ⊃ ∀x Mx
If everything is a penguin then everything is a mammal
If whatever you choose, it's a penguin, then whatever you choose, it's a mammal.
Suppose that everything is a penguin. then everything is a mammal ▪

This sentence is true in the real world. The reason is the definition of **⊃.** Since ∀**x Px** is false, the conditional is true. Contrast this true sentence with the following false one. The difference between them is another scope distinction.

∀x (Px ⊃ Mx)
Everything that is a penguin is a mammal
All penguins are mammals
Choose anything: if it turns out to be a penguin then it will turn out to be a mammal
This is universally true: if it's a penguin then it's a mammal

6 (of 8) relations and variables

When we quantify relations we have to be careful that our use of variables does not make unintended claims. For example, if we want to say that there is something that **R**espects something, we should not write it in logic as ∃**x**∃**x Rxx**, because that would seem to say that there is something, represented by the variable **x**, that respects itself. We want to leave it open whether thing giving respect is the same as the thing receiving respect. So what we should write is ∃**x**∃**y Rxy**. It is important to see that this does not require that the two things are different. There are models for ∃**x**∃**y Rxy** where one thing respects itself. But ∃**x**∃**y Rxy** leaves the question open.

We have the same problem in everyday language. If we say "something respects something" it could at least suggest that something respects itself. We sometimes deal with this by saying "something respects something else". But this has the opposite problem: it seems to suggest that the two are distinct, when we simply want to leave the issue open. We can say "something respects something, which may or may not be the same thing". But that is pretty cumbersome. Another thing we can say is "there is something that respects and something that it has respect for". There are in fact many ways, many of them using abstract nouns such as "respect" and "the bearer of respect" as distinguished from "the object of respect".

We can stick with something a bit like spoken language and use something analogous to the distinct variables that would do the job in logic. Suppose we want to make clear what is going on with "Mo is looking at twins: one is a boy and one is a girl". We cannot say "Mo is looking at something: it is twins and it is a boy and it is a girl". That is absurd. We have to say something like "There are two things. The first is a boy and the second is a girl and the first is a twin of the second." Along these lines, we could say

> Consider anything-one, any thing-two. If one and two are twins and one is a boy and two is a girl then Mo is looking at one and Mo is looking at two.

And we can make variations on this such as these:

> Consider anything-one, anything-two. If one is an ant and two is an ant and one smells two then one follows two ▪
>
> Consider any pair of numbers, number_a, number_b . if number_a is bigger than number_b then number_b is smaller than number_a ▪
>
> In the yard there is person one, person two. One has betrayed two ▪

(When lawyers are writing a contract and they want to be really clear, sometimes instead of writing "she" or "he" they write "the party of the first part", "the party of the second part", and so on.)

\>> why is the version with subscripts better when the domain is numbers?

>> “take any three people: if all pairs of them are friends, then one of them will hate one of them.” how would you say this without using the words “one” , “two” or "three" or "pair"?

Consider the *ant* sentence above. Suppose that an ant smells itself (perhaps it has walked in a circle and come on its own trail.) Then, to repeat the main point of this section, it is important to see that saying that when an ant smells an ant it follows it, does not mean that when an ant smells *another* ant it follows it. If the sentence is true then when an ant smells itself it follows itself. With variables as we use them in logic there is no assumption that different variables have to apply to different things. So in a model in which there is only one individual, and a relation **R** which that one individual bears to itself, all three of the following sentences are true.

∃x Rxx there is something that has R to itself
∃x ∃y Rxy there is a first individual and a second individual, where the first has R to the second
∃y ∃x Ryx something has R to something

>> when might "everyone hates anyone who remembers their crimes" be true, but "everyone hates anyone else who remembers their crimes" be false?

>> is there some way it might be better if different variables could never refer to the same thing?

Sentences of logic will have to allow many quantifiers and many variables, even though beyond a point they become too complicated for us to understand. Exercise 14 at the end of this chapter discusses how to state the rules for a logical language that has infinitely many variables. Sentences beginning with two existentials or two universals — ∃**x** ∃**y** or ∀**x** ∀**y**— are not hard to understand, though, since ∃**x** ∃**y** means simply “there is a pair of individuals” and ∀**x** ∀**y** means simply “for all pairs of individuals.”

>> why might we want to have sentences that are so complicated no one can understand them?

>> are there English sentences which are too complicated for anyone to understand?

9:7 (of 8) accumulating details

Often a complex English sentence is made by adding details to a simpler one. Then we can make a version of it in logic by adding the same details. This often gives a procedure that you can follow to represent the thought expressed by an English sentence with one in quantifier logic. The idea is to begin with a very simple core, such as this.

> **$\forall$x (Bx ⊃ Dx)**
> All bears are dangerous
> Bears are invariably dangerous
> What's a bear, is dangerous

Then we add qualifications to the attributes. In simpler cases we can take one attribute as central, dangerousness for example, and then we can ask "what is dangerous? Bears, which bears? Bears when they are pregnant." And then we get the following.

> **$\forall$x ((Bx & Px) ⊃ Dx)**
> All bears that are pregnant are dangerous
> All pregnant bears are dangerous
> When it is a bear and it is pregnant then it is dangerous
> Take any**thing**: if **that thing** is a bear and pregnant then **that thing** is dangerous
> Consider a pregnant bear: they are always dangerous
>
> >> we could also say "all pregnant individuals that are bears are dangerous". this is rather un-natural English. why?

Note that when we add the qualifying conditions we have to insert some brackets, to keep it well-formed and unambiguous. This is like adding commas or pauses in natural language.

We can also take a different attribute as central. We could ask "what kinds of things are bears? Well, they're dangerous omnivores."

∀x (Bx ⊃ (Dx & Ox))
All bears are dangerous omnivores
If it's a bear then, whatever else, it's a dangerous omnivore

And we can do the same when the quantifier is existential rather than universal. We can start with "There is a bear/bears exist/something is a bear", which would be done in logic as "**∃x Bx**", and qualify it via a what-question, to get "There's a bear: what kind? A pregnant grizzly".

∃x (Bx & Px & Gx)
There is something that is a bear and pregnant and a grizzly
A bear, also pregnant and of the grizzly kind: such things exist

∀x ((Bx & ~Mx) ⊃ Ax)
All bears that are not mammals are aardvarks
Take anything: if it is a bear and not a mammal then it is an ardvark.
Whatever **it** is. suppose **it** is a bear and **it** is not a mammal. then **it** is an aardvark ▪

(This one is, perhaps surprisingly, true, in the real world. For since **Bx & ~Mx** is false for all **x**, the conditional is always true. See the remark above about "all unicorns can fly".)

∃x ((Sx &Mx) & ~Bx)
Some seals that are mammals are not bears
There is something that is a seal and mammal and it is not a bear.
You can find **one**, where **that one** is a seal and a mammal, and **it** is not a bear.

There is more practice with English and logic along these lines in the exercises at the end of the chapter.

The "which ones?" strategy applies also to sentences with relations. We can start with core constructions like "something terrifies something" Or "everything terrifies everything". Which are easily rendered as

∃x ∃y Txy and **∀x∀y Txy**

But these are so general that we are unlikely to use them. We can get more specific by asking "what" questions: *what terrifies something? something terrifies what? what terrifies everything? everything terrifies what? what terrifies what?* Then we can fill in the details with answers to these questions, as follows, giving suitable meanings to the attribute and relation symbols.

∃x ∃y (Px & My & Rxy)
a pig terrifies a man
something that is a pig terrifies something that is a man
I'm discussing two things. the first is a pig and the second is a man and the first terrifies the second ▪

∀x ∀y ((Px & My) ⊃ Txy)
pigs terrify men
take a pair of individuals. if the first is a pig and the second is a man then the first terrifies the second ▪
all pigs terrify all men

Note how we treat universal and existential differently when we make the content more specific: ⊃ versus **&**. This is just like what we saw above, discussing the analogous process with attributes rather than relations.

We can specify more by further qualifying "pig" or "man" ("speckled pig", "man with a beard") but more details emerge if we specify in terms of another relation.

>> just to be sure you are following, how would you represent in logic "all men with beards terrify all speckled pigs"?

∀x ∀y ((Px & ∀z (Cz ⊃ Fxz)) ⊃ Txy)
Pigs that fear carrots terrify all dogs
Consider any pair where the first is a pig and the second is a dog. Suppose moreover that the first is frightened by something whenever that something is a carrot. Then the first terrifies the second ▪
One thing will terrify another under certain conditions. this will be true whenever the first is a pig and all carrots frighten it, and the second is a dog

Note that the last of these mangled English sentences is again expressed in the opposite order to the logic sentence, with the central relation at the beginning rather than the end. This does fit with the idea of taking a simple sentence and progressively qualifying it. (So the **T**errifies relation in these examples is slightly analogous to the central connective in a purely Boolean sentence. You can only push the analogy so far, though.) This can be seen as a two-dimensional branching construction. Let me show how, for the English and the quantificational logic side-by-side.

the English

a thing terrifies a thing

what? what?

anything that is some kind of Pig anything that is a Dog

what kind of pig?

pigs that fear carrots

So: all pigs that fear carrots terrify all dogs

the logic

$\forall$**x** $\forall$**y** **Txy**

| **Px** **& Dy** **⊃ Txy**

| **Px & ∀z (Cz ⊃ Fxz)** **& Dy** **⊃ Txy**

So $\forall$**x** $\forall$**y** **((Px & ∀z (Cz ⊃ Fxz)) & Dy) ⊃ Txy)**

I suggested earlier in this book that thoughts were not best seen as having a linear structure, and these considerations might be taken as another, far from conclusive, reason for thinking this. You will notice that in the English it is the attributes and relations that we follow to get from the core sentence to the elaborations, while in the logic version it is the quantified variables (the **x** and **y**.) The English pattern does have definite disadvantages, though, which emerge in the next chapter.

A similar contrast between the strategy for putting something into logic and the corresponding strategy in a natural language such as English also applies when we are understanding rather than formulating expressions in either form. In natural language we typically first grasp what is being spoken about and then add detail about what is being said of it. In logic, on the other hand, we typically first grasp what the logical form of the assertion is, and then add detail about what is being described with an assertion of this form. It may not be until the very end of the process that we understand what individuals are being discussed. The result is a less immediate understanding of the topic but a more immediate understanding of what kind of thing is being said of it.

Mathematical forms of expression and everyday language often contrast in roughly this way. Even in arithmetic we would understand $(\sqrt{636}/\sqrt{318})^4$ as the fourth power of a quotient of square roots, before we figured out the values of these expressions. This would often be a more efficient way of proceeding; in this example we would then be able to see this as the square of 636/318 which is 4. If you calculated the roots by hand to start with you would be calculating all day, and even if you use a calculator you would be more likely to make a mistake. This applies with greater force in more advanced mathematics, so that for example $dx^2 sinx/dx$ is understood first as the derivative of a product, so that the product rule can be applied. Only then are the components of the product evaluated. You would not attempt to think of $x^2 sinx$ as a single function. (Ignore this second example if it means nothing to you.)

The point is just that in mathematics we typically look first at the overall form and fill in the details later. Mathematically sophisticated people do this automatically, but there is a potentially helpful halfway between the styles where we first give the form without the details and only when that has sunk in provide the details. My advice is to read anything mathematical in a series of successive passes, first absorbing the general form and then honing in successively on more and more detail. This contrasts with the way we usually understand sentences of spoken languages, where we tried to make sense of everything as soon as we encounter it, and tends to halt in confusion if we cannot do so immediately. (The difference between sound and sight may be relevant here: most mathematics is written.)

This is connected with the contrast I made in chapter 5 between inside-out and outside-in ways of understanding a sentence. Mathematics is more often outside-in and spoken language is more often inside-out. It also connects with the advice early in this book to make some forms of thinking automatic, though under the control of conscious reasoning. (See chapter 2 section 3 and exercise 14, and chapter 5 section 6.) The connection is that when we understand the general form of an assertion we can see what particular kind of automatic thinking we should do. The general form is consciously understood and the details are thought through automatically.

>> have you noticed that there are some people who focus exclusively on the topics you are mentioning, and not on what you are saying about these topics? you say "I do not think your behaviour was wrong" and all they hear is "think – behaviour – wrong" and then get upset. this is an extreme form of inside-out language processing.

9:8 (of 8) extra: English to logic

Given a sentence of quantifier logic, and a context in which it is used, one can construct an English sentence that says roughly the same. Or a sentence of almost any other natural language. This is not to say there are easy rules for doing this. You have to think what the sentence means, which models it would make it true, in the given context. The same is true in the other direction, English-to-logic. In this chapter I have concentrated on helping you to think in terms of quantifiers, so that you can see what quantified logic sentences mean and then be able to reproduce them in English. But there is also the opposite task, of saying in logic terms what an English sentence means. I cannot give rules for translating between English and quantifier logic. (And if I can't, no one can!) I am sure there are no rules that do not have many exceptions. For one thing the meaning of an English sentence depends much more on the context in which it is produced — when and where it was said, by whom and to whom against what background assumptions — than a sentence of logic. Still, if you have understood the chapter so far, helped by some exercises at the end of the chapter, you should be able to represent in logic terms most of the English sentences that you see involving "all", "some", "each", "every", "there is", and similar words. This last section of the chapter describes some strange irregular features of quantification in English. It is not meant to make translation easier for you as much as to persuade you that any firm rules for doing it must be fiendishly complicated, and to make you appreciate the relative tidiness of logic. (And the relative subtlety and expressiveness of natural language.)

A complication here is that there are usually several different sentences in symbolic logic which are equally good as renditions of a given English sentence in a given context. Very often, these are logically equivalent, true in the same models. But we do not discuss logical consequence and equivalence for sentences with quantifiers until chapter eleven. So sometimes the version that seems right to you will be different from the version in this text or which is mentioned in class, but just as good. The only way of dealing with this is to speak up and ask a question!

>> which of the following seem intuitively equivalent?

no cats purr	all cats do not purr
cats are purring animals	cats purr
cats chase mice	there is a cat that chases all mice
every cat chases every mouse	every cat chases some mouse

Begin with sentences in which no quantifiers are mentioned, such as "Amy likes cats", or "Cats kill birds". These can mean many things. "Amy likes cats" can mean "take any cat: Amy likes it" — $\forall \mathbf{x}\ (\mathbf{Lax} \supset \mathbf{Cx})$ — and it can also be used to express the thought "take anything: If Amy likes it then it's a cat" — $\forall$**x (Lax ⊃ Cx)** — whether or not it can literally mean this. (Suppose you are comparing the animal tastes of three people. George only likes dogs, Mo adores frogs, and Amy does not share either of their tastes, but she likes cats.) Very often "Amy likes cats" will mean something that cannot be expressed exactly with "all", such as "Amy likes many cats" or "there have been cats that Amy has liked". "Cats kill birds" can mean "take any cat: it kills some birds". And it could mean "take any cat and any bird: the cat will kill the bird" (that would take a very special context: we mustn't pair kittens and eagles). It often means "cats typically kill birds", where "typically" says something much more subtle than adverbs of universal quantification such as "invariably" or "always". (This is what linguists call a "generic" sense of the sentence.) Sometimes, too, a bare plural can have an existential rather than a universal meaning. Contrasting with the "all" meanings of "Amy likes cats" there is the "some" meaning of "Amy was feeding cats". This means roughly that there were cats and Amy was feeding them.

>> say "Amy likes cats", meaning these different things, and see if your stress pattern and intonation vary. can you draw any general conclusions?

I said that "Amy likes cats" can be used to communicate that Amy only likes cats. But "only" has its own complications. It is unlikely that in saying "Amy only likes cats" we are denying that Amy likes food, or friendship, or breathing. Most likely we mean that among animals cats are her favourites. What about "Amy only likes black cats." Here are three things it can mean, even restricting ourselves to literal general "all"-type meanings.

if Amy likes it, it's a black cat $\forall$**x (Lax ⊃ (Cx & Bx))**

if it's a cat and Amy likes it, then it's black $\forall$**x ((Lax& Cx) ⊃ Bx)** if it's black and Amy likes it, then it's a cat **∀x ((Lax & Bx) ⊃ Cx)**

(The last of these requires the most context or to be said with just the right stress. Suppose we know that Amy generally goes for bright colours: she wears purple clothes and has a yellow car. But there is an exception, she thinks of herself as a witch and black cats as her familiars: she only likes black *cats*.)

>> describe situations in which any one of these would be true and the other two false

We can say similar things about "Amy likes all black cats": she likes *all* black cats, or all *black* cats, or all black *cats*.

>> write these three out in logical symbols

Then there are the differences between "all", "each" and "every", and between "some", "there is", "there are", and "exists". Some of these are very subtle and hard to capture in terms of standard symbolic logic. To complicate things, many of these words can have both universal and existential meanings. The most ambiguous word that can indicate a quantifier is the indefinite article "a", as already mentioned. In "if a dog is thirsty, it will bark" or "a barking dog is dangerous", the "a" represents a universal quantifier. "If a **D**og is **T**hirsty, it will bark" has the form ∀**x ((Dx & Tx) ⊃ Bx**) and "a barking dog is d**A**ngerous" has the form ∀**x ((Dx & Bx) ⊃ Ax)**. But in "a barking dog is standing in the yard", the "a" indicates an existential quantifier. The sentence has the form ∃**x (Dx & Bx & Yx)**. Why is this? It may be just the effect of what we expect to be true. Suppose that someone did not know anything about dogs but is looking for a barking dog. Then that person might understood the sentence "a barking dog stands in the yard" as information about barking dogs in general: you can find them because they stand in the yard.

To show how bad it can get, though "some" normally indicates an existential quantifier, there are sentences where it can be taken as universal. Consider for example "if some stranger comes to the door, don't let them in". That is clearly $\forall$**x ((Sx & Dx) ⊃ ~Lx)**. Consider also "if there are honest politicians, they are Canadian" which seems to have the form $\forall$**x ((Hx & Px) ⊃ Cx**). In both of these, the influence of the "if" seems to be to lure "some" or "there are" from their usual existential meanings to universal ones. (This may be linked to the rules for "prenex form" discussed in chapter ten.) Quantifiers that one might think of as universal can be used with an existential sense, too. In "I doubt that any cat has ever written a haiku" the "any" has the sense of "some".

These are the exceptions: most English quantified sentences are better behaved, so these are puzzles that will not often affect us. But they do underline one of the main points of this chapter, that it is better to understand quantifier logic in its own terms rather than by translating from a natural language.

words in this chapter that it would be a good idea to understand: inside-out, outside-in, pronoun, quantifier, variable.

exercises for chapter nine

A – core

1) In the **F**ears model

Fears	**a**lice	**b**o	**c**arlos	**d**eepa
alice	YES	YES	YES	YES
bo	YES	NO	YES	NO
carlos	YES	YES	YES	YES
deepa	YES	YES	NO	YES

(a) i) **Find x: Fxx**

ii) **Find x:** ∃**y (Fxy & ~~Fyx)**

iii) **Find x:** ∀**y (Fxy** ⊃ **~Fyx)**

(b) in each case describe in English the property being searched for.

(c) which of these is true in the model?

i) ∀**x Fxx**

ii) ∃**x Fxx**

iii) ∀**x** ∃**y (Fxy & ~Fyx)**

iv) ∃**x** ∃**y (Fxy &~Fyx)**

v) ∀**x** ∀**y (Fxy** ⊃ **~Fyx)**

vi) ∃**x** ∀**y (Fxy** ⊃ **~Fyx)**

(iii and vi go beyond what is discussed in this chapter, so take them as a challenge.)

2) Which of these English sentences mean roughly the same? (some of them do not have such clear meanings)

a) there is a cat that wears pyjamas

b) there are cats that wear pyjamas

c) if it's a cat, it wears pyjamas

d) alice only loves cats in pyjamas

e) all pyjama-wearing cats are loved by alice

f) there are some pyjama-wearing cats

g) every cat wears pyjamas

h) alice loves cats if they are wearing pyjamas

i) all cats wear pyjamas

j) if cats are wearing pyjamas, alice loves them

k) cats wear pyjamas

l) if it's a cat and wears pyjamas then alice loves it

m) alice loves cats in pyjamas

n) if alice loves it then it's a cat and wears pyjamas

o) if alice loves it and it wears pyjamas then it's a cat

p) alice loves cats only when they are wearing pyjamas

q) alice only loves pyjama-wearing cats

r) some cats wear pyjamas

3) Which English sentence corresponds to each of these sentences of quantifier logic?

(a) ∀**x ((Cx & Hx)** ⊃ **Fx)**
(b) ∀**x ((Fx & Hx)** ⊃ **Cx)**
(c) ∃**x (Cx & Fx)**
(d) ∀**x ((Hx & Cx)** ⊃ **Fx)**
(e) ∀**x Cx &** ∀**x Hx**
(f) ∀**x (Cx** ⊃ **Hx)**
(g) ∃**x (Cx & Hx & Tx)**
(h) ∃**x (Cx & Hx) &** ∃**x (Cx & Tx)**
(j) ∀**x Cx** ⊃ ∀**x Hx**

(i) all cats are hungry

(ii) everything is a cat and everything is hungry

(iii) if everything is a cat then everything is hungry

(iv) some cats are both hungry and thirsty

(v) some cats are hungry and some cats are thirsty

(vi) all hungry cats are fierce

(vii) all fierce and hungry things are cats

(viii) some hungry cats are fierce

(ix) there are fierce cats

4) Which sentence of quantifier logic corresponds to each of these English sentences? (**Dx** = x is a dog, **Lxy** = x loves y , **a** = Alice)

first group

Everything loves itself
All dogs love themselves
All dogs love Alice
Alice loves all dogs

i) ∀**x (Dx ⊃ Lax)**
ii) ∀**x (Dx ⊃ Lxa)**
iii) ∀**x Lxx**
iv) ∀**x (Dx ⊃ Lxx)**

second group

Some dog loves all dogs who love themselves
Alice loves all dogs who love her
Alice loves all dogs who love some dog

v) ∃**x (Dx &** ∀**y (Dy & Lyy) ⊃ Lxy)**
vi) ∀**x ((Dx &** ∃**y Lxy) ⊃ Lax)**
vii) ∀**x (Dx & Lxa) ⊃ Lax)**
viii) ∃**x (Dx &** ∀**y ((Dy & Lyy) ⊃ Lyx))**

5)

	Prudent	**Q**uarrelsome	**R**ealistic
ai	YES	YES	NO
bertram	NO	YES	YES
chiu	NO	NO	NO
destry	YES	NO	NO
elspeth	NO	YES	YES

Which of the following are true?

a) ∀**x (Px ⊃ Rx)** b) ∀**x (Px ⊃ ~Rx)** c) ∃**x (Px & Qx)**

d) ∃**x (Px & Rx)** e) ∀**x ((~Rx & Qx) ⊃ Px)** f) ∀**x ((Px & Qx & Rx) ⊃ ~Px)**

g) ∀**x ((Px & Qx) ⊃ Rx)** h) ∀**x (Px & Qx) ⊃** ∀**x Rx)**

6) You probably were first aware of variables when you met them in algebra class in school. (Though you used them in the form of pronouns expressing quantifiers long before, as explained in chapter nine.) Usually when we use a variable in a mathematical expression there is an unstated quantifier. Rephrase the following and supply quantifiers, so that the result is true.

$ax^2 + bx + c = 0$ has two real or imaginary roots

the solution to $3x - 12 = 0$ is $x = 4$

Newton's law of gravitation giving force F for masses m_1, m_2 separated by distance r is $F = km_1m_2/r^2$

B – more

7) For each of the sentences in quantifier logic below write the letter of the English sentence that best captures its meaning. **Cx** = x is Crazy , **Bx** = x is Boring, **l** = Lee , **m** = Mo ,**Rxy** = x is happier than y

~(Cl v Bl)	**∃x (Cx & Bx)**	**∀x (Cx ⊃ Bx)**
∀x (Cx ⊃ ~Bx)	**∀x (Bx ⊃ ~Cx)**	**∃x ∀y Hxy**
∀x ∃y Hxy	**∀x ∃y Hyx**	**∀x (Hxm ⊃ Cx)**
∀x (Cx ⊃ Hxm)	**∀x ∃y (Cy & Hyx)**	**∀x ∃y (Cy & Hxy)**

(a) Crazy people are not boring
(b) Someone is both crazy and boring
(c) Lee is neither crazy nor boring
(d) Crazy people are boring
(e) No-one boring is crazy
(f) For everyone there is someone happier than them.
(g) Some person is happier than everyone.

(h) Everyone is happier than someone.
(i) Everyone is happier than some crazy person.
(j) All crazy people are happier than Mo.
(k) Given any person there is a crazy person happier than them.
(l) Everyone happier than Mo is crazy

8) The map below shows nine locations, where three individuals are found. We know that all individuals are found at one of these nine locations. They are related by 'x is to the **N**orth of y' (that is, due north and further north) and 'x is to the **West** of y' (that is, due West and further west). One individual, **w**ally, satisfies the following conditions,

∃x Nwx
∃x Nxw
~∃x Wxw

Where's Wally? And where are the other two?

North

?	?	?
?	?	?
?	?	?

C – harder

9) On the map below three **S**hips are marked ship 1,2, 3, and three **I**cebergs are marked iceberg 1,2, 3. Correlate the logic and the English versions of the following claims

<u>logic</u>

∀x (Ix ⊃ ∃y (Sy & Nxy)

∀x ∀y (Sx & Iy & ~∃z ((Nxz & Nzy) v (Wxz & Wzy)) ⊃ ~∃w (Nwx v Nwy)

∀x (∃y (Sx & Iy & Wxy) ⊃ ∃z (Sz & Wzx))

∀x (Sx ⊃ ∃y (Iy & Nxy)

<u>English</u>

Every ship is to the north of an iceberg

Every iceberg is to the north of a ship

If a ship is west of an iceberg then it is east of a ship

All the ships and icebergs that are next to one another are as far north as one can go.

North

ship 1	**ship 2**	**iceberg 1**
ship 3		
iceberg 2	**iceberg 3**	

Which are true, which false

10) In the appendix to chapter five I gave eight rules to define the well-formed formulas (sentences, propositions) of propositional logic. Modify rule one to read

$\mathbf{R_{nm}}$ is a relation symbol for each integer n and m.

(This gives us infinitely many n-place relations $\mathbf{R_{nm}}$, for each n.)

$\mathbf{v_m}$ is a variable for every integer m. (This gives us infinitely many variables. We can abbreviate $\mathbf{v_1, v_2, v_3, v_4, v_5, v_6}$ as **x**, **y**, **z**, **u**, **v**, **w**.)

If s is a variable or a string of variables then sm is a string of variables, for every integer m.

This gives us infinitely many variables $\mathbf{v_1}$, $\mathbf{v_2}$, …. (We can abbreviate $\mathbf{v_1}$, $\mathbf{v_2}$,$\mathbf{v_3}$ as **x**, **y**, **z**.)

If R is a relation symbol and s is a string of variables then Rs is a well-formed formula. This gives us infinitely many atomic propositions $\mathbf{R_{nm}}$ **v1,..,.vn**, where **v1,..,.vn** is a string of variables.

"What other rules need to be added, to define the well-formed formulas of quantifier logic?

chapter ten: multiple quantification

This chapter discusses sentences with several quantifiers. There is even more of a contrast with spoken languages here than in the one quantifier sentences of the previous chapter, though you cannot appreciate quite how incomplete and ambiguous a language such as English is in this respect until you can express the meanings that it misses or runs together. The topic is too big for this chapter, though, or indeed for this book. In this chapter I focus on strings of quantifiers in a row: to give a fore-taste, the sort of thing we find in "every student takes some course that at all later times they remember with delight" or "there is a mountain on which all members of the team trained in some season". This is a source of much of the richness and power of symbolic logic. In the next chapter, the last, we will partially connect this wealth with the other main source of logic's power, the ability to join closed and open sentences with Boolean connectives.

10: 1 (of 10) scope distinctions: negation and quantifier order

What did Abraham Lincoln mean when he said "you can fool some of the people all of the time"? It could be that there are people who will always be fooled, or it could be that at any moment we can find people who are fooled (but it may be different people at different times.) This is a scope ambiguity with "some" and "all", like the ambiguities involving OR and AND.

>> do you think that people who cite the Abraham Lincoln saying know which interpretation they mean?

The simplest cases involve just one quantifier and negation. "John is not rich or happy" is confusing because it could be heard as meaning "John is not rich OR John is happy" or "it is not the case that John is rich or happy". (And the second of these is equivalent to "John is not rich AND John is not happy", you'll remember: de Morgan's laws.) The same holds for the universal and existential quantifiers. This is not surprising given the analogy between the universal quantifier, ∀, and AND, and the existential quantifier, ∃, and OR. Suppose someone says "all of my term papers are not stolen". (Perhaps someone has accused him or her of stealing them off the internet.) What does the person mean? Here are two possibilities.

(a) **~∀x (Tx ⊃ Sx)** The following is false. consider anything. if it is a term paper of mine it is stolen ▪

(b) ∀**x (Tx ⊃ ~Sx)** Consider anything. if it is a term paper of mine it is false that it is stolen ▪

These are different. (a) says just that not all of the papers are stolen, while (b) says that all of them are not stolen. So if exactly half are stolen (a) is true, though (b) is false.

Both are possible meanings in different conversations. (a) "You stole all your papers!" "No, all of my term papers are not stolen – just some." (b) "I think some of your papers were stolen." "No all of them are not stolen." The ambiguity can be resolved in English by rephrasing. "Not all my term papers are stolen" is clearly (a), and "Every one of my term papers is not stolen" is clearly (b). (So one reason we have "each" and "every", besides "all", is to allow us ways to clarify these matters. But the differences between these in English are subtle and complicated.) The fact remains that there are English sentences that can easily be taken as being like (a) or like (b). And this sets up a tendency in us to slide in reasoning from "not all" to "all not".

>> what is the difference between "most of my cats do not have fleas", and "it is not the case that most of my cats have fleas"? describe a situation where one is true and the other false

We use the all/every distinction in everyday English to help with another scope distinction also, to signal which of a universal and an existential quantifier is within the scope of the other. Consider the difference between "all the plants were covered with a plastic sheet" and "every plant was covered with a plastic sheet". The first of these can most easily be understood as saying that there is one plastic sheet that covers all the plants, while the second can most easily be understood as saying that each plant is covered by its own plastic sheet. So the "all" version says $\exists$**x** $\forall$**y (Sx & (Py ⊃ Cxy)).** SOME ALL: there is a sheet and if it's a plant then that sheet covers it. The "each" version says

$\forall$**y** $\exists$**x (Py ⊃ (Py & Cxy))**. ALL SOME: take any plant: you can find a sheet that covers it.

>> do you see how the routine for using "if" with "all" and "and" with "some" applies here?

Consider again the Abraham Lincoln quotation: "You can fool all of the people some of the time". It might mean that there are occasions when you can fool everyone, and it might mean that for each person there are times when you can fool that particular person. The first is SOME ALL — there is one or more times such that for all people — and the other is ALL SOME — for all people there is one or more times. Suppose there are only three potential fools concerned, and that on Mondays, Tuesdays, and Wednesdays you can fool Alice but not Bob or Carol, on Thursdays and Fridays you can fool Bob, but not Alice or Carol, and on Saturdays and Sundays you can fool Carol, but not Alice or Bob. Then it is true that for each person there is a time at which you can fool them, but not true that there is a time at which you can fool all the people.

>> the Lincoln quotation continues "And you can fool some of the people all of the time,

but you cannot fool all of the people all of the time." again we have a some/all sentence, but it is different from either of the two we've just seen. state the two meanings, and make a days of the week model in which one of them is true and the other false, and in which *both* the the interpretations of "you can fool all of the people some of the time" are false.

These are simple sentences but quite confusing. There are several reasons that they get one's mind in a twist. One of them is that there are four possibilities, and when we understand these sentences we tend to contrast them with others that might have been said, but the alternatives sound similar. Here are the four possibilities: in each case I will state the meaning in mangled (improvised, approximate) English designed to make it intuitively clear which ones are consequences of which others. The four:

a) **∀t ∃x Fxt** every time has its sucker

at every time this is true: someone gets fooled

b) **∃x ∀t Fxt** there's some constant victim

there's a person such that this is true: they get fooled at all times

c) **∃t ∀x Fxt** there's a time when everyone's a victim

there's a time such that this is true: everyone gets fooled

d) **∀x ∃t Fxt** everyone has their vulnerable moment

for every person this is true: they are sometimes fooled

Think about these until it is clear to you that they correspond to one another. This should help get it fixed in your mind how to understand quantifiers in the scope of other quantifiers. If you state these in terms of 'some' and 'all' it is very easy to slide from one to another. But in fact a) is a logical consequence of b), d) is a logical consequence of c), and all the others are independent of one another. We will not be able to show this in a careful way until the next chapter, but the versions in brown should make it intuitively clear. If (b) is true then someone, call him Mr Victim, is always getting fooled, so then, (a), at every time there is someone, him, who is fooled. And if (c) there's a moment, say Wednesday, when everyone is fooled, then (d) everyone is fooled then. But (a) can be true when (b) and (c) are false, if every time has a different sucker and there are more people than times, when c) and (d) are also false. Similarly (d) can be true when (c) is false, if everyone is fooled at a different time and there are more times than people, when (b) and (a) are also false.

>> how can (b) be true and (c) and (d) be false? how can (c) be true and (b) and (a) be false?

>> "but his name might not be Mr Victim", "the day might not be Wednesday". why are these irrelevant worries?

10:2 (of 10) quantifier words pretending to be names

I have used (a) to (c) make the scopes clear. But this is not the case for the original Lincoln sentences, or for many other ways of presenting quantifiers in English. The reason is that one central way that English, and many other languages, make quantified assertions is to put a quantifier word ("someone/everyone", "something/everything", "some cats/all cats", "a few cats", "most cats" and so on) in the same places in a sentence where we might find a name such as "Maggie". For example just as we say "Maggie is sneaky" we might say "Someone is sneaky", "Every cat is sneaky", "Most cats are sneaky", and so on. This has its puzzling side: who is this "Someone" who is sneaky as long as even one (other?) person is sneaky? If it is true that someone is sneaky and someone is not sneaky, is Someone then both sneaky and not sneaky?

> >> why would it make *No-one* even more of a puzzle than *Someone* if we thought quantifiers were a kind of name?

One could refuse to be puzzled and continue to speak this way. But the trouble gets deeper when we have two quantifiers in the same sentence, and then it gives another reason why we often find them confusing. Consider a simple sentence with two different quantifiers, such as "everyone was angry with some person". Does that mean "we can take any person and we can find someone who they were angry with", or "there is some person who everyone was angry with"? Either way we can tell a story so that it is natural to understand it in that way.

>> tell the stories

We can make it clear which quantifier is in the scope of which, in perfectly ordinary English. We can say "each person was angry with someone or other", "everyone had someone they were angry with" and so on, to be explicit that the existential is within the scope of the universal. And we can say "there was some particular person who everyone was angry with", or "the same person was the object of everyone's anger", to be explicit that the universal is within the scope of the existential. But the fact remains that for many English sentences we cannot tell from the grammar of the sentence which quantifier has the widest scope, and must rely on what we know about who produced it, who was the audience, and what is most likely to be taken as true. We cannot assume that the quantifier that comes first is meant to have the widest scope. For example compare the two sentences (sandwich) "someone stole every sandwich" and (death) "someone died every minute". They have very parallel structures, and it is hard to see how anything about their grammar will give "someone" and "every" different scopes in the two sentences. But the natural way of understanding (sandwich) gives "someone" the widest scope — there is a person who stole every sandwich — and the natural way of understanding (death) gives "every" the widest scope — each moment was one where some particular person died. (The natural reading of (sandwich) is even more dominant if we rephrase it as "some thief stole every sandwich".) And the reason is obvious: it is easy to see how there could be a thief who took all the sandwiches, but not easy to see how there could be a person who died and then died again a minute later. So we tend to choose the interpretation that is most likely to be true.

>> find a story in which it makes sense to understand "some thief stole every sandwich" so that "every" has widest scope.

>> find a story in which it makes sense to understand "someone died every minute" so that "someone" has widest scope.

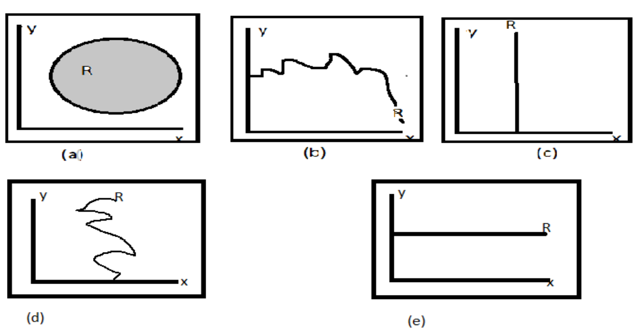
The same ambiguity and the same sensitivity to context is found with other natural language ways of expressing quantifiers. "Someone is always guarding the store" can mean "at any time there is someone who is the person guarding the store at that time", or "there is a person who at all times guards the store", depending on the conversation or story it is part of. "Sometimes if you guard the store you fall asleep" can mean "there are times when anyone guarding the store falls asleep" or "anyone guarding the store will at certain times fall asleep". The fact is that natural languages rely on context as much as on grammar to make clear the scope of quantifiers, and that one basic reason for this is that many quantifiers are treated like names.

>> is this a problem with natural (spoken) language? is anything wrong with relying on context to settle ambiguity about the scope of quantifiers ?

>> make stories in which these two ambiguous sentences are pushed towards one or the other interpretation.

10:3 (of 10) a graphical representation of quantifier order

Quantifier order can be illustrated graphically. If we number the individuals in the domain (or if they *are* numbers) a two-place relation **R** corresponds to an area in the plane, the set of points (x,y) such that **Rx,y**, as illustrated in (a) below. When ∀**x**∃**y Rxy** is true there is a point vertically above every point on the x-axis, where **Rxy**, as illustrated in (b). When ∃**x**∀**y Rxy** is true there is a point on the x-axis from which a vertical line extends all the way up, as illustrated in (c). When ∀**y** ∃**x Rxy** is true there is a point horizontally across from ever point on the y-axis, as illustrated in (d). And when ∃**x**∀**y Rxy** is true there is a point on the y-axis from which a horizontal line extends all the way out, as illustrated in (e). There are many ways of distributing points which make ∀**x**∃**y Rxy** or ∃**x**∀**y Rxy** true, but I have chosen (b) and (c) so that ∃**x**∀**y Rxy** is *not* true in (b) and ∀**x**∃**y Rxy** is *not* true in (c). And I have chosen (d) and (e) so that ∀**y** ∃**x Rxy** is not true in (d) and ∀**y** ∃**x Rxy** is not true in (e). You will see that ∀**x**∃**y Rxy** is true in (c), as it is in any model for ∃**x**∀**y Rxy**. Any graphical depiction of ∀**x**∃**y Rxy** will have some sort of horizontal barrier all the way across, ad any graphical depiction of ∃**x**∀**y Rxy** will have a vertical line extending upwards from some point on the X axis, but they may have other points as well: these are the core models, pared down to contain only what they need to make the corresponding sentences true.



These graphs may remind you of something from chapter 1. In a relational grid we have a row of YESs when there is an individual that has the relation to every individual, and we have a column of YESs when there is an individual that every individual has the relation to. When every individual has the relation to some individual there is a YES somewhere on every row, and when some individual has the relation to every individual there is a YES somewhere on every column. (And we have a column of YESs when there is an individual that has the relation to every individual, also a YES somewhere on every column when every individual has the relation to some individual.) These are really the same as the diagrams above, given the convention of making the horizontal axis the x-axis, except that a relational grid can only be used to show a small finite number of individuals.

Notice that (b) and (d), and (c) and (e), are flipped versions of one another. Exchanging two variables is the same as flipping the graph around a diagonal. This may help with a point in the next chapter.

4 (of 10) why these ambiguities matter

Suppose we have a sentence whose grammar is ambiguous between two readings but which is much more likely to be true if understood one way than the other. Isn't that interpretation then the obvious one to use?

Often, it is. But there are situations where it is not. Here are four of them.

conjecture Sometimes it is clear that something is true, but a less obvious thing might also be true. So we want to be able to say "yes, sure, but here is a more subtle possibility." For example, most likely time had a beginning, so that for every time there is an earlier one: but it is also possible that time goes back forever, so that for all times there is an earlier one. We want to be able to state these so that it is clear that they are different.

saying the unlikely In a similar way, sometimes we want to make a claim which will surprise people, and we don't want them to say "oh yes, everyone knows that". For example, we might want to say "There is a cause for all diseases", and mean not the unsurprising "every disease has a cause" but the bold and implausible "there is a cause which is at the origin of all diseases." Someone claiming this would have to choose their words carefully so that it was clear what a radical suggestion it was.

interpretations It is often clear which way of understanding a sentence is more likely to be true, when the words mean what they usually do. But often they do not. An extreme example is when words are used with very different meanings to their normal ones. So we might use names of animals as names for sports teams (the cougars, the bears) and refer to the results of sports contexts in exaggerated terms (the cougars were lucky to escape with their lives, the ducks destroyed the bears.) Then your expectations of what is true are completely unreliable, and given a claim such as "all bears can take care of a cougar" you do not know which way to understand it.

complicated argument When we consider subtle arguments in philosophy or complex proofs in mathematics we are often operating at the limits of our understanding. So instead of relying on a real grasp of what we are talking about we often hold onto the bare words. But this makes us victims of a kind of "bait and switch" trick, where we allow an assumption because it seems harmless and then If we are not careful it is used in a much less harmless form. For example we might begin by assuming that every physical system can be described mechanically and later reason as if we had assumed that the universe is one big machine. Or we might start by thinking that people can be mistaken about any particular item of information and go on as if this was the same as thinking that we can be mistaken in everything we think. These are both examples of ∀/∃ scope reversal.

>> why are they?

A historically important example from mathematics is the assumptions that are needed to make sense of differentiation and integration in calculus. These were at first confusing and inconsistent, and to state them clearly mathematicians had to give clear definitions of limit, derivative, integral, and various kinds of continuity. Care about quantifier scope was crucial in doing this, and it is surely no coincidence that only after this had made the topic important did logicians come up with adequate treatments of quantifiers[22]. Another example is axiomatic set theory, where the axioms have to be stated with great care, which the precision of symbolic logic, particularly with respect to the order of quantifiers, makes possible.[23]

What to conclude from all this? Just that natural language works fine as long as each little utterance is surrounded by a sea of context, people speaking to one another share a lot of assumptions about the world around them, and their aim is to communicate definite truths rather than conjectures, possibilities, or interesting ideas. Take these away, and language needs a lot of propping up. We should not hold this against it, but we

22 A nice discussion of these issues is in David M. Bressoud *A Radical Approach to Real Analysis*, The Mathematical Association of America 2007.

23 An exception is David Lewis's *Parts of Classes* (Wiley-Blackwell 1991), a sophisticated book on set theory which uses no symbols. A marvel of clear writing.

should learn how to do the propping up on those occasions when it is needed. We can do a lot of the propping from within our usual language, but a little outside help and, most importantly, an awareness of what the dangers are, is always useful.

10:5 (of 10) invisible quantifiers

Many sentences have several quantifiers. We tend not to realise how many because we hide them in various ways. (Perhaps we do this because if we spot them we think about them, and then we get confused.) Here are some ways quantifiers can hide.

words with quantified meanings A person is a parent if they are a parent of *someone*, a daughter if they are the daughter of *someone*, and so on. A person is a president if there is *some* organization of which they are president. But we would not say, for example "if there is someone of whom a given person is a parent then in very few cases is there an organization of which that person is president." Instead we will make the words swallow the quantifiers and say "parents are rarely presidents" .

Most of these words have absorbed existential quantifiers. But there are a few cases in which the quantifier is universal. A view is panoramic if one can see in *all* directions, a wrench is universal if it can be used to turn *all* bolts. A person is promiscuous if they have sex with *many* other people.

>> find other examples

passives A thing is broken if *something* has broken it; a bottle is opened if *something* has opened it; a person has been stabbed if *someone* has stabbed them. Passives reverse the order of a relation: if x loves y then y is loved by x. But they also serve to avoid mentioning a quantifier. We say "x is loved" to mean "*someone* loves x". (We also say "x loves" to mean "x loves someone". "I have loved and been loved" is "I have loved someone and someone has loved me".) English and some related languages have passive versions of verbs; many languages do not (although some have anti-passives, which quantify the object rather than the subject.)

>> find other examples

tense In English we say that, for example, someone has eaten all the cherries, to communicate that there is a time which is past and that at that time someone at all the cherries. This has the form

∃**x (P**erson**x &** ∀**y (C**herry**y** ⊃ ∃**t** (**T**ime**t & P**ast**t & E**ats**xyt)))**

The future tense is similar, except that for **P**ast we have **F**uture.

There are more complicated tenses. We might say "**a**lice ate an **A**pple and by then she had **E**aten a **P**ickle", which has the form, if we write it as a single sentence of quantifier logic,

∃**x** ∃**y** ∃**s** ∃**t (Ax & Py & T**ime**s & T**ime**t & P**ast**t & B**efore**st & Exys & Exyt)**

It is easier to understand if we write it as a series of English semi-sentences:

> We are discussing two times, both in the past. At the second Alice eats an apple and at the first she eats a pickle. The second is later than the first.

The "had" tense (the past perfect, language teachers will say) asserts that *there exists* a past time at which something happened, and *there exists* another past time, at which something else happened, and the first of these is earlier than the second. This looks really complicated, written out like this, so you can see why language might choose to hide the quantifiers with a device such as the tense of a verb.

>> why do the time quantifiers usually come within the scope of other quantifiers? (hard question)

>> if you speak a language that does not have these ways of hiding quantifiers, does it have different ones?

There are other ways of hiding quantifiers. Every language has its own ways of keeping things simple. I suspect that these devices, hidden in the meanings of words, in passive constructions, in tenses, and elsewhere, have their own rules for making the scopes of quantifiers clearer. But these rules are not obvious. Consider tenses for example. "Every student argued with the teacher" is ambiguous: it could mean (a) there is a past time t such that for every student s, s argues with the teacher at t, or (b) for every student s there is a past time t such that s argues with the teacher at t. To express (b) we might be more likely to say "every student has argued with the teacher" which suggests that one function of having both simple past tenses ("argued") and perfect tenses ("has argued") may be, instead of or as well as what language teachers will tell you, to clarify the scopes of the quantifiers that are hidden in tenses.

>> I wrote "student s quarrels with the teacher at time t", using a present tense. I had to use some tense or other, to make a good English sentence. is this a problem for the view that tenses are quantifiers over times?

Another example concerns adverbs of quantification. Consider "all the hummingbirds were occasionally at the feeder". This can be understood as (a) for every hummingbird h there were times t such that h was at the feeder at t, or (b) there were times t when for every hummingbird h, h was at the feeder at t. To my ear (b) is the only meaning of "Occasionally all the hummingbirds were at the feeder", and (a) is the only meaning of "Each hummingbird was on occasion at the feeder".

Or consider "some elephants will invariably fear mice". It can mean (a) there is at least one particular elephant which is frightened of any mouse, or (b) for any mouse there are elephants which are afraid of it. I suspect that (a) is the more natural meaning and that a very particular context will be needed to push the sentence towards (b). (But think of this story: I am telling you about the elephants in a particular herd, and about the mice that live nearby, and I say "Most of the elephants will be ok with the mice but I'm sure that there are one or two elephants that will be spooked: some elephants will invariably fear (these) mice.")

The ambiguities are even found with words which have an implicit existential quantifier. "All the men are uncles" can mean (a) for any man m there is a person p such that m is uncle of p". But it can also mean (b) there is a person p such that all the men are uncles of p. (You ask why a group of men at a wedding are having their photograph taken together and you are told "they are all uncles", meaning that they are all uncles of the bride or groom, and it is a rare occasion when they meet.)

There may be subtle rules which make it likely that a sentence involving hidden quantifiers together with the choice of traditional quantifier word ('each', 'every', 'some', 'there is, and so on) will tend to one scope order rather than another. That is, these sentences though sensitive to context and expectation, may be less sensitive, not so easily moved from one meaning to another, given these subtle rules. But I know that, though I have been speaking English fluently every day for many decades, I could not state these rules to save my life.

10:6 (of 10) from two to three

We often think and communicate thoughts which would contain three, four, or more quantifiers if we analyzed them. These are often hidden, in ways that I have discussed, and when stated explicitly they can be very ambiguous and confusing. Some of language's devices for making the order of the quantifiers clear do not apply well, and the range of possible meanings can overwhelm us. Logical symbols avoid these problems, but many people are simply baffled by them when they get beyond a certain complexity. The purpose of this section is to get you to understand sentences with three and four quantifiers. A string of quantifiers at the beginning of a sentence of quantifier logic is called the *quantifier prefix*, and the simple or complex relation between the variables is called the *matrix*. We will only discuss the simplest case, in which the quantifiers come all together first, followed by a matrix which is a simple relation between the variables. (But already this is different from the spirit of ordinary language, where in inside-out fashion we state very few quantifiers at the beginning of a sentence and bury the rest within it.)

As I have remarked in earlier chapters, it is often helpful to understand mathematical notation in an outside-in way. Then we can process them in stages, leaving the inside content till later while focusing on the outside content. For an example that has nothing to do with formal logic consider the summation operator. (I once had a very intelligent colleague who said he could understand all mathematics up to the level where ∑ enters. I expect he is typical of many people, and variable binding operators prefixed to functions are a kind of barrier, requiring an approach that is typicallymathematical.) Expressions like $\sum_{0 \leq n \leq 100} 3n^6+5$ — don't worry: it is explained below — can be hard to digest if you try to understand them as if they were unified combinations of symbols whose meanings had be put together all at once. A better way is to think of this along the following lines. "We are adding up a series of numbers. Each member of the series is a function of the variable n, and we will add them up beginning when n is 0 and stopping when n is 100. The function in question is $3n^6+5$." A single expression has been unpacked into a series of linked sentences, each of which can be understood in the way that we ordinarily understand language. Notice that the formula, when we understand it this way, shows us how to do a mechanical calculation of its value, without thinking at every stage what we are doing.

>> how does it show how to do a mechanical calculation?

We can do the same with complex quantifier sentences. (This is not surprising, since they are a form of variable-binding operator, and the analogy with summations and integrals must have been in the minds of their inventors.) Begin with the very simplest cases, the two quantifier sentences we have already seen. Consider for example ∀**x** ∃**y** **Bx** **D**etests **xy** . To explain it by separating the quantifier prefix and the matrix into separate sentences that were as natural English we could say

Everyone has one. Someone they detest, that is.•

And in the same vein we could explain the contrasting sentence ∃**x** ∀**y** **Bxy** as

There is a person who has this relation to everyone. That is, he or she detests them•

These make the contrast between the sentences clear. But the English is less natural-feeling in other more complicated examples. So here is a variety of similar English versions. First for ∀**x** ∃**y** **Bx** **D**etests **xy** .

Pick any person. That determines another person. The first person detests the second•

Everyone has their un-favourite. They detest that person•

Take any person. They have someone who they hate•

For every person there is a second person. That person detests the second person•

For any first person there will be some second person. The first will detest the second•

Next for ∃**x** ∀**y Bxy.**

There is a particular person. They detest everyone•

There is a hate-filled person. They detest everyone•

There is a first-person who has an attitude to everyone. That attitude is detestation•

Now that you have seen these you can invent many variants for yourself. They show very clearly the difference that the order of the quantifiers makes. You may think that they are just clumsy ways of saying things that can be expressed by simpler and more natural English sentences. You would be right. But they introduce means of expression that can make the meanings of sentences with many quantifiers easy to understand. So I am introducing them in cases where you already know the meanings. As the number of quantifiers increases it will be increasingly useful to be able to break a sentence of quantifier logic into a sequence of sentences that can be understood in an outside-in way.

All of the versions I have listed have to find some substitute for a useful device in spoken language: we use names of kinds of things to mark the difference between variables. So we can say "every cat fears some dog" or "there is a cat that fears all dogs" and we can stretch these out as "for every cat there is a dog. The cat fears the dog" and "There is a particular cat. She fears all dogs." When we are talking about arbitrary individuals, arbitrary people, or whatever is in the domain of some model, we do not have this resource.

Now consider sentences with three quantifiers. They are easiest to understand when they relate three different kinds of things, so I shall use the resource I just mentioned.

Consider an English sentence such as "every skier skied at one of the resorts on some day". Think of the meanings it can have. (Suppose we are discussing two skiers, two resorts, and three days of some week.) Some of these meanings can be given in stretched out form as follows.

> For each person there was a place and a time when the person did it. They skied at that resort on that day•
>
> On one particular day and one particular resort everyone did it – skiing•
>
> At one particular resort everyone did it on one day or other. That is, they skied•
>
> On one particular day everyone did it somewhere or other. What they did was skiing•

>> make stories ending with "every skier skied at one of the resorts on some day" that make the sentence have each one of these meanings

>> choose two of these sentences and describe a situation where one is true and the other false, then a situation where the second is true and the first false

In logical symbols, we can abbreviate these as follows, using **Ssrt** to mean that skier **s** skied at resort **r** at time **t**. (They're in the same order and colour as the quasi-English versions.)

> **∀s ∃r ∃t Ssrt**
>
> **∃r ∃t ∀s Ssrt**
>
> **∃r ∀s ∃t Ssrt**

∃t ∀s ∃r Ssrt

It is important to see that these are all different. They are got by putting the universal "skier" quantifier in different positions with respect to the two existential "resort" and "time" quantifiers. So you should think of models where some are true and others false. A model cannot make the second one— **∃r ∃t ∀s Ssrt** — true without also making the first one — **∀s ∃r ∃t Ssrt** — true, but all other combinations of truth and falsity are possible.

> so which ones are logical consequences of which others? we have not given any rules for this, yet, so the question is just what seems intuitively to follow from each of them

I have made these easier to read by writing **∀s** for "all skiers" and **∃r** for "there is a resort", and so on. If we are discussing a model which has skiers, resorts, times, and other things (cats, professors, prime numbers) in one domain, then we will have to say that it is a skier that fills the first place of the relation. So for example the first of the four sentences above would become

∀x ∃y ∃z (Sx ⊃ (Ry & Dy & Sxyz))

(for anything we can find two others, so that if it is a skier then they are a resort and a day when the skier skis there) or some variant using different letters.

There are more possibilities, if we combine the quantifiers and the skiers, resorts, and times more freely. We could say "Some skier skied at all the resorts at some times", which would itself have a range of possible meanings. (Compare the four meanings in the Abraham Lincoln example in section 1.) And there are others. See exercises 6, 11, 12 for more practice with 3-quantifier sentences.

> state the meanings of "Some skier skied at all the resorts at some times" in less ambiguous words and in logical symbols

10:7 (of 10) picturing models with many-place relations

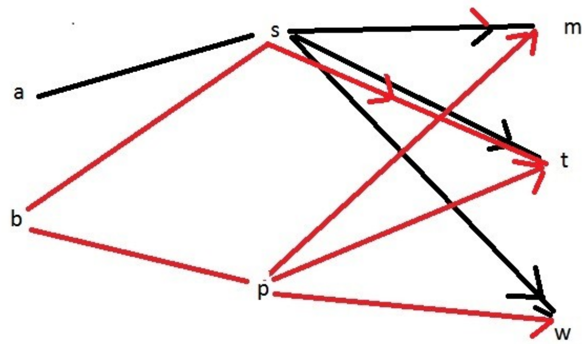
So far we have discussed the cases of 2 and 3 quantifier prefixes where the matrix has no quantifiers, and often consists of a single relation. More than three explicit quantifiers in a single sentence are rare in spoken language, but there are often more hidden quantifiers that would increase the number if they were made explicit. ("Some deserts arrived, but all the uncles and grandmothers had been disgusted and so they weren't eaten." 6!) My aim is to give you an intellectual tool that will allow you to think in terms of longer quantifier prefixes, expressing very subtle thoughts. At the end of this chapter I discuss examples with six explicit quantifiers, to show that they are really not so terrifying. Before we get to this, we need a way of describing models for 3-place and 4-place relations, so we can discuss the logical connections among 3 and 4 quantifier sentences.

Consider the skiers-resorts-days example again. One way of picturing the kind of situation these sentences discuss is a modified arrow diagram of the three-place relation "s skis at r on day d". Here, for just two skiers, two resorts, and three days, is a model of one possibility:

skiers	resorts	days
alice, **b**o	**s**noparadise	Mon, Tues,
	powderheaven	Wed

black arrows: where and when Alice skis

red arrows: where and when Bo skis



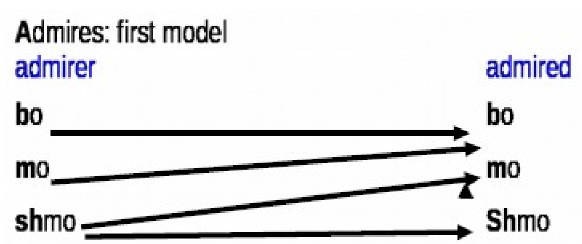
The diagram shows that Alice skis at Snoparadise on Monday, Tuesday and Wednesday, and Bo skis at Snoparadise on Monday, and Bo also skis at Powderheaven on Monday, Tuesday, and Wednesday. The diagram would get messy and hard to read if there were more skiers, resorts, or days. I doubt that there is a way of making diagrams for relations with three or more places that does not involve some serious compromises.

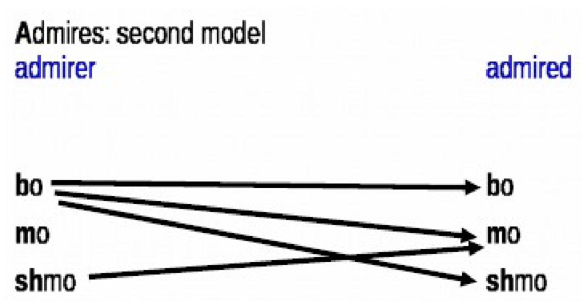
>> which of the skier sentences above are true in this model?

>> how would you rewrite the model presented this way as a database for a relation, as discussed in chapter one?

>> if diagrams for many-place relations are so tricky, might we do better just to express them in spoken language?

We can make diagrams like this for two-place relations also, and they are useful in seeing the difference that quantifier order makes. Here are presentations of models for the relation "admires" between two people.





Note that in these diagrams the same individuals are named in both columns. In the first model ∀**x** ∃**y** **Axy** is true and ∃**x** ∀**y** **Axy** is false, and in the second model ∀**x** ∃**y** **Axy** is false and ∃**x** ∀**y** **Axy** is true. (Be sure that you see why this is so.) Because of diagrams like these, relations such that ∀**x** ∃**y** **Axy** is true are sometimes called ladder relations, and relations such that ∃**x** ∀**y** **Axy** are sometimes called wheel relations. (Notice that there can be extra rungs and spokes, besides the ones that picture a ladder or a wheel, so that the words are just tricks for remembering the two kinds of relations.)

>> what are the truth-values of ∀**y** ∃**x** **Axy** and ∃**y** ∀**x** **Axy** in these two models?

>> what are the advantages and disadvantages of representing a model for a two-place relation this way, as opposed to arrow diagrams?

There are other ways of presenting models for relations with three or more places. Some are used in the exercises at the end of this chapter. It is good to be able to make and understand a variety of diagrams since, to repeat, making a suitable diagram is often the key to solving a practical logic problem.

10:8 (of 10) 4-place relations

Diagrams for relations can suggest and help us understand language that expresses the scope relations between quantifiers. There are two aspects to this. First, we see a relation as a kind of chain: it goes from one argument place to another, to another, depending on how many argument places it has. For example the skis relation goes from skiers to resorts to days. Second, we see that sometimes where a relation goes from one place depends on where it has come from. For example in "for every skier there is somewhere that she skis on some day" the place and time that a particular skier skis depends on which skier you choose. And in "for any skier and resort there is a day that the skier skis there" the day that a particular skier skis at a particular resort depends on which skier and which resort you choose. There are many ways of doing this using the resources of a language such as English. We can break long complete sentences down into linked series of less complete sentences, as we've seen several times in this chapter and previous ones. And we can use prepositions such as "for" to indicate the function of a quantifier. These can combine the prefix and the matrix into separate chunks, which can be understood independently. We can handle very large prefixes and matrices in these terms.

When we have four variables we can have two existential and two universal quantifiers so it is possible to alternate twice, ∀∃∀∃ or ∃∀∃∀, This is more interesting, potentially more expressive, because with three two will be of one kind and one of the other, and for example ∀∀∃ is not so different from ∀∃, since it can be taken as saying "for all pairs of individuals there is an individual such that ..". ("For every happy couple there is an apparent friend who would like to see them separate.") Consider these four-quantifier sentences. Be aware of the fact that the order of the variables in the quantifier prefix, or of the individuals referred to in the statement of the chain, is often not the same as the order in the matrix or the relational statement. They serve different functions: the prefix says which choice of which variables depends on which and the matrix says what individuals are linked by the relation. It is a basic advantage of quantifier logic over everyday language that it keeps these separate.

<u>sentence I</u>

$\forall$**v** $\exists$**p** $\forall$**m** $\exists$**c Pvpmc**

From which villages are there paths leading up which mountains from where you can see which **c**hurches? Start from any village and it will have a path that goes from the village up all of the mountains, and then, depending on which village and which mountain it is, one or another church will be visible•

<u>sentence II</u>

$\forall$**v** $\exists$**m** $\forall$**p** $\exists$**c Pvpmc**

Talking about villages with paths to mountains where you can see churches: start with any village. in terms of that village there will be a mountain, and whatever path we take will lead from that village up that mountain to where some church will be in sight•

<u>sentence III</u>

$\forall$**p** $\exists$**v** $\forall$**c** $\exists$**m Pvpmc**

Villages and paths from them lead up mountains to see churches: Consider a path, any path, and it will lead from some village up a mountain. Choose a church and given any one there is a mountain from which, whatever church you've chosen, it can be seen•

Convince yourself that all the English-ish sentences have different meanings, and that their differences are not just the different choices of words and sentence-structure. Then see how each fits the quantifier prefix of its formal version. And to check all this, think which of them is true in which of the models given by the following diagrams for “from village v path p leads up mountain m from where church c is visible”.

Diagram A

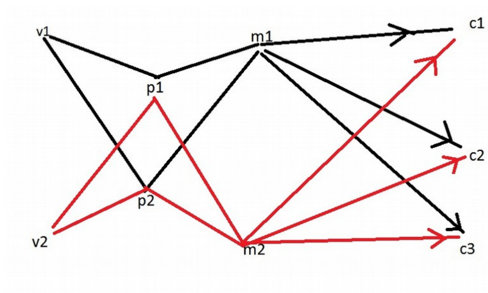


Diagram B

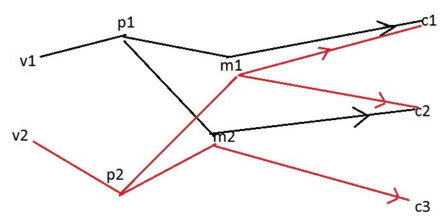
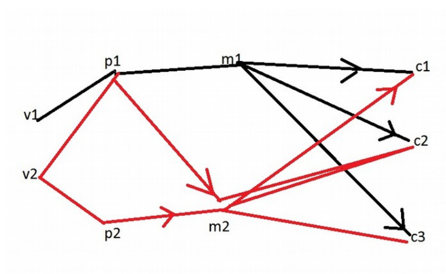


Diagram C



.

So which is true in which? Please think about this for yourself before reading below. Some hints:

> In sentence I the four quantifiers are village-path-mountain-church, which is the same order as in the matrix. So we want to look for a model in which — starting from the left — every village is connected to a path which is connected to some

mountain which is connected to every church.

In sentence II the order of the variables in the quantifier prefix is different to the order in the matrix. It says that whatever village you choose you can find a mountain to which every path from that village leads and goes on to some church.

In sentence III the order is again different. The prefix has the same universal-existential-universal-existential order as in sentence II but in this case it is path-village-church-mountain. So the sentence says that starting with any path you can find a village that it goes from which leads via some mountain to all churches.

The diagrams can have extra links besides the ones that are necessary to make them represent a model for a sentence. In particular, when we have a universal quantifier followed by an existential quantifier all that is required is that everything of the first kind be connected to something of the second, but in the model everything of the first kind may be connected to more than one thing of the second kind. ("Everyone has a friend" just requires that for each person there is someone who likes them, but it is still true if for each person there are several who like them.)

Now think hard about which sentence is true in which model. To encourage you to think rather than just look out for the answers I shall now include some further examples before the answers which are way way down.

Two people can work together to frustrate the plans of a third. Take any of the people we're discussing and there will be a third corresponding to him or her. Then taking anyone at all as the second person we find that the first of these and this second frustrate the plans of that third•

This is clearly of the form ∀**x** ∃**z** ∀**y** **Fxyz** rather than ∀**x** ∃**y** ∀**z** **Fxyz** .

Let's discuss cases where the number of people working in four buildings forms a geometrical series. This will be relevant to designing a disaster evacuation plan for the neighbourhood. In fact if we choose suitably we can find one which is the second in such a series whatever we choose as the first, where the third and fourth are chosen in terms of them•.

This is clearly of the form ∃**y** ∀**x** ∃**z** ∃**w** ∃**z** **Sxyzw** rather than ∀**x** ∃**y** ∃**z** ∃**w** **Sxyzw** .

>> How in these two last examples could we express the 'rather than' formulas in English-ish?

An attempt to make friends with someone can be thwarted by other people's quarrels. In this instance whoever we consider, their quarrel with someone had the effect of thwarting all of some person's overtures to people they would like to be friends with.

This is clearly of the form ∀**z** ∃**w** ∃**x** ∃**z** ∀**y** **Axyzw** , taking the relation to be "x's attempt to make friends with y is thwarted by z's quarrel with w', rather than

∀**x** ∃**y** ∃**z** ∃**w** **Sxyzw** . Contrast this with

> An attempt to make friends with someone can be thwarted by other people's quarrels. One particular person's attempt to make friends with another is blocked by the fact that everyone has someone who is quarrelling with them.

This is clearly of the form ∃**x** ∀**y** ∀**z** ∃**w** **Axyzw** , rather than

∃**x** ∀**y** ∃**w** ∀**z** **Axyzw** or ∃**z** ∃**w** ∀**x** ∃**y** ∃**z** ∃**w** **Sxyzw** .

> \>\> How might we express these two alternatives in English-ish?

It is now time to confess that I lied and the answers are not way way below but right here. Sentence I is true in the model represented by diagram B, sentence II in A, and III in A B, and C.

10:9 (of 10) inside-out

To repeat, the hard cases are when the variables in the prefix are in a different order to those in the matrix, when written in logical notation. These are hard to give in clear English. I have been producing "outside-in" English which mimics the way these are handled in logic. But there are also "inside out" ways of saying the same things. I will briefly give some examples. I use a three-place relation because the English passive voice can play less of a role. The relation Gxyz is "x gives y to z"; so there is a giver, a present, and a receiver. Consider the sentence of logic ∀**x** ∀**y** ∃**z** **Gzyx**. This does not say that everyone gave everyone a present, or that there is a present that everyone gave to everyone, as by now you understand. In terms of givers and presents it says that for every present and receiver there is a giver who gave it to the receiver. (So, incidentally, we see how having a rich and redundant vocabulary — *giver*, *receiver*, *present*, as well as *gives* — allows more ways to express yourself clearly.) But let us try to express this with just the relation G and quantifier words. Here are some possibilities.

> Someone — but which person depends on what they give and who they give it to — gives each present to each person.
>
> Some person, but we cannot tell until we know what is given and who it is given to, gives any present you choose to any person you choose.
>
> Someone, depending on the present and the receiver, gives each present to each receiver.

The last of these is probably the clearest, but it uses the vocabulary of giver/present/receiver to get the effect of reordering the relation. The moral is that we can if we really want put quantifiers in the places where names go, but if we do we also need ways of showing which ones depend on which others. This will often mean referring forward to something that has not been mentioned yet.

But isn't it simpler to say this, in outside-in style?

> Consider any person and any thing: someone gives it to her.

>> how would you say "everyone gave everyone a present" and "there is a present that everyone gave to everyone" in this inside-out style?

10:10 (of 10) complicating the matrix

In the examples so far the matrix has been a single relation, to focus on the quantifier prefix. But the matrix can be complicated too. Think about the meanings of the following examples, which use the two-place relational symbols **I** and **A** so that **Ixy** when **x** is **I**mpressed with **y**, and **Axy** when **x** **A**dmires y, and the three-place **R** so that **Rxyz** when

x Refuses to help **y** in bullying **z**.

$\exists x \forall y (Axy \supset Ayy)$

One person, m, has this relation to everyone: when she admires a person then that person admires herself

Admiration works this way for some person: when she admires anyone then that one admires herself•

There is someone who admires people only when they admire themselves

$\exists x \forall y \forall z (Axy \supset {\sim}Ayz)$

Admirers and the people admired by them: for one person, and any two further people if the first admires one of the pair then the second does not admire the second of the pair•

There's someone who admires people only when they don't admire anyone.

>> if this sentence is true then anyone the person admires does not admire them, in fact admires no-one. Why?

>>what is the difference between this sentence and "everyone admires only people who do not admire anyone?

$\forall x \exists y \forall z (Axy \supset Rxzy)$

Here's one connection between admiration and refusal to take part in bullying. Take any person and in terms of them there is another so that given any third, if the first admires the second then she will refuse to take part in that third person's bullying of the second•

Admiration and not taking part in persecution: everyone has someone who, if they admired them, they would refuse to take part whoever was bullying them•

Everyone has someone who if they admired them they would refuse to take part in anyone's bullying of them

>> this can be true *without* its being true that for everyone there is someone who's persecution by anyone they will refuse to take part in. how?

∀x ∃y ∃z (Rxzy & Ayx & ~Ayz)

Refusing to go along with bullying can lead to admiration. Every person generates two further people, a bully and a victim. She refuses to join the bully in persecuting the victim; the bully admires her; the bully does not admire the victim.

For every person there are two more: she refuses to join the first of these in bullying the second, is admired by the first who does not admire the second•

>> "two *further* people", "two *more*": these phrases are potentially misleading. why? how could we avoid them?

To end this chapter and to convince you that you can now understand things you could not before taking this course, here are a couple of sentences with six quantifiers.

Whichever door you take there will be a hallway leading to a bank of elevators, every one of which goes to a floor where every hallway has a dragon that is the same colour as the door.

Doors lead to hallways lead to elevators up to floors with dragons, but you have to take the right ones. Start with any door and then a suitable choice of hallway is possible, so you can continue with any elevator and make a suitable choice of floor to leave it; then take any hallway and finish by choosing a dragon on that hallway. This sequence of door, hallway, elevator, floor, dragon fits the condition that you can follow it from door to same-coloured dragon.

>> how could this sentence be true without it being true that any route from door to dragon gives one of the same colour? describe a situation where this sentence might be true but "any sequence of door, hallway, elevator, floor, hallway, dragon leads to a door-dragon colour match" is false.

This sentence was of the form "∀**x** ∃**y** ∀**z** ∃**m** ∀**n** ∃**o** **Axyzmno**". Six quantifiers, though they are best thought of as three ∀∃ pairs. The six variables are connected by the relation A, which could be analysed as a long conjunction

Dx & Hy & Hm & Ez & Fw & Do & Lxy & Lyz & Lzm & Lno & Lno & Sxo

where the one place attributes are "**D**oor", "**H**allway", "**E**levator", "**F**loor",and "d**R**agon", the two-pace relation **L** is "leads to", and the two-place relation **S** is "has the same colour as". The meaning of the sentence should be clear by this point, and it should be clear that the meaning is different from variants, such as this:

∀**x** ∃**y** ∃**z** ∃**m** ∀**n** ∃**o** **Axyzmno**

Start with any door, and then make a suitable choice of hallway, elevator, and floor so that any choice of hallway will then lead to a dragon of the same colour as the door.

More subtly, both of these are different from

∀**x** ∀**z** ∃**y** ∃**m** ∀**n** ∃**o** **Axyzmno**

Start with any door, and go to any elevator by a suitable choice of hallway, then there is a floor where for every hallway there is a dragon of the same colour as the door.

>> how do these versions separate the quantifier prefix from the matrix? (To what extent do they allow you to think of them separately?)

The difference is that in the sentence we are considering now there is a suitable choice of hallway for every combination of door and elevator, while with the previous one there is a hallway that is part of a path to a dragon from any door and any elevator it leads to. (For example suppose that several hallways leads from any door to several elevators. The previous sentence requires only that one of these hallways lead to elevators all of which get us to a dragon. But this second sentence requires that any elevator be joined to any door by a hallway that allows us to continue.)

>> draw a diagram of a situation where one of these is true and the other false.

words in this chapter that it would be a good idea to understand: quantifier prefix, quantifier scope, matrix of a quantified sentence, multiple qualification.

exercises for chapter ten

A – core

1) "Someone does not love someone" can have three meanings. State all three so it is clear that each is different from the others, and tell a mini-story in which it would benatural to say "someone does not love someone" with this meaning.

2) Which of the following sentences are naturally taken to have the ∃∀ "wheel" pattern, iven the facts in the real world? (Like "someone ate all the sandwiches [who was he?]") Which have the ∀∃ Ladder pattern? (Like "someone dies every minute.")

a) Someone is the mother of every child.

b) Someone loves everyone. [Some living human !]

c) Some positive number is smaller than each positive number. ["positive number" includes zero.]

d) Somewhere is at least as far north as anywhere.

e) Some chicken lays every egg.

3) In the model below, taking the relation as **R** and the attribute as **P**

a) **Find x: Rxo**	b) **Find x: Rox**
c) **Find x: Rox**	d) **Find x: Rxo**
c) **Find x: Rxl**	d) **Find x: Rlx**

(You may want to look back at chapter 1 to remind yourself of how the difference between Rxy and Ryx is shown in a relational grid.)

Which of the following are true?

e) $\forall x\ Rxo$ f) $\forall x\ Rox$ g) $\forall x\ Rxl$

h) $\forall x\ Rlx$ i) $\forall x\ (Px \supset Rxo)$ j) $\forall x\ (Rxo \supset Px)$

k) $\forall x\ (Rxl \supset Px)$ l) $\forall x\ (Rxl \supset Px)$ m) $\exists x\ Rxm$

n) $\exists x\ Rmx$

Once these are decided, settle the truth values of

o) $\forall x \exists y\ Rxy$ p) $\forall x \exists y\ Ryx$ q) $\exists x\ \forall y\ Rxy$

r) $\exists x\ \forall y\ Rxy$ s) $\exists x\ \exists x\ (Rxy\ \&\ {\sim}Ryx)$ t) $\forall x\ \forall y\ (Px \supset Rxy)$

u) $\forall x\ \forall y\ (Px \supset Ryx)$ v) $\forall x\ \forall y\ (Rxy \supset Ryx)$ w) $\forall x\ \forall y\ (Rxy \supset {\sim}Ryx)$

	P		**R**	**l**	**m**	**n**	**o**
l	YES		**l**	YES	NO	YES	YES
m	NO		**m**	YES	NO	NO	YES
n	NO		**n**	YES	NO	NO	YES
o	YES		**o**	YES	NO	NO	YES

4) Here are four stories about deception (‘fooling’).

(a) The sun is setting as all three of our characters enter the house. Each has a problem, since Aidan has told Beth that he loves her, but needs to reassure his real love, Charlie. Beth needs a thousand dollars in a hurry to pay a blackmailer. Charlie has to get rid of an incriminating photo in a hurry. Aidan is hand in hand with Beth but pulls it to his stomach and groans “oh, I need a bathroom immediately”. “Sure says Beth, just rush, and I’ll help by holding your briefcase.” Aidan runs upstairs to meet Charlie, gives Charlie a kiss and explains that he needs to seduce Beth to get her answers to a logic quiz. Charlie is actually reassured, but pretends not to be and tells Aidan to eat the crumpled photo as a sign of devotion. Aidan does so, and returns to Beth, who meanwhile has gone through his briefcase and removed a thousand dollars in cash. So, you see, there are times everyone fools everyone.

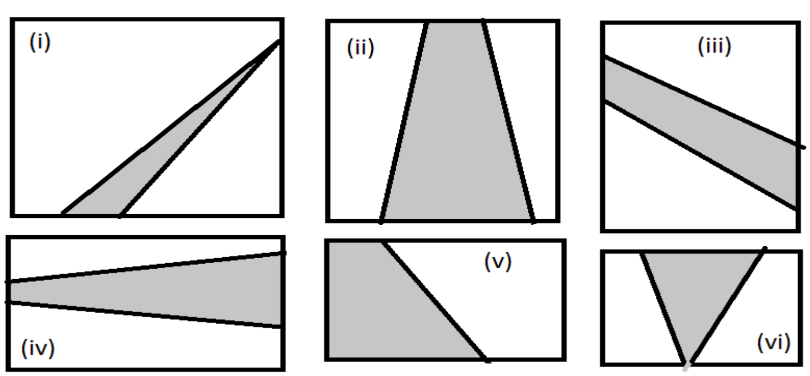
(b) Our three characters have all left the house on different evil purposes. Aidan returns home on Monday, sends an email to the others saying that there has been a leak from the upstairs water tank and he urgently needs two hundred from each to get it repaired. To his surprise, they all believe this lie and send him the money. He goes out on a binge and spends it all that evening. The next day Beth returns and is relieved to see no visible water damage, but pretends to collapse, so that Aidan sends another email, to Charlie, saying to return immediately. Once Charlie has arrived Beth persuades Aidan and Charlie to work on her logic homework, while she lies groaning on the sofa. his takes them all night, eating pizza that Charlie pays for. In the morning Charlie gives a forged version of the pizza bill to Aidan and Beth, and gets them to pay twice what the pizza cost. The moral of this little tale is that there are times when everyone fools everyone.

(c) Aidan and Beth decide to play a trick on Charlie. They lie on the floor with fake knives sticking out of their chests and fake blood all over the place. Charlie screams and calls an ambulance, and by the time it arrives Aidan and Beth have cleaned up and pretend Charlie is crazy. Charlie decides to take revenge and the next day persuades Aidan to cooperate in a trick on Beth. They hack into her email and choose a particularly embarrassing item, which they send to everyone on her address list. Of course Beth is furious, and to get back at Aidan she persuades Charlie to help her make a realistic dummy of Aidan's mother, who he fears and avoids. When Aidan comes home he goes to his room and finds his "mother" sitting on his bed. He rushes out of the house in a panic, and they lock the door, leaving him shoe-less and coat-less in the snow. One conclusion from the antics of these three is that there are times when everyone fools everyone.

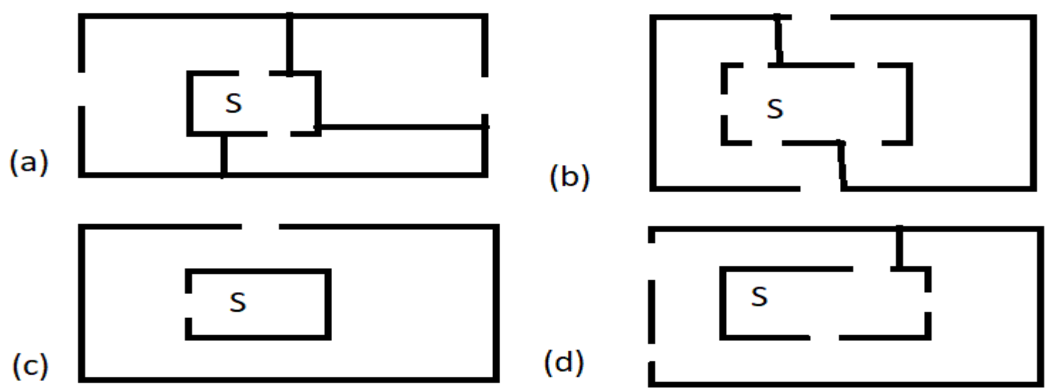
In these three stories "there are times when everyone fools everyone" has three different meanings. In one story for all people x there is a person y such that for some time t x fools y at t. In one, for all people x there is a time t such that for all people y x fools y at t. And in one, for all people x and all people y there is a time t such that x fools y at t. Which is which?

(In fact, you have to understand "all people y" as "all people y distinct from x". This should not cause you problems, and the stories would have got more complicated if I wanted to avoid this.)

5) In the six graphs below say which depict a model where (a) ∀**x**∃**y Rxy** (b) ∃**x** ∀**y Rxy** (c) ∀**y** ∃**x Rxy** (d) ∀**y** ∃**x Rxy** is true, as described in section 3 of this chapter. In some of them more than one, or none, may be true. The graphs are drawn so that the bottom edge is the x-axis, the left edge is the y-axis, the shaded area is the set of pairs (x,y) such that **Rxy**, and the whole rectangle is the domain of the model.



6) Below there are plans of the floors of a fire trap building. Each has a central room, marked S, and you can get to exits by paths from the doors of S. (Both the exits on the outside and the doors from S are indicated by gaps in the walls.) A path is a sequence of moves in a straight line allowing 90° turns but not allowing one to go back in a direction one has already taken. When a path gets beside an exit it has two go through it. (This is what you would naturally take a path to an exit to be, but I am trying to forestall quibbles.)



[A] Which floor plan satisfies which of the following conditions:

(i) For all doors from S all paths lead to an exit

(ii) There is a door from S such that some path from that door leads to an exit

(iii) For all doors from S there is a path leading to an exit

(iv) There is a door from S from which for each exit there is a path leading to it

(v) For all doors from S and all exits there is a path leading from the door to the exit.

[B] Using a single 3-place relation between doors, paths, and exits write) (i) to (v) in the language of quantifier logic state in words what the relation means.

7) Given the relational grid below, what are

(i) the individuals that have R to some individual

(ii) the individuals that have R to all individuals

(iii) the individuals to which all individuals have R

(iv) how do the answers to questions (i), (ii), (iii) relate to the issues in sections 1 and 2 of this chapter?

..R	**a**	**b**	**c**	**d**
a	YES	YES	YES	NO
b	YES	YES	YES	YES
c	NO	YES	YES	YES
d	NO	NO	YES	YES

8) Relations can be classified by restrictions they place on all the individuals they relate. A relation **R** is

reflexive if ∀**x** ∀**y Rxx**

symmetric if ∀**x**∀**y (Rxy ⊃ Ryx)**

antisymmetric if ∀**x** ∀**y (Rxy ⊃ ~Ryx)**

transitive if ∀**x**∀**y**∀**z ((Rxy & Ryz) ⊃ Rxz)**

a) One of these means that when the relation holds in one direction it always holds in the opposite direction. One of them means that when the relation holds in one direction it never holds in the opposite direction. One of them means that the relation holds between every individual and itself. And one of them means that an individual always has the relation to all things that something it has the relation to has the relation to. Which of these applies to which of the four definitions above?

b) Which of the relations below is reflexive, which symmetric, which antisymmetric, and which transitive? (A relation can be of more than one of these kinds.)

a) bigger than   b) has the same birthday as   c) loves

d) lived 300 years before   e) is in a course given by   f) mother of

f) got a better grade than   g) has the same GPA as

9) a), b), c) are quasi-English 3-quantifier sentences. Which of them corresponds to which of i) – ix) below?

a) Which farmer feeds which carrot to which donkey? Start with any farmer and then this fixes a donkey, but then any carrot will do•

b) Chains of roads link between buildings: suppose you begin with the right building. then you can take any road and find a suitable building for the third link•

c) About which lure is right to catch which fish under which of these weather conditions: choose a fish first — it can be any fish at all — and that will determine the weather, and then you can find a lure that will work for that fish under those conditions•

i)) ∃**f** ∀**d** ∃**c Ffcd**   ii) ∀**f** ∃**d** ∀**c Ffcd**   iii) ∀**f** ∃**d** ∃**c Ffcd**

iv) ∃**b** ∀**r** ∃**c Crbc**   v) ∃**b** ∀**r** ∃**c Cbrc**   vi) ∀**b** ∃**r** ∀**c Ffcd**

viii) ∀**f** ∃**w** ∃**l Ffwl**   viii) ∀**f** ∃**w** ∀**l Flfc**   ix) ∀**f** ∃**w** ∃**l Flfc**

10) Revisit exercise **14** of Chapter 2. That exercise involved searching for individuals satisfying a quantified criterion, as you can now see. The criterion was stated as " we are looking for individuals **l** such that there is another, **m**, where **Rlmxy** and **Rzsmt** — x, y, z, s, t can be any individuals at all".

(a) state this criterion using quantifiers

(b) that exercise presented a model, in the form of an arrow diagram for a 4-place relation **R**, where searches for individuals satisfying the criterion could be performed. The model satisfies the sentence that we get by putting the universal or the existential quantifier at the beginning of the criterion, binding the variable l, but not the sentence that we get by using the other. Which?

(c) [harder] a universal or an existential quantifier can be put further into the matrix, for example in second place so that it is within the scope of the quantifier binding the variable **m**. Again there are two choices, the universal and the existential quantifier. For which choice(s) is the sentence we now get true in the model?

B –more

11) Let **L** be the relation "is to the left of" as applied to these one-dimensional maps

i) [a b c] ii) [a b] iii) [a b c d e f] iv) [a]

So in the first **Lab, Lbc, Lac.**

Which of the following are true in which of these maps, taken as models for **L**?

a) ∀**x** ∃y **Lxy** b) ∃**x** ∃**y** **Bxy** c) ∀**x** ∃**y** **~Lxy**

d) ∀**x** **Lxx** e) ∀**x** ∀**y** (**Lxy** ⊃ **Lyx)** f) ∀**x** ∃**y** ∀**z** **(Lyx & ~Lyz)**

12) (You may want to look back at question **8** for definitions of reflexive and symmetric relations.) There are stronger and weaker opposites of reflexive and symmetric properties of relations. They are scope distinctions. A relation is irreflexive when it is not always reflexive, and it is anti-reflexive when it is never reflexive. A relation is asymmetric when it is not always symmetric, and it is antisymmetric when it is never symmetric. Give definitions of these two pairs of opposites using quantifier logic.

13) We have three farmers, McDonald, McTavish, McGregor, three donkeys, alice, beth, carlos, and three days, Monday, Tuesday, Wednesday. The tables below say when each farmer rides each donkey.

mon				**tues**				**wed**			
	alice	**beth**	**carlos**		**alice**	**beth**	**carlos**		**alice**	**beth**	**carlos**
mcD	NO	YES	NO		YES	YES	NO		NO	YES	YES
mcT	YES	NO	NO		YES	NO	YES		YES	NO	NO
mcG	NO	NO	YES		NO	YES	YES		YES	NO	YES

a) Writing **Rfdt** for "**f** **R**ides **d** on **t**" which of these are true?

i) ∀**f** ∃**d** ∀**t** **Rfdt**

ii) ∃**f** ∀**d** ∀**t** **Rfdt**

iii) ∀**f** ∀**t** ∃**d** **Rfdt**

iv) ∀**f** ∀**d** ∃**t** **Rfdt**

b) Suppose you want to assert ∀**f** ∀**t** ∃**d** **Rfdt** and deny ∀**f** ∃**d** ∀**t** **Rfdt** . Which of the following would be the clearest way of expressing yourself?

a) for every farmer on every day we can find a donkey that he rides that day, but we can't find for every farmer a donkey that he rides every day.

b) for every farmer we can find a donkey that he rides every day, but we can't find for every farmer and every day a donkey that he rides that day.

c) there's a donkey that all the farmers ride each day, but there isn't a donkey that is ridden each day by each farmer.

d) each farmer each day rides a donkey, but each farmer does not have a donkey that he rides every day.

14) We have three elevators and two outside doors on the ground floor of an office building, as in the diagrams below. In the night, people have come out of the elevators and left trails of muddly footprints on the floor. Focus on the relation "trail t **L**eads from elevator e to door d" — **Lted** . There are four possible patterns of footprints: in which of them is each of the sentences below true?

∀e ∀d ∃t Lted **take any elevator: for all doors there is a trail leading from that elevator to them**

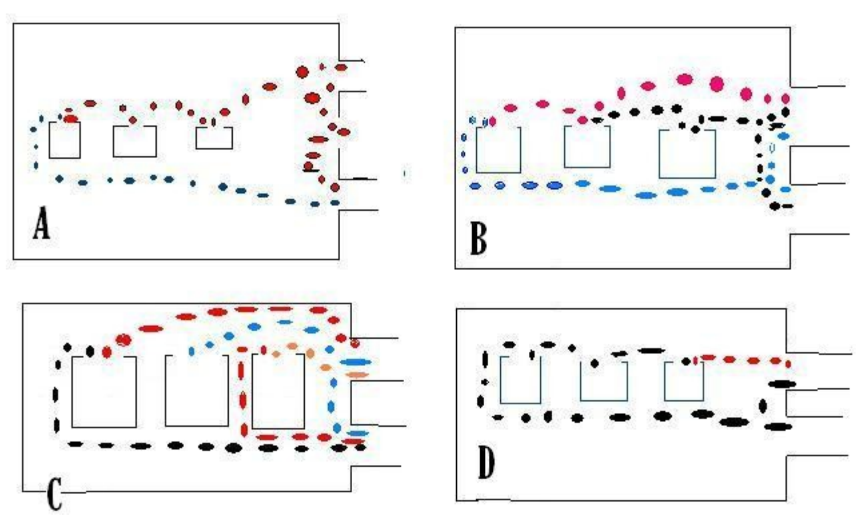
∀e ∃t ∀d Lted **take any elevator: there is a trail leading from it to all doors**

∃t ∀d ∀e Lted **there is a trail: it leads from all elevators to all doors**

∀d ∃t ∀e Lted **take any door: there is a trail leading from all the elevators to that door**

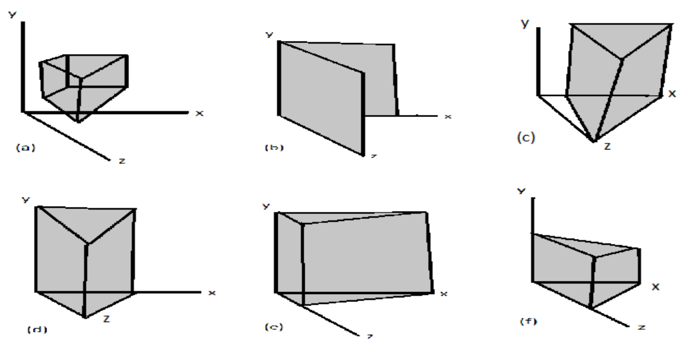
(You could find yourself expressing any of these with "there's a trail from any elevator to any door". But they're all different.)

This is a good opportunity for controlled automatic thinking, as described in question 13 of chapter 3.



15) This is an extension of exercise **5**, and so it refers to section 3 of the chapter. Below there are six diagrams, each representing a model for a 3-place relation R, and seven sentences of quantifier logic. Say of each sentence which if any model it is true in. In each diagram assume that there are no individuals beyond each axis as drawn.

I think the best way to approach this question is first to look at the diagrams without considering the sentences and think which quantifier patterns are true in each. Then with this as preparation look at the sentences and match them to the models.

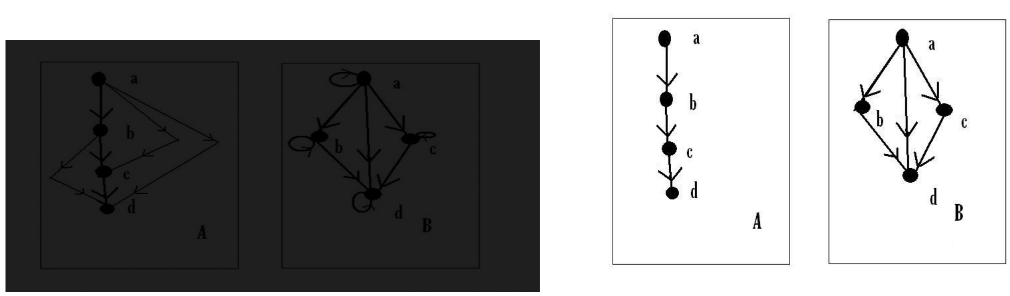


(i) ∀**x** ∃**y** ∃**z Rxyz** (ii) ∀**x** ∃**y** ∀**z Rxyz** (iii) ∀**y** ∀**z** ∃**z Rxyz** (iv) ∀**x** ∀**y** ∃**z Rxyz**

(v) ∀**x** ∀**y** ∃**z Rxyz** (vi) ∃**x** ∃**y** ∃**z Rxyz** (vii) ∃**x** ∀**y Rxyz**

C - harder

16 Call a relation **R** a *partial ordering* if it is antireflexive, antisymmetric and transitive (for definitions see question 8). **R** is also "total" if ∀**x** ∀**y (Rxy v Ryx)**. Which of the four relations below is a partial ordering, and which total?



.

17) (You may want to look back at exercise 8**)** before answering this.) A non-transitive relation is simply a relation that is not transitive. But because there are three quantifiers involved, there are several candidates for anti-transitive. Write three using quantifier logic. Which is the most plausible candidate for "never transitive"?

18) Make up three stories like those in question 5, to illustrate the corresponding quantifier orders where the relation is “x meets y at a store owned by z”

19) The two quantifier sentences ∀**x**∀**y Rxy** and ∃**x**∃**y Bx**y are closely related to a search for pairs **Find (x, y): Rxy**. How?

There is no such simple relation for ∀**x**∃**y Rxy** and ∃**x** ∀**y Bx**y . Why?

Can you describe a search: and a condition on its results that correlates with the truth of each of these?

20) The Univers∀l and ∃xistential quantifiers are sensitive to their order. ∀**x** ∃**y Rxy** is different from ∃**x** ∀**y Rxy**. But either one by itself is not sensitive: ∀**x** ∀**y Rxy** is equivalent to ∀**y** ∀**x Rxy**. (For example " everyone loves everyone" is true whenever "everyone is loved by everyone" is true. Do you see why the difference between these comes down to a difference in the order of universal quantifiers?) And ∃**x** ∃**y Rxy** is equivalent to ∃**y** ∃**x Rxy**. But not all quantifiers are like this. An example is the "Most" quantifier: "**Mostx Px**" is true when most (more than half) of the individuals in the domain are **P.** "**Mostx Mosty Rxy**" is not equivalent to "**Mosty Mostx Rxy**". ("Most people love most people" is not the same as "most people are loved by most people".) This fact is somewhat surprising, but the examples to show it can be very simple. Give a domain with just three individuals, and a relation **R**, such that **"Mostx Mosty Rxy**" is true but "**Mosty Mostx Rxy**" is false.

21) Part I of this book was about searching *within* models/databases. Part I I was about searching *for* models. Now part III is about quantifiers. Quantifiers can make very rich criteria for queries, so the topic has not changed that much. There is another connection, though. Consider three relations, two one place relations (attributes) **R**, **L** and a two place relation **S.** We are going to search in a model M given by the table below.

.

	P			**Q**	**S**	**a**	**b**	**c**	**d**	**e**
a	YES		**a**	NO	**a**	NO	NO	NO	NO	NO
b	NO		**b**	NO	**b**	YES	NO	YES	YES	NO
c	YES		**c**	NO	**c**	NO	NO	YES	NO	YES
d	YES		**d**	YES	**d**	YES	YES	YES	YES	YES
e	NO		**e**	NO	**e**	NO	NO	NO	NO	YES

(a) What are the results of these two searches? **Find x: ∃y Rxy , Find x: ∀y Rxy**

(b) How do these results relate to the attributes **P** and **Q**?

(c) Can you make the same connection for other quantified searches using R?

(d) How does this relate searching for a model to searching in a model? (The answer is hinted in chapter 2.)

(e) Can you generalize this to three-place and four- place relations, and in general relations with any number of places?

chapter 11: logical consequence in quantifier logic

11:1 (of 6) derivations and counter-models

People are pretty good at reasoning — for example when thinking about how to find things — except when their thinking is affected by several disturbing factors. Common pitfalls of human thinking are our tendency to be too confident in our beliefs, our limited capacity to keep complex information in short term memory, and our reliance on the ways of presenting information that spoken language suggests. Because we are generally pretty good reasoners, and because we are prone to some kinds of error, we usually find principles of logic fairly obvious. Who is surprised when told that from "if it is raining the sidewalks are wet" and "it is raining" we can conclude "the sidewalks are wet"? But when told that some conclusions are invalidly derived we can be surprised. We might easily derive "it is not green and not a book" from "it is not a green book". So the Boolean principle "not (green and book) if and only if not green or not book", is a useful counter to the effects of being misled by language.

When we think using quantifiers we are particularly subject to memory overload and the misleading effects of language. The range of models making premises true can be confusingly large. And language encourages us to confuse the scopes of "some", "all" and "not". So we need ways of catching plausible but invalid quantifier deductions.

One way of doing this would be to present a system of derivations for quantifier logic like the system of Boolean derivations in chapter seven. Then we might hope that a deduction that could not be reproduced in terms of the derivations would be invalid. Such systems are well developed in logic. Their basic principles are very simple, and yet when we combine them with the very complex sentences that we can have when we put quantifiers within the scope of other quantifiers the result is an extremely powerful system of deductions. It is in basic and important ways vastly more powerful than Boolean logic. Unfortunately its power is also its problem. The variety of derivations one can make is so great, and some of them are so un-obvious, that trying to incriminate invalid inferences by listing valid ones is hopeless. (In fact, to show that something was not a logical consequence by this method, one would have to check through infinitely many possibilities, so the problems are not simply practical. They raise very fundamental issues.) So although as an appendix to this chapter I describe a system of derivations for quantifier logic, this chapter has two less ambitious aims. The first is to discuss counter-models to invalid arguments involving quantifiers, that is, ways of showing that the premises can be true although the supposed conclusion is false. And the second is to describe some ways in which we can get conclusions from premises with quantifiers.

>> even in Boolean logic, is it safe to think "I cannot find a valid derivation showing this, so it must be invalid"?

>> this section seems to identify good reasoning with logical deduction, although chapter 6 section 4 warned against this. how would you put things in a more subtle way?

11:2 (of 6) scope distinctions

"Not for all" is different from "for all not" and "not for some" is different from "for some not". As we saw in the previous chapter the difference between "for every A there is a B" and "there is a B which for all A" is just the tip of a large series of differences. It is not hard to make counter-models illustrating these points.

For example if we want to confirm that "not all" is different from "all not" we should show that the following four arguments are invalid. I give a counter-model for each of them. Note that in each case one of the premises is ambiguous in everyday English, and the formulation in logical symbols fixes a meaning for it. So one effect of showing that these are invalid is to say: watch what you really mean.

not everything is A	**~∀x Ax**	**INVALID!**
everything is not A	**∀x ~Ax**	

counter-model

	A
a	NO
b	YES

You can see that in the model the premise, that not everything is **A**, is true because one thing, **a**, is not **A**. But the conclusion, that everything is not **A**, is false, since something, **b**, is A. So the argument is invalid. The conclusion is not a logical consequence of the premise.

not all A are B	**~∀x (Ax ⊃Bx)**	**INVALID!**
all A are not B	**∀x (Ax ⊃ ~Bx)**	

counter-model

	A	**B**
a	YES	NO
b	YES	YES

You can see that in the model it is not the case that all A are B, since one thing that is A, **a,** is not B. So the premise is true. But the conclusion, that all As are not B, is false, since one A thing, **b**, is B. So the argument is invalid.

There are also valid arguments relating the two quantifiers. If not everything is enjoyable then something is not enjoyable: **~∀x Jx** ⊨ ∃**x ~Jx .** And if it is not the case that something is perfect (if not even one thing is perfect) then everything is not perfect: **~∃x Px** ⊨ ∀**x ~Px .** These are special cases of two equivalences:

~∀x P is equivalent to ∃**x ~P** , whatever **P** is. And

~∃x P is equivalent to ∀**x ~P** , whatever **P** is

>> so ∀**x P** and **~∃x ~P** , and also ∃**x P** and **~∀x~P**, are equivalent. do you see why?

In spite of these equivalences, there are very similar patterns that are not valid. "Not all cats hunt mice" does *not* entail "Some cats do not hunt mice". This may seem surprising. But consider the following:

Not everything he said was false does not entail that something he said was not false: for he may have said nothing.

Not all the pearls in the drawer are black does not entail that some pearl in the drawer is not black: for there may be no pearls in the drawer.

So if we are being really careful we should say that "Not all cats hunt mice" *plus the additional assumption that there are cats*, entail "some cats do not hunt mice".

>> "But everyone knows there are cats, so we don't really need to state this assumption." describe situations when this is not a good reply.

These deductions are more familiar in English if we also say "Nothing" and "No one" for "it is not the case that some", and use "even one" or "at least one" as a variant on "there is". So it follows from "everyone is not happy" that "no one is happy" or equivalently "not even one is happy". And it follows from "not all individuals are listed" that "at least one individual is not listed".

>> write these "nothing", "no one", "at least one" sentences using the quantifiers ∀ and ∃.

A surprising invalidity, closely related to the fact that "not all cats chase mice" does not entail "some cats do not chase mice", concerns deriving "some" from "all". "Everything is A" entails "Something is A", whatever we choose for A. The reason is that we only consider models with non-empty domains, and if the domain is not empty and everything in it is A then any of those things will show that something is A. Contrast this to "any life on Mars is carbon-based" which can be true when "there is no life on Mars" or to "all the pearls in drawers 1 to 10 are black" which entails "all the pearls in drawer number 3 are black", even though there may be no pearls in drawer number 3.

>> is the difference between "all" and "any" relevant here?

>> the pearls example is a bit different. how?

In logic, as you know, "All unicorns have horns" is symbolised with the pattern **∀x (Ux ⊃ Hx)** and "some unicorns have horns" with the pattern **∃x (Ux & Hx)**. So in a domain that has no unicorns the former is true and the latter is false. The "all" sentence is true because the antecedent of the conditional **Ux ⊃ Hx** is always false, and so, given the truth table for ⊃, the whole sentence is false. (See the next section for a little more detail here.) So it will also be true that all unicorns fly. And also that all unicorns do not fly!

>> so are "all unicorns fly" and "all unicorns do not fly" both true? If not which one is false? both?

>> we do not use models or databases with empty domains. what might the reasons be?

You may have noticed that while I have given counter-models for some invalid arguments I have listed some as valid without really showing why they are. We begin this in the next section. Before then, one more basic kind of counter-model, in this case involving a relation. Again, the English sentences are ambiguous and could be read in different ways.

everything has R to something	$\forall x\ \exists y\ Rxy$	**INVALID!**
something has R to everything	$\exists x\ \forall y\ Rxy$	

everything has R to something	$\forall x\ \exists y\ Rxy$	
something is such that everything has R to it	$\exists x\ \forall y\ Ryx$	**INVALID!**

We can use the same counter-model for both of these.

R	a	b
a	YES	NO
b	NO	YES

You can see that in the model each individual has R to something, so the premise (of both arguments) is true. But no individual has R to everything, so the conclusion of the first argument is false. And there is no individual that everything has R to. So the conclusion of the second argument is false. So both arguments are invalid.

11:3 (of 6) arguments that are almost Boolean

"All people are animals; all animals die. Therefore all people die." That is a valid argument. In fact it is a syllogism, a traditional form where we have two premises each of which involves only one quantifier[24]. In this book we are not particularly concerned with syllogisms, but they are an easy source of examples. Put in logical symbols the argument has this form:

$$\forall x\ (Ax \supset Bx)$$
$$\underline{\forall x\ (Bx \supset Cx)}$$
$$\forall x\ (Ax \supset Cx)$$

>> what about "some people are animals; some animals die. therefore some people

24Syllogisms were for centuries one of the two standard cases of careful reasoning in philosophy and mathematics. The other standard case was Euclid's geometry. But there is an irony here: syllogistic is not adequate for representing the reasoning in Euclid's proofs, which involve intrinsically many place relations. As far as I can see, nobody noticed this.

die", that is

$$\begin{array}{l} \exists x\ (Ax\ \&\ Bx) \\ \underline{\exists x\ (Bx\ \&\ Cx)} \\ \exists x\ (Ax\ \&\ Cx) \qquad ? \end{array}$$

is this a valid argument?

The validity of the "all people are animals..." argument can be seen in two ways, which are in fact closely related. In this section I describe only the first way, leaving the second way, which your course may be skipping, to the following section.

first way: small domains Suppose we are considering a model domain of only two individuals. Call them **a**, and **b**. Then the two premises are true in the model if and only if

(Aa ⊃ Ba) & (Ab ⊃ Bb) and

(Ba ⊃ Cb) & (Ba ⊃ Cb) are true.

But these have **(Aa ⊃ Ca) & (Aa ⊃ Ca)** as a logical consequence. And if it is true in the model then ∀**x (Ax ⊃ Cx)** is true. So if the premises of this syllogism are true in this model then the conclusion is also true.

We can reason the same way for a 3-individual model, and in fact for any model whose domain contains finitely many individuals. So for finite models at any rate, if the premises of this argument are true then its conclusion is true.

We can make a case for the validity of many quantifier arguments in this way, by considering ∀ as a big conjunction, one conjunct for every member of the domain, and ∃ as a big disjunction. Note that the equivalence of ∀**x P** and **~**∃**x ~P** then becomes an instance of de Morgan's law, that **(p&q)** is equivalent to **~(~p v ~q)** , but with a large number of conjuncts or disjuncts. In fact we often express de Morgan's law with something near to a quantifier. We say "if they're both true then it cannot be that one of them is false", or "if they are not both true then one of them is false."

>> what does the equivalence of **~∃x P** and ∀**x ~P** become?

We can handle premises and conclusions with more than one quantifier in this way, too. Consider for example the following valid argument. (It resembles the invalid quantifier patterns of the previous section in a potentially confusing way; that might be part of the explanation why those earlier arguments can seem valid.)

<u>something has R to everything</u>	**<u>∃x ∀y Rxy</u>**
for everything there is something that has R to it	**∀x ∃y Ryx**
	VALID!

>> this valid argument can also be seen as ∃**x** ∀**y Rxy** |= ∀**y** ∃**x Rxy**. do you see why? does this way of writing it make it seem more or less obviously valid?

Consider a two element domain again. The premise is true when

(Raa & Rab) v (Rba & Rbb) is true, and the conclusion is true when

(Rab v Rab) & (Rba v Rbb) is true. Again the conclusion is a Boolean consequence of the premise. It is worth getting out pencil and paper, seeing how the premise and conclusion do take these forms under the conjunction and disjunction interpretations of the quantifiers, and seeing that the conclusion really does follow from the premise, by propositional logic. It is **p v (q & r) |= (p v q) & (p v r)**, but where **p** is itself a conjunction.

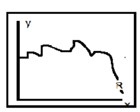
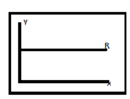
(Consider a particular case, with just Bo and Mo. One of them is angry at both of them, so either Bo is angry at Bo and Bo is angry at Mo or Mo is angry at Bo and Mo is angry at Bo. So it is true that one of the two is angry at Bo and that one of the two is angry at

Mo.)

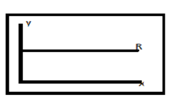
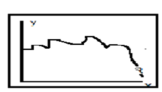
Again we can do this not only in a 2 element domain, but in any finite domain. This isn't a watertight proof that the argument preserves truth in all domains whatever, since there are some arguments that are valid in all finite domains but not in all domains. (See exercise **13** at the end of this chapter.) Still, these considerations should give us some reassurance about these arguments. See exercise 1(b) and 3 for other arguments involving quantifiers which can be turned into Boolean validities.

>> give an example of something that is true in a two-element domain that is *not* true in some larger finite domain.

The fact that $\exists \mathbf{x}\, \forall \mathbf{y}\, \mathbf{Rxy} \models \forall \mathbf{y}\, \exists \mathbf{x}\, \mathbf{Rxy}$ — or equivalently $\exists \mathbf{x}\, \forall \mathbf{y}\, \mathbf{Rxy} \models \forall \mathbf{x}\, \exists \mathbf{y}\, \mathbf{Ryx}$ **,** $\exists \mathbf{y}\, \forall \mathbf{x}\, \mathbf{Rxy} \models \forall \mathbf{x}\, \exists \mathbf{y}\, \mathbf{Rxy}$ **,** and $\exists \mathbf{y}\, \forall \mathbf{x}\, \mathbf{Rxy} \models \forall \mathbf{x}\, \exists \mathbf{y}\, \mathbf{Ryx}$, which differ just by using different variables **—** holds not just in finite domains but taking all domains into account, can be seen intuitively by considering diagrams like those from section 3 of the previous chapter. Although it is NOT true that

 $\models$ 

It is true that

 $\models$ 

And this is intuitively right: a straight line absolutely horizontal to the right is a special case of a horizontal barrier in general, but not the other way around. (And similarly an absolutely vertical straight line is a special case of a vertical barrier in general, but not the other way around: the various equivalent formulations correspond to flipping the diagram around some diagonals, without changing its basic shape.).)

11:4 (of 6) second way: indirect argument **[Check with your teacher whether you should study this section.]**

We often reason, especially in mathematics, using variables for which we can substitute anything in the domain, or any member of some particular class of things, for. For example if we say that $(3x)^2 = 9x^2$ we mean that this equality is true when x is zero, and when x is 0.5, and when x is 423, and in fact whenever x is a real number.

>> is this the same as saying "$\forall x\ (\text{Real}x \supset (3x)^2 = 9x)^2$"? exactly the same?

So consider the syllogism "all humans are animals; all animals die; therefore all humans die". Assume that the members of the domain are d_1, d_2, ... without assuming anything about how many these are. Then make a derivation as follows

$\forall$**x (Ax $\supset$ Bx)** (1)

$\forall$**x (Bx $\supset$ Cx)** (2)

$d_1 \supset Bd_1$ (3_1) (see note 1)

$Ad_2 \supset Bd_2$ (3_2) (see note 1)

....

$Bd_1 \supset Cd_1$ (4_1) (see note 1)

$Bd_2 \supset Cd_2$ (4_2) (see note 1)

....

/ \

$Ad_1 \supset Cd_1$ | **$\sim(Ad_1 \supset Cd_1)$** (5_1) EM

| **Ad_1 & $\sim Cd_1$** (6_1)

| **Ad_1** (7_1)

| **$\sim Cd_1$** (8_1)

| / \

| **$\sim Ad_1$** X | **Bd_1** (9_1) from (3_1)

| / \

| **$\sim Bd_1$** X | **Cd_1** X (10_1)

|

$Ad_2 \supset Cd_2$ | **$\sim(Ad_2 \supset Cd_2)$** (5_2) EM

| **Ad_2 & $\sim Cd_2$** (6_1)

| **Ad_2** (7_1)

| **$\sim Cd_2$** (8_1)

| / \

| **$\sim Ad_2$** X | **Bd_2** (9_1) from (3_1)

| / \

| **$\sim Bd_2$** X | **Cd_2** X (10_1)

|

.... see note 2!

|

$\forall$**x (Ax $\supset$ Cx)** (11) (see note 3)

This is a lot like the Boolean derivations we studied in chapters 7 and 8. There are significant differences, though, since it is potentially bigger than any Boolean derivation could get. Note three important points.

note 1: Line 3 is where the derivation gets big. The "...." indicates more lines, for $d_{3,}$ $d_{4,}$ or as many as there are individuals in the domain. Clearly unless the domain is small we can not write it on the page without the "....". I have written out Line **(3_1)** and line **(3_1).** They are all consequence of the premise at (1), and all the lines 4 are consequence of the premise at (2) because that is what $\forall$ means: for each member of the domain. But this is not something that our rules for Boolean derivations covered.

note 2: The derivation of $\mathbf{Ad_2 \supset Cd_2}$ by closing all other branches would be exactly parallel to that of $\mathbf{Ad_1 \supset Cd_1}$, as would the corresponding derivations for all the other individuals in the domain. Notice that one branch includes all of these unclosed conditionals.

note 3: Line 11 adapts conjunctive argument, discussed in chapter 8, because of the fact that $\mathbf{Ad_1 \supset Cd_1}$, $\mathbf{Ad_2 \supset Cd_2}$ **....** are the only branches of the tree that have not closed, and the meaning of ∀, given the fact that $\mathbf{d_1}$, $\mathbf{d_2}$**,....** are all the members of the domain. We are applying conjunctive argument in an extended way, to a long branch — perhaps *very* long — but it is basically the same rule.

>> is the fact that the derivation if fully written out could have many, perhaps infinitely many, branches a reason to mistrust it?

>> (for mathematicians) some models have indenumerable (uncountable) domains: is this an extra worry here? (it turns out that this makes no difference, but that is not an obvious fact.)

We can make similar derivations with the ∃ quantifier, though now we are extending disjunctive argument from chapter 8 rather than conjunctive argument, and the derivation gets very wide rather than very long. A simple, almost trivial, example is **∃x (Ax & ~Bx)** ⊨ **∃x ~(Ax & Bx)**. ("Some anarchists are not Brazilian: therefore some people are not both anarchist and Brazilian"). We can turn this into a derivation as follows.

1)	<u>**∃x (Ax & ~Bx)**</u>	
	/ ..../..... | ....\......\	
2)	$\mathbf{Ad_1}$ **& ~**$\mathbf{Bd_1}$ **..** $\mathbf{Ad_n}$ **& ~**$\mathbf{Bd_n}$ **.................**	(note 1)
3)	**~(**$\mathbf{Ad_1}$ **&** $\mathbf{Bd_1}$**) .. ~(**$\mathbf{Aa_n}$ **&** $\mathbf{Bd_n}$**) ....**	(note 2)
4)	**∃x ~(Ax & Bx) .. ∃x ~(Ax & Bx) ....**	(note 3)
5)	**∃x ~(Ax & Bx)**	(note 4)

note 1: this uses the analogy between ∃ and **v**. Think of the existential quantifier as an

infinite disjunction and then the **v**-rule of derivations applies. But it is also where the derivation gets infinitely wide.

note 2: in each case here a few steps of derivation in propositional logic could be inserted. I have left them out for simplicity, given that the items on line 3 are so obviously consequences of the corresponding items on line 2.

note 3: here we could again appeal to the analogy between the existential quantifier and disjunction. All we could use a very basic property of the existential quantifier, **Aa** ⊨ ∃**x Ax** (if a particular thing has an attribute then there is something that has it).

note 4: this is where the indirect principle of disjunctive argument is used.

∃**x ~(Ax & Bx)** is a consequence of each of the disjuncts we get when we construe the premise as an infinite disjunction, so it is a consequence of the disjunction itself.

These were very simple examples. They did not feature the main source of the power of quantifiers, the interactions between them when several quantifiers are found in the same sentence. But they do illustrate how systems of derivations can be extended to quantifier logic.

There are many ways of making systems of deduction and derivation for quantifier logic. They typically do not involve derivations that are infinitely wide or infinitely long. Instead, they make subtle use of variables, involving restrictions which have to be stated and followed precisely, and often principles of indirect argument which resemble the uses of conjunctive and disjunctive argument from this section. These are often combined. In the appendix to this chapter I describe a a fairly simple example of this. It combines the use of variables and one form of indirect argument, argument by contradiction. But it will not capture the patterns of many everyday arguments and mathematical proofs.

The standard systems are complete, meaning that if one sentence is a logical consequence of a set of premises then they will always show that it is. This is a remarkable fact, as suggested by the assumptions behind my two sample derivations. They assumed that we can write down a series of names that apply to everything in the domain, and in fact to everything in any domain in which the premises are true. But even with infinitely many names, this is not obvious. (There may be into innumerably many individuals in the domain, and even if there are no more than there are integers the names might get attached to, say, just the even-numbered ones.) And though logical consequence is a matter of what holds in all models for the premises, including infinitely many infinitely big ones, derivations in these systems are finite. So the fact that there are complete systems of quantifier logic, that the syntactical and the semantic approaches coincide, is not something to take for granted before one sees a proof. But I am not going to give a proof, either here or in the appendix.

One way of seeing what the completeness proofs show is that a system of derivations for quantifier logic can construct enough models to substitute for all the possible models for the sentences involved, much as truth tables can substitute for all the possible truth assignments. That is how they tame infinity. But this is itself a remarkable fact.

11:4 (of 6) to and from prenex form

In everyday language we distribute quantifier words throughout a sentence, in roughly the places names can go. The previous chapter argued that we appreciate the force of quantifier logic when we see its resources for putting quantifiers at the beginning of a sentence, so that the variables bound by the quantifier rather than the quantifiers themselves, are distributed within the sentence like names. But sentences in logic with quantifiers can have them in the middle of the sentence too. The (false) English sentence "every Canadian loves some moose" could be translated into logic as either of the following:

$\forall$**x (Cx** $\supset$ $\exists$**y (My & Lxy))** or

$\forall$**x** $\exists$**y (Cx** $\supset$ **(My & Lxy))**

We do not have to worry about the choice between these, because they are logically equivalent. But this does raise an important question: when is a sentence equivalent to one whose quantifiers all come at the beginning? When can we 'move' the quantifers to the front? A sentence with all its quantifiers at the beginning is said to be in *prenex form*. In fact every sentence is logically equivalent to one in prenex form, and in this section we will see the rules for this. But first some *non*-equivalences, both as warnings against haste and as practice in finding counter-models.

$\exists$**x (Ax & Bx)** is not the same as $\exists$**x Ax &** $\exists$**x Bx** . There are birds and there are insects, but there is nothing that is both a bird and an insect. (not even a bee hummingbird.) The simplest counter-model is

	A	B
m	YES	NO
n	NO	YES

>> this counter-model shows that $\exists$**x Ax &** $\exists$**x Bx** can be true while $\exists$**x (Ax & Bx)** is false. so which one is not a logical consequence of which?

$\forall$**x (Ax v Bx)** is not the same as $\forall$**x Ax v** $\forall$**x Bx**. Everything is either a bird or a non-bird, but it is not true that either everything is a bird or everything is a non-bird. The same counter-model illustrates this, showing this time that ∀**x (Ax v Bx)** can be true while ∀**x Ax v** ∀**x Bx** is false.

>> so which is not a logical consequence of which?

∀**x (Ax ⊃ Bx)** is not the same as ∀**x Ax ⊃** ∀**x Bx** . It is false that all moose have wings" but "if everything is a moose then everything has wings" is true, on the Boolean reading of "if", since it is false that everything is a moose. The same counter-model that we have used for the previous two cases will do for this.

>> give Venn diagrams as counter-models for these

Now to state the ways in which we can validly move quantifiers to the beginning of a sentence. First consider cases such as ∀**x Ax & P** , where **Ax** contains the variable **x** and **P** does not. (**P** might be like **Cd** or like ∃**y Dy**, in any case there is no variable **x** in it.) Then ∀**x Ax & P** is equivalent to ∀**x (Ax & P)**. Everyone is happy and 2+2=4 is true if and only if everyone is such that they are happy and 2+2=4. To argue it more carefully, note that if **P** is true then ∀**x Ax & P** is true if and only if if all individuals in in the domain are **A**, which is the case if and only if every individual x is such that **Ax & P**. And if **P** is false then both ∀**x Ax & P** and ∀**x (Ax & P)** are false.

Similar reasoning will show that

∃**x Ax v P** is equivalent to ∃**x (Ax v P)**
∃**x Ax ⊃P** is equivalent to ∃**x (Ax ⊃ P)**
∀**x Ax v P** is equivalent to ∀**x (Ax v P)**
P ⊃ ∀**xBx** is equivalent to ∀**x (P ⊃ Bx)**

There are a couple of similar formulations that I haven't included because they would be wrong; we'll get to them.

Often applying these equivalences will give two quantifiers at the front, where there was originally only one. For example

∃**x Ax &** ∃**x Bx** is equivalent to ∃**x Ax &** ∃**y By** is equvalent to ∃**x** ∃**x (Ax & By)**

∀**x Ax v** ∀**x Bx** is equivalent to ∀**x Ax v** ∀**y By** is equivalent to ∀**x** ∀**x (Ax v By)**

∀**x Ax** ⊃ ∀**x Bx** is equivalent to ∀**x Ax** ⊃ ∀**y Bx** is equivalent to ∀**x** ∀**x (Ax** ⊃ **Bx)**

>> between the second and third of each of these equivalences there could be an intermediate step, in which only one of the two quantifiers is moved. can you state this step?

The general message of these equivalences is: move quantifiers to the front, after changing variables to avoid conflicts between distinct quantifiers. There are a few exceptions to this message, though. The first pair we have already seen, and is not surprising. ~∃**x Ax** is equivalent to ∀**x ~Ax** (rather than to **~**∃**x Ax**) and **~**∀**x Ax** is equivalent to ∃**x ~Ax** (rather than to ∀**x** ~**Ax**.) The other pair can seem surprising

∃**x Ax** ⊃ **P** is equivalent to ∀**x (Ax** ⊃ **P)**

∀**x Ax** ⊃ **P** is equivalent to ∃**x (Ax** ⊃ **P)**

∃ becomes ∀ when it is moved from the antecedent of a conditional, and vice versa. This makes sense when we think of **P** ⊃ **Q** as equivalent to **~P v Q** : the antecedent of a (material) conditional is a negatively flavoured position. And the first of these can sound correct in ordinary language: "if even one person comes to the party I'll be delighted" sounds equivalent to "consider anyone: if she comes to the party I'll be delighted". But the second can definitely sound wrong. It identifies, for example, "if all the sandwiches are mouldy then we have no lunch" with "there is a sandwich such that if it is mouldy then we have no lunch". Why should one sandwich be more important than the others?

But think of the sandwich case this way: suppose we are examining all the sandwiches, hoping to find at least one that is edible, and stopping when we do. Think of the last one we look at. If even that one is mouldy then we can say Good-Bye to lunch. So saying "if all the sandwiches are mouldy then we have no lunch" is the same as saying "consider the last sandwich we examine: if even that one is mouldy then we have no lunch." The intuitively troubling nature of the equivalence is a consequence of thinking of the Boolean conditional as if it were the "if" of everyday language and thinking that "there is" requires us to identify an individual in advance. (There is also a mysterious way in which "if" affects our understanding of scope, so that we tend to think of "something is such that (if it is A then P)", as — wrongly — "something is such that if it is A, then P".)

>> find other examples where the equivalence of ∀**x Ax** ⊃ **P** and ∃**x (Ax** ⊃ **P)** seems at first surprising, and analyse them in ways like my treatment of the sandwich example.

The general rule for converting a sentence to prenex form is as follows:

First whenever two quantifiers use the same variable re-letter one of them so they are different. *Second* move each quantifier in turn as far to the front of the sentence as you can without moving it past any sentence it was in the scope of. *But* in doing this universal quantifiers become existential and existential become universal if they are negated or in the antecedent of a conditional.

This sounds complicated but in practice it is straightforward. For example

∀**x Ax** ⊃ ∃**x** ∀**y (Rxy & Ryx)** becomes ∃**x** ∃**y** ∀**z (Ax** ⊃ **(Ryz & Rzy)**

$\forall$**x ($\exists$y Rxy $\supset$ $\exists$y Ryx)** becomes $\forall$**x $\forall$y $\exists$z (Rxy $\supset$ Rzx)**

$\exists$**x $\forall$y (Ay $\supset$ Rxy) $\supset$ $\exists$x $\forall$y Sxy** becomes

$\forall$**x $\exists$y $\exists$z $\forall$w ((Ay $\supset$ Rxy) $\supset$ Szw)**

Sometimes the rules can be applied in two ways to give two different results (for example $\forall$**x Ax** $\supset$ $\forall$**x Bx** can become both $\exists$**x $\forall$y(Ax $\supset$ By)** and $\forall$**y $\exists$x (Ax $\supset$ By)** .) But when this happens the results are equivalent. ("There is one of us such that if that person is heartless then we are all doomed" is equivalent to "We are all in this situation: if there is a heartless person among us then we are doomed.")

>> give several interpretations of A and B for $\forall$**x Ax** $\supset$ $\forall$**x Bx** and rephrase the two prenex forms so that it is clear that they are equivalent.

The forms with the quantifiers spread through the sentence are often easier to understand using resources derived from everyday language, and the prenex forms are usually easiest to manipulate in purely mechanical ways, and also often give more easily give logical insight. For example consider the second example of the previous paragraph: the version nearer to everyday language is

$\forall$**x ($\exists$y Rxy $\supset$ $\exists$y Ryx)**

and the prenex version is

$\forall$**x $\forall$y $\exists$z (Rxy $\supset$ Rzx)** .

For example

consider anyone: if they have a child then they have a friend

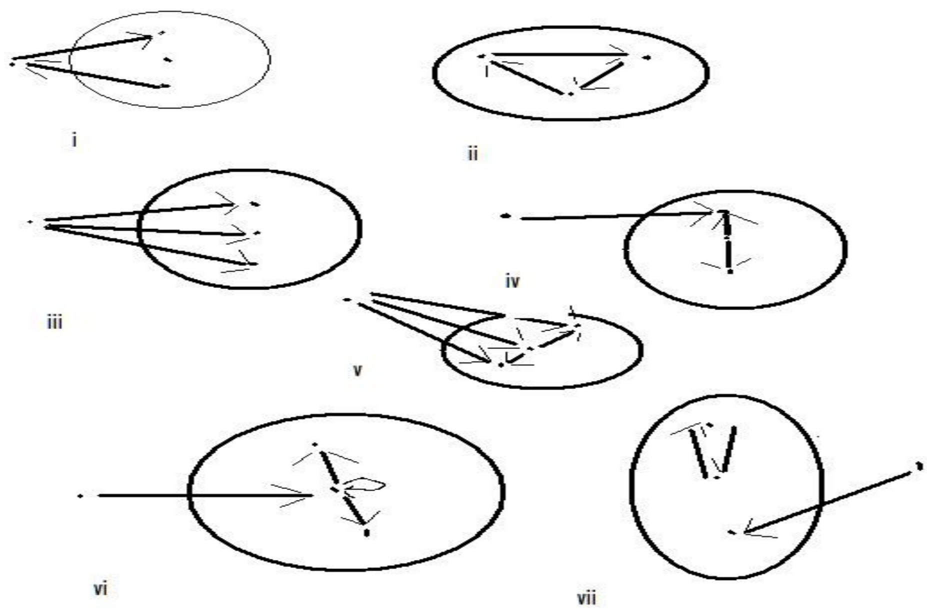
consider any pair of people: there is a person who if the second of the pair is child of the first then that person is friend to the first. Rephrased: take any parent and child and you'll be able to find a friend of the parent.

The first version is easier to understand quickly. But the truth of the second might be easier to check. Instead of checking of each person whether they had a child and then if they have a child checking if they have a friend, we would list all pairs and filter them for parenthood, and for those that got through check for the existence of a friend.

11:5 (of 6) Goldilocks and the three bears

An important part of understanding a sentence is knowing what models it can have. Knowing what is and is not a logical consequence of the sentence, and what it is and is not a logical consequence of, helps here. But many sentences have a large number of models, including some that are very different from others. In this section I try to give you a sense of how wide the variety of models for a sentence is, by giving practice in answering the question "is this sentence true in this model?"

The models will be presented as arrow diagrams, with arrows for a two-pace relation and circles for attributes. Interpret the the arrows as a relation S, “sees” , and the circle as an attribute B, that of being a bear. Look at model (i). What is true in it?



True in (i): That there are three bears; that there is someone, not a bear who sees one of them; that a bear sees 'her'. So these are true in (i).

∃x Bx ∃x ~Bx ∃x ∃y Sxy ∃x ∃y (By & Sxy)

∃x ∃y (~By & Sxy) ∃x ∃y (Bx & ~By & Sxy)

These say that there is a (at least one) bear, and one non-bear, that someone sees someone, that there is someone who sees a bear, that there is someone who sees a non-bear, that some bear sees a non-bear.

Contrast (i) with (iii). At first they may look similar. The difference is that in (iii) all the see-arrows go from the non-bear to the bears, and in fact to all the bears. So *if* something is a bear the non-bear sees it.

∃x ∀y (By ⊃ Sxy)

It is also true in (iii) that the person only sees bears. If she sees it, it's a bear.

∃x ∀y (Sxy ⊃ By)

Now look at (ii). We've lost Goldilocks. But still there is a lot of seeing going on between the bears. You can imagine the three of them as pictured in the bear enclosure, each one looking at the back of the next. So all of these are true

∀x (Bx ⊃ ∃y Sxy) **∀x ∃y (Bx & Sxy)**

∀x ∀y (Sxy ⊃ (Bx & By)) **∀x ∀y (Sxy ⊃ ~Syx)**

>> how can we say each of these in English? which ones are also true in (i) and (iii)? (don't look for the answers. think first.)

Now think in the opposite direction, from sentences to models. Consider the sentences

∃x ∃y (Bx & Sxy) **∃x Sxx** **∃x ∃y ∃z (Sxy & Sxz & ~Syz)**

Which models are these true in? The first requires two individuals, of which the first is a bear and sees the second, which may or may not be a bear. So it is true in all except (iii). The second requires one individual, which sees itself. So it is true only in (vi). The third requires three individuals such that the first sees the second and the first sees the third, and the second does not see the third. In English we'd say "Someone sees two others, who don't see each other." This is true in (iii), (iv), (v), (vi).

So in which of the models is each of the following true?

∀x ∀y (Sxy ⊃ Bx) **∀x ∀y (Sxy ⊃ By)**

∀x ∀y (Sxy ⊃ ∃z (Bz & Sxz))

∃x ∀y (∃z (Bz & Syz) ⊃ Sxy)

The first two should not be hard (but notice how they are different.) In the third focus on **∃z (Bz & Sxz)**. This says that x sees a bear. So the whole sentence says that if an individual sees an individual then the first individual sees a bear. We could say this in looser English as "anything that sees anything sees a bear", but we would have to be careful not to understand this as "anything that sees everything sees a bear". So it is true in all the models except (i).

In the fourth focus first on the ∃**z (Bz& Syz)**. It says that y sees a bear. So the whole sentence says "There is something. if anything sees a bear, then it also sees that thing." In more regular English: there is something that sees anything that sees a bear. So it is true in none of the models.

There is a third kind of task to be considered. It is more creative: given a sentence to think of models in which it is true, ideally as different from one another as possible. Start with a simple case, ∀**x** ∀**y (Sxy ⊃ (Bx & By))** . In any model for this sentence whenever one individual has the relation S to another both have the attribute B. Consider the models below. (a) to (d) are all models for this sentence. But they are very different from one another. (To put it in mathematical terms there is no structure-preserving map, no isomorphism, between any two.) (a) is what you might have imagined on understanding the sentence: bears see bears and only bears see bears. This is true in (b) also, but most bears do not see anything. The sentence is true in (c), though surprisingly because there are no bears. In (d) there are lots of bears, but they fall into two classes, which ask to be defined. In (e) only bears see or are seen, but most of them do neither. In (f) again the bears fall into two classes, and the diagram suggests more generalization. ("Mother bears watch baby bears, but the babies don't look back" perhaps.)

>> how could we define the two kinds of bears in (d)? in (f)?

A more formal way of asking "make varied models for this sentence". Is "make a model for S_1 in which S_2 is not true." In other words "show that S_2 is not a logical consequence of S_1".

(a) shows that ∀**x** ∀**y** **(Sxy ⊃ Syx)** does not follow from

∀**x** ∀**y** **(Sxy ⊃ (Bx & By))**,

(b) shows that ∀**x** ∀**y** ∀**z** **((Sxy & Sxz)⊃ Syz)** does not follow, from the same assumption.

(c) shows that ∃**x Bx**., does not follow from it,

(d) that ∀**x (Bx ⊃**(∃**y Syx v** ∃**y Sxy))** does not follow from it.

(e) that ∀**x ~(Ax & Bx)** ⊃ ∀**x** ∀**y** (**Ax** ⊃ ~ **Sxy)** does not follow from it.

And if your first idea of a model for the sentence was along the lines of (f), you would think that ∀**x** ∀**y** ∀**z** **(Sxy** ⊃ ~ **Szx)** followed, but (a) and more subtly (d) show that it does not.

>> express all these in everyday language

11:6 (of 6). Infinity

Some of the power of quantifier logic comes from the fact that it does not restrict the sizes of domains. ∀**x Ax** is true in a domain {**a, b**} if **Aa & Ab** is true, and it is true in a domain {**a, b, c}** if **Aa & Ab & Ac** is true, and so on. This is what forced the justification of simple logical consequences in section 3 above to be so complicated. (We can avoid some of the complications by subtle use of variables, as in the appendix to this chapter. But the use has to be very subtle, easy to get wrong.) This suggests a very basic question. Given that some sentences with quantifiers are true in models with very small domains, and that some need very large domains — something that is plausible but which we have not actually proved — are there sentences which are true only in models with infinite domains?

Yes. There are sentences that are true only in infinite models. Let's try to make one.

In an infinite model given any finite number of individuals there will be one that is "beyond" them. So we might begin with ∀**x** ∃**y** **Bxy** . "Take anything. there is something beyond it." (Notice that the quantifiers have to be in this order. ∃**x** ∀**y** **Bxy** won't do it.)

>> why will ∃**x** ∀**y** **Bxy** not do it?

But this allows a 2-individual model in which each is beyond the other. (See i below.) We can rule this out by adding another condition: anti-symmetry, **B** never goes both ways.

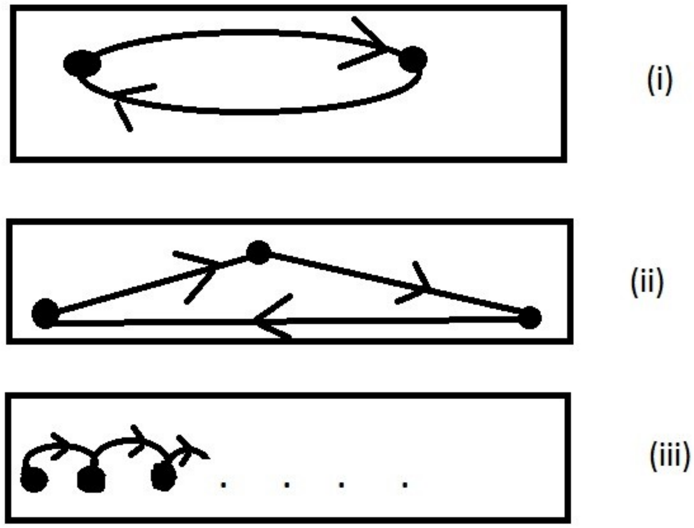
∀**x** ∀**y** **(Bxy** ⊃ ~**Byx**)

This rules out the model we just considered. But it allows another unintended model, in which we have three individuals in a cyclic pattern. (see ii.) Then **B** never goes directly backwards, but goes round in a cycle, a finite cycle. We can rule this out by requiring B to be transitive.

∀**x** ∀**y** ∀**z** **((Bxy & Byz)** ⊃ **Bxz**)

Then we have what we want. Given any series of individuals linked by **B**, there will be another that is beyond all of them. So if we take all three conditions and make a single sentence out of their conjunction, that sentence will only have infinite models. The simplest will look like the numbers 0,1,2,3,... related by < . (See iii.)

>> find another model for these sentences, besides this one. it will also be infinite.



This example illustrates the importance in our thinking of being able to have one quantifier in the scope of another. It is because of this that we can have the concept of infinity. (Creatures who could not understand embedded quantifiers could not think about infinity.) We can think “for every moment there is a later (or an earlier) one” or “for every event there is a cause”, or “for every number there is a bigger one”. For humans these are not very difficult thoughts, but they have many interesting consequences.

You can take this as the final persuasion that quantifiers are a big step beyond Boolean connectives. In an infinite domain $\forall$**x Bx** cannot mean **Ba_1 & Ba_2 & ...& Ba_n** for any integer n. It has to go on forever. And that isn’t a regular Boolean connective. It is something quite different, a quantifier

words in this chapter that it would be good idea to understand: almost Boolean argument, empty domain, prenex form, syllogism.

exercises for chapter 11

The exercises for this chapter are not in the three parts that organized the exercises for earlier chapters. But they are in a rough order from easier to harder. So if you want a challenge try those towards the end, and if you find the later ones difficult do not let this trouble you.

1) The study of syllogisms is the simplest part of quantifier logic. A syllogism is an argument with two premises and a conclusion, all of the form "all A's are B's", "some A's are B's", or "no A's are B's". Aristotle studied syllogisms more than 2000 years ago.

a) Below are four valid syllogisms. Symbolize each of them in the language of quantifier logic.

i) all cats are animals
<u>no martians are animals</u>
no martians are cats

ii) all cats are felines
<u>some animals are cats</u>
some animals are felines

iii) there is an honest philosopher
<u>no-one honest is happy</u>
some philosopher is not happy

iv) all oilers fans are drunks
<u>some oilers fans love hockey</u>
some drunks love hockey

(note: *no As are B = all A are not B =* ∀**x (Ax ⊃ ~Bx)** .)

b) When we are considering a finite domain the universal quantifier is closely related to conjunction, and the existential quantifier to disjunction. For example in a domain of three individuals, **a**, **b**, **c**, ∀**x Ax** is true if and only if **Aa & Ab & Ac** is true, and ∃**x Ax** is true if and only if **Aa v Ab v Ac** is true. (Do you see why this is so?) Turning the universal and existential quantifiers into conjunction and disjunction, and considering a domain of three individuals, show that the conclusion of each of the syllogisms in **a)** becomes a logical consequence of the premises, similarly transformed, in propositional logic..

c) why does this not show that the arguments are valid (though they all are valid), taking all models into account?

2) Below are four *invalid* syllogisms. The conclusions are not logical consequences of the premises. Symbolise each of them in the language of quantifier logic and make a counter-model, in which the premises are true and the conclusion false. The counter-models can all be finite, and in fact very small.

i) all cats are animals
all animals have hearts
anything with a heart is an animal

ii) all felines are cats
some animals are cats
some animals are felines

iii) there is an honest philosopher
no-one honest is happy
some happy person is not a philosopher

iv) some oilers fans are drunks
some drunks love hockey
some oilers fans love hockey

3) The equivalence of $\forall \mathbf{x}\ P\mathbf{x}$ and $\sim\exists \mathbf{x} \sim P\mathbf{x}$, and the equivalence of $\exists \mathbf{x}\ P\mathbf{x}$ and $\sim\forall \mathbf{x} \sim P\mathbf{x}$ are like Morgan's laws from chapters 4 and 5. Use a finite model to show the resemblance, as in question 1(b).

4) Which of the following arguments are logically valid (trust your intuitions, but think) If they are not valid describe a counter-model.

i) There is always an earlier time
There is an earliest time

ii) For every rich man there is an even richer one
Some man is infinitely wealthy

iii) Not all students want to work in the oil industry
all students do not want to work in the oil industry

iv) Not all students want to work in IT
Some students do not want to work in IT (think unicorns)

5) Give a counter-model to show that $\forall \mathbf{x}\, \exists \mathbf{y}\, \mathbf{Rxy}$ is not a logical consequence of $\exists \mathbf{x}\, \forall \mathbf{y}\, \mathbf{Rxy}$. (Note how this is different from the valid argument mentioned in section 3.)

6) Which of these are equivalent according to the rules for prenex form? (Assume that they contain no variables besides those shown.)

(i) $\exists \mathbf{x}\, \forall \mathbf{y} (\mathbf{Ax} \supset \mathbf{By})$ (ii) $\exists \mathbf{x}\, (\mathbf{Ax} \supset \forall \mathbf{y} \mathbf{By})$

(iii) $\forall \mathbf{x}\, \exists \mathbf{y}\, (\mathbf{Rxy} \supset \mathbf{By})$ (iv) $\forall \mathbf{y} (\forall \mathbf{x}\, \mathbf{Ax} \supset \mathbf{By})$

(v) $\exists \mathbf{x}\, \forall \mathbf{y} (\mathbf{Ax}\ \&\ \mathbf{By})$ (vi) $\forall \mathbf{x}\, \exists \mathbf{y}\, (\mathbf{Rxy} \supset \mathbf{By})$

(vii) $\forall \mathbf{x}\, \exists \mathbf{y}\, \mathbf{Rxy}\ \&\ \forall \mathbf{x}\, \exists \mathbf{y}\, \mathbf{Sxy}$ (viii) $\forall \mathbf{x}\, \exists \mathbf{y}\, \forall \mathbf{z}\, \exists \mathbf{w}\, (\mathbf{Rxy}\ \&\ \mathbf{Szw})$

(ix) $\exists \mathbf{x}\, \forall \mathbf{y}\, \exists \mathbf{z}\, ({\sim}\mathbf{Rxy} \vee \mathbf{Dz})$ (x) ${\sim}\forall \mathbf{x}\, \exists \mathbf{y}\, \mathbf{Rxy} \vee \exists \mathbf{y}\, \mathbf{Dy}$

(x) $\exists \mathbf{x}\, \exists \mathbf{y}\, \forall \mathbf{z}\, ({\sim}\mathbf{Ryz} \vee \mathbf{Dx})$ (xi) $\forall \mathbf{x}\, \mathbf{Ax} \supset \forall \mathbf{y} \mathbf{By}$

7) Facts: A princess has a number of suitors. She is not sure how many as she is too busy choosing dresses and gossiping about her friends. One suitor, or perhaps more than one, is an expert rider. There may be an archer among her suitors, though she has not been paying attention. There is also at least one champion swordsman. All of her suitors are rich, and none of them is intelligent. There is one rider who is not a poet. Three poets have written descriptions of her beauty. Moreover, no archers are riders, though some riders are poets. All archers have red hair. No red haired people are rich. All the swordsmen are intelligent, with just one exception.

Question: How many suitors does she have?

8) (a) Which of the following are true in the model below, when **C** is **C**at, **F** is **F**ierce, **D** is **D**ogs, and the arrows represent the relation **L** "Looks at"?

i) ∀**x(Cx** ⊃ **Fx)**

ii) ∀**x** ∃**y Lyx**

iii) ∃**x** ∃**y (Lxy & Lyx)**

iv) ∃**x** ∀**y (Dx & ((Lxy & Cy)** ⊃ **Fy))**

v) ∃**x** ∀**y ((Dx & ((Fy & Cy)** ⊃ **Lxy)))**

vi) ∀**x** ∀**y ((Lxy & Lyx)** ⊃ **(Dx & Cy))**

vii) ∀**x ((Fx& Cx)** ⊃ ∃**y (Dy & ~Lyx))**

viii) ∀**x** ∀**x ((Lxy & Lyx)** ⊃ **((Dx & Cy) v (Dy & Cx)))**

(b) For each of these English sentences say which of the formulas in (a) it is a rough translation of

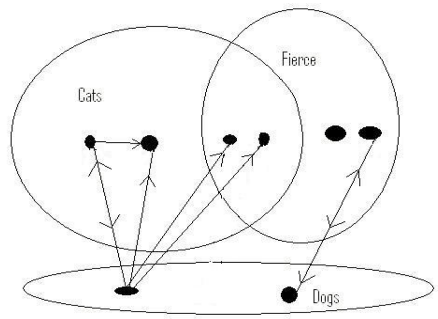
s) when two animals look at each other, one is a dog and the other a cat

t) some dog looks at all fierce cats

u) some dog looks only at fierce cats

v) every fierce cat does not look at some dog

w) Every animal is looked at



(c) Find all individuals that:

(i) are fierce and look at something

(ii) are fierce and look at something that looks at a cat

(iii) are fierce cats and look at something that something looks at

(iv) look at all the dogs that a fierce cat looks at

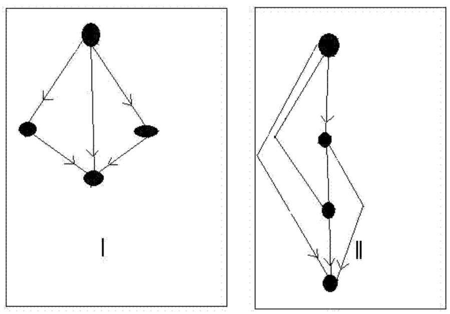
(v) look at something that looks at her cat

(vi) look at all the dogs that a fierce cat looks at

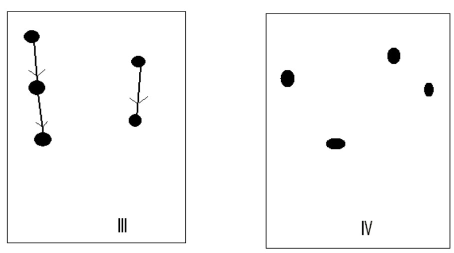
(d) write each of these in the language of quantifier logic.

9) ∀**xAx** ⊃ ∀**y By** is equivalent to ∃**x** ∀**y(Ax** ⊃ **By)** and also to ∀**y** ∃**x (Ax** ⊃ **By)**, converting to prenex form in two different ways. So the two prenex forms are equivalent. Use this fact to show that ∃**x** ∀**y Rxy** |= ∀**y** ∃**x Rxy** . Why does this not lead to a proof of the complementary entailment ∀**y** ∃**x Rxy** |= ∃**x** ∀**y Rxy** also?

10) The arrow diagrams I and II below represent two typical strict partial orderings (antireflexive, antisymmetrical, and transitive — see exercises 5 and 8 of chapter 9.) Taking them as typical, one might think that all strict partial orderings can represent a comparison of the extent to which individuals have some quantity. (Both of these, for example, could be interpreted as "is at least as happy as" or many other comparisons.)



But there are models with partial orderings which are much less easily represented as comparing the individuals that have the relation. Examples are the partial orderings below.



(Check that III and IV are partial orderings.) State an assumption that rules out models such as these. Express it both in English and in quantifier logic.

11) In exercise 10 a strict partial ordering was defined as a relation that is antireflexive, antisymmetric, and transitive. In fact the requirement of antireflexivity is not needed, as any relation that is antisymmetric and transitive is antireflexive. Give an informal argument to show this.

12) Adapt the argument in section six that there is a sentence that is true only in infinite domains, to show that there is an invalid argument whose only countermodels are infinite. That is, there are arguments that are valid if we only consider models with finite domains.

13) The universal and existential quantifiers are not the only quantifiers represented in English or other languages. It may be a historical accident that following on from Aristotle's logic subsequent logical systems have given them a central place. A quantifier that is similar to the universal quantifier is "most". **Mostx Px** is true in a model if and only if more than half of the members of the domain of the model satisfy **P**. But the logic of this quantifier is rather different from that of the universal quantifier. There are logically valid arguments turning on "most". For a trivial example

All As are Bs
Most As are Bs

but on the other hand the following is (perhaps surprisingly) *not* a logically valid argument.

Most As are Bs
Most Bs are Cs
Most As are Cs

(Compare exercise **16** of chapter 10.)

Find a finite model in which the premises of this argument are true but the conclusion is false.

14) Give a counter-model to show that "Most **A**s are **B**s" is not equivalent to "**Most x: Ax ⊃ Bx**".

16) Summation, $\sum$, and product, $\prod$, are variable-binding operators (see sections 1 and 3 of chapter 3, and section 5 of chapter 10.) Give examples

(a) where $\sum_{0<n<10} \prod_{0<m<10} f(n,m)$ is not the same as $\prod_{0<m<10} \sum_{0<n<10} f(n,m)$.

(b) where these are not the same in the system of arithmetic from question **41** of chapter 5.

(c) where $\sum_{0<n<10} \prod_{0<m<10} \sum_{0<r<10} f(n,m,r)$ is not the same as $\sum_{0<n<10} \sum_{0<m<10} \prod_{0<r<10} f(n,m,r)$.

(And more examples where generally three-operator combinations are sensitive to the order of the operators.)

appendix: a system of derivations for quantifier logic

There are many obviously valid arguments involving quantifiers that cannot be captured by the principles of Boolean logic. Valid syllogisms are an example. Besides syllogisms, consider the following:

∀x Px	Everything is pretty
~ ∃x ~ Px	it is not the case that there is something that is not pretty
~∃x Px	Nothing is pretty
∀x ~ Px	Everything is not pretty
∀x (Hx ⊃ Ax)	all horses are animals
∀y (∃x (Hx & Tyx) ⊃ ∃x (Ax & Tyx))	the tail of a horse is the tail of an animal

And there are arguments that are valid, but not obviously so. We need a logical theory to guide us through these. For example

∃x ∀y Rxy	Someone loves everyone
∀y ∃x Rxy	Everyone is loved by someone
∀x (∃y Hxy ⊃ Ax)	all creatures with hearts are animals .
∀x ∀y (Hxy ⊃ Ax)	if one thing is the heart of another then the second thing is an animal

Any model in which all the assumptions of any of these arguments are true will make its conclusion true.

I shall present a system of derivations for the logic of quantifiers. It is similar in spirit to the system of derivations for Boolean logic of chapter seven. It has all the rules of that system plus some additional ones, to handle quantifiers. For simplicity I will not formulate the rules so that when a conclusion is a consequence of some premises one can get the conclusion by a series of steps from the premises. Instead I shall rely on the other alternative way of showing that an argument is valid used in chapter eight: to show that C follows from $\{P_1,\ldots,P_n\}$ we show that $\{P_1,\ldots,P_n,\sim C\}$ is a contradiction.

The Boolean rules, repeated from chapter seven, are:

$$\frac{A \,\&\, B}{A} \qquad \frac{A \,\&\, B}{B} \qquad (\&)$$

A v B
/ \
A B
(v)

A ⊃ B
/ \
~A B
(⊃)

$$\frac{\sim(A \,\&\, B)}{\sim A \vee \sim B} \quad (\sim\&) \qquad \frac{\sim(A \vee B)}{\sim A \,\&\, \sim B} \quad (\sim v)$$

$$\frac{\sim(A \supset B)}{A \,\&\, \sim B} \quad (\sim\supset)$$

A
/ \
B ~B
(EM)

I will not repeat the explanations of these rules. Check back in chapter seven if need be. To them we add four more. The first two describe the interaction of the quantifiers with negation.

$$\frac{\sim\exists n\, A}{\forall n \sim A} \quad (\sim\exists) \qquad \frac{\sim\forall n\, A}{\exists n \sim A} \quad (\sim\forall)$$

n here can be any variable, and A any sentence, open or closed. Notice the similarity of these rules to (~v) and (~&).

The next two rules describe the relation between sentences with quantifiers and sentences without. They allow us to add and remove quantifiers.

$$\frac{\exists m\, A}{A[n]} \quad (\exists)$$

what this means is that if you have an existential quantification, where the ∃ is at the beginning of the sentence — m represents any variable at all — you can derive the sentence got by stripping the ∃ off the beginning of the sentence and replacing the variable that it bound with a name, which to state the rule I am calling n. It must be a new name, not one occurring in A or anywhere else in the derivation so far. Two examples of the ∃ rule:

∃x (Ax & Bx)	∃z ∀y (Rzyn & By)
An & Bn	**∀y (Rzyn & By)**

In the second of these we could not have derived **∀y (Rzy & By)** . because that would have meant substituting **n** for the variable **z**, which is forbidden since **n** already occurs in the premise.

The idea behind the rule is that if an existential quantification is true then there is something in the domain that makes it true. But we do not want to beg any questions about which object in the domain it is, so until the derivation forces us we make it completely distinct from everything else that has been mentioned.

The other rule is

$$\frac{\forall m\ A}{A[n]} \qquad (\forall)$$

where n can be any name that has occurred in the derivation, or a new one. Moreover, when — by use of the rule (∃) – a new name is introduced later in the derivation it also may be substituted for the variable m.

Two examples of the ∀ rule (the second example uses both rules, so we can see them working together).

```
 An                                    ∀x ∃y Rxsy
∀x (Ax ⊃ Rnx)                           ∃y Rmsy          (∀)
 An ⊃ Rnn            (∀)                 ∃y Rssy          (∀)
   /    \                                 Rmyt            (∃)
~An     Rnn
 X
```

Note how in the first of these the fact that **n** already appears in the premises requires us to use it as a substitution for **x**. Note how in the second of these the fact that we have used the (∃) rule requires us to go back and make a substitution using the new name that the (∃) rule has introduced.

The idea behind this rule is that if a universal quantification is true then everything in the domain has the attribute in question. So we should attribute it to everything that we can mention.

These are all the rules. They are fairly simple, but they are very powerful. The interaction of the (∃) and (∀) rules means that a derivation can keep on going, looping round and making new substitutions. It need never stop. This is important as some of the models that make quantified sentences true are infinite.

Some examples of the rules in action. It will help if I state the general strategy that the combination of rules is meant to fit. We prove that a conclusion follows from some premises by showing that the premises plus the negation of the conclusion lead to a contradiction. So we are heading for contradictions. We want to make branches close. We do this by stripping off quantifiers and trying to find simple sentences that contradict other simple sentences. So the slogan is: break up complex sentences and make as many small sentences as you can, hoping that some will contradict others. Of course you will generally have to use more specific strategies than that, if you are not to blunder around aimlessly. But they will fit into this general slogan. After each derivation I will give an informal version in English fitting the same strategy.

$\forall$**x Px** ⊨ $\forall$**y Py**

(1)	$\forall$**x Px**	(P, Premise)
(2)	**~$\forall$y Py**	(P, Premise) (note: this is the negation of the conclusion, as we are going to show that $\forall$**x Px** and ~$\forall$**y Py** are inconsistent.)
(3)	$\exists$**y ~Py**	(2, ~$\forall$)
	~Pn	(3, $\exists$)
.	**Pn**	(1,$\forall$)
	X	

"Everything is **P**erfect, so assume that everything is not perfect, and we get a contradiction, since any arbitrary thing will be both perfect and not perfect. So however you express it everything is perfect.)"

$\forall$**x ~ Px** ⊨ ~$\exists$x**P**x

(1)	$\forall$**x ~Px**	(P)
(2)	$\exists$x**P**x	(P)
(3)	**Pn**	(2, $\exists$)
(4)	**~Pn**	(1, $\forall$)
	X	

"Everything is not **P**erfect. (Each thing fails to be perfect.) Assume that there was one perfect thing. Call it **n**. It would be perfect, but since everything is not perfect this is a contradiction"

$\forall$**x Px** ⊨ **~$\exists$x ~Px**

(1)	$\forall$**x Px**	(P)
(2	**~~$\exists$x ~ Px**	(P)
(3)	$\exists$**x ~Px**	(2, Shortcut)
(4)	**~Pn**	(3, $\exists$x)
(5)	**Pn**	(1, $\forall$)
	X	

"Everything is perfect. Assume that there is not even one thing that fails to be perfect. Call it n. It will be both perfect and not perfect. So there cannot be any such thing."

Here, and elsewhere below I use a shortcut, marked Shortcut. (See line 3 above.)

Shortcut: if you have a sentence S and S |= T can be shown using the rules for Boolean derivations in chapter 7, then derive T in the same branch as S.

Of course before you can make a complete derivation using this shortcut rule you have to have shown using the Boolean rules that S |= T.

∀x (Hx ⊃ Ax) |= **∀y (∃x (Hx & Tyx) ⊃ ∃z (Az & Tyz))**

(1)	**∀x (Hx ⊃ Ax)**	(P)
(2)	**~∀y (∃x (Hx & Tyx) ⊃ ∃z(Az & Tyx))**	(P)
(3)	**∃y ~(∃x (Hx & Tyx) ⊃ ∃z(Az & Tyz))**	(2: ~∀x)
(4)	**~(∃x (Hx & Ttx) ⊃ ∃z(Az & Ttz))**	(3: ∀)
(5)	**∃x (Hx & Ttx) & ~∃z(Az & Ttz)**	(4: ~ ⊃)
(6)	**∃x (Hx & Ttx)**	(5: &)
(7)	**Hh & Tth**	(6: ∃)
(8)	**Hh**	(7: &)
(9)	**Tth**	(7: &)
(10)	**~∃z(Az & Ttz)**	(5: &)
(11)	**∀z ~(Az & Ttz)**	(10: ∀)
(12)	**~(Ah & Tth)**	(10: ∀)
(13)	**~Ah v ~Tth**	(12: ~&)

/ \

(10)	**~Ah**	**~Tth**	(v)
(11)	**Ah** (1:∀, shortcut)	X	
(12)	X		

"All horses are animals. So assume that some **t**ail is a tail of some horse **h**. It will also be an animal tail. Assuming that there are horsetails and there are animal tails that are not horsetails will be contradictory: given our assumption about horses and animals. It will either contradict our assumption that **h** is a horse or our assumption that **t** is the tail of **h**."

∃x ∀y Lxy |= **∀y ∃x Lxy**

(1)	**∃x ∀y Rxy**	(P)
(2)	**~∀y ∃x Rxy**	(P)
(3)	**∀y Rly**	(1, ∃)
(4)	**∃y ~∃x Rxy**	(2, ~∀)
(5)	**~∃x Rxl**	(4, ∃)

(6)	**∀x ~Rxl**	(5, ~∃)
(7)	**~Rlm**	(6, ∀)
(8)	**Rlm**	(3, ∀)
	X	(7, 8)

"Someone loves everyone. To see that this means that everyone is loved by someone, assume the contrary. Call the guy who loves everyone **l**overboy. **l** loves any person you choose. But the contrary assumption is that there is an unloved person **u**. However **l** loves **u**, so this cannot be."

∀x (∃y Hxy ⊃ Ax) ⊨ **∀x ∀y (Hxy ⊃ Ax)**

(1)	**∀x (∃y Hxy ⊃ Ax)**	(P)
(2)	**~∀x ∀y (Hxy ⊃ Ax)**	(P)
(3)	**∃x ~ ∀y (Hxy ⊃ Ax)**	(2,~∃)
(4)	**~∀y (Hny ⊃ Ax)**	(3, ∃)
(5)	**∃y ~(Hny ⊃ Ax)**	(4, ~∀)
(6)	**~(Hnm ⊃ Am)**	(5, ∃)
(7)	**Hnm & ~Am**	(6,~ &)
(8)	**Hnm**	(7, &)
(9)	**~Am**	(7, &)
(10)	**∃y Hxm ⊃ Am**	(1, ∀)
(11)	~**∃y Hxm**	(shortcut, 9, 10)
(12)	**∀y ~Hxm**	(11, ∃)
(13)	**~Hnm**	(12, ∀)
	X	(13,8)

"If you have even one enemy then you will be alert to dangers. So whenever one person has an **e**nemy they are alert. Assume the contrary and you will find that **p** has **e** as an enemy and also does not."

These examples suggest what is in fact the case, that the rules we have just seen allow a wide variety of derivations, and some of them are very subtle. Each time a rule is applied it is a mechanical business to verify which rule has been applied and that it has been applied correctly. But there is no mechanical way of testing whether one sentence follows from another or whether a set of sentences is consistent. In fact, many famous unsolved mathematical problems amount to asking whether we have logical consequence, consistency, or inconsistency. Surprisingly, it can be proven that when one sentence of quantifier logic is a logical consequence of another, then there is a derivation showing this. (The result is Gödel's completeness theorem for quantifier logic of 1929, not to be confused with his even more famous incompleteness theorem for arithmetic of 1931.) In theory, if one sentence is a logical consequence of another then by following out all possible lines of derivation we will eventually hit on a successful one. But it may take longer than any human life, and if the one sentence is not a logical consequence of the other then we can try all these derivations for the lifetime of the universe and be no wiser at the end. This fits with the points that I illustrated at the end of chapter 8, that "checking whether a derivation is correct is easy, knowing whether one formula follows from another is harder, and knowing whether one formula does *not* follow from another is hardest of all." Quantifier logic is complex enough that we can begin to illustrate these points in more detail. But logic can get much more complicated than that.

Given that quantifier logic is complicated and there is no automatic strategy for making derivations with it, you may wonder why we bother with it rather than with informal alternatives like those I have given after each derivation. There are two reasons. The first is that there are many arguments which we cannot capture with the informal versions, or that we will find too confusing. Often they involve rich quantifier prefixes. The second reason is that the informal reasoning uses names in a way that can easily go wrong. For example the "loverboy" argument above is very similar to the following: "Everyone is loved by somebody. So if we choose an arbitrary person **p** we can find a person **l** who loves them. But **p** was chosen arbitrarily — it could be anyone — so we always have that **l** loves **p**. So there is at least one person who loves everyone." No doubt this argument feels wrong to you, but you are unlikely to be able to say what is actually wrong about it. And anything you can say will probably fail when there are many quantifiers or the topic is unfamiliar and you do not have familiar truths to steer by.

This leaves us with a dilemma. Either we use the formal logic of quantifiers and have to concentrate to keep the details straight, or we argue informally and are likely to fall into fallacies with complicated or unfamiliar material. A wise combination of the two is probably the best response. Another response is to work on variations on spoken language that can be understood in the ways that we readily learn and communicate verbally, but which can be given clear rules which are less prone to ambiguity. There are many experiments along these lines in this book.

index

This is a list of topics by chapter sections rather than by pages. It should be stable under mild editing of the text. I have underlined the sections where a term is explained or defined. “n:m” means “chapter n section m”. I have only listed exercises when they go further than the chapters.

⊃ 2:6 2:7, 3:4, 4:2

∃ 9:4

~ 3:2, 4:2

⊨ 7:2

∀ 9:4

almost Boolean arguments 11:3
argument 1:1, 6:1, 6:4, sound argument 6:4, valid argument 6:4
argument by contradiction 8:1, 8:2
arrow diagram 1:11, 2:9
assumption see premise
atomic proposition 1:4, 5:1, 5:2
attribute 1:4, 1:5

Boogle (Boolean search on the Internet) 5:9
Boolean connectives 2:2, 5:2
Boolean search 3:2, on a library database 3:6, on the internet 3:6, 3.7, 5:9
branch of a derivation 7:1, 7:2
branching argument 8:3
branching searches 2:4

C-derivation 7:5
central connective 5:4
closed branch 4:2, 7:2
complement 2:2
completeness 6:4, of propositional logic, of quantifier logic 11: appendix
conditional (⊃) 2:6, 2:7, 3:4, 4:2, consequence for 6:7
conditional argument 8:3
contraries vs contradictories 7: exercises 7:4, 10, 11
conditional rule 7:1
conjunction (**&**) 1:6, 2:5, 3:2, 4:2, 6:7
conjunction rule 7:1,
conjunctive argument 8:3
conjunction/disjunction confusions 2:5, 7: exercise 12
conjunctive normal form 5: exercise 35
consequence see logical consequence
consistent propositions 8:4, 7:5
contradiction 5:6, 7:5
contraposition 6:7
counter model 6:3 , 11:1

database 1:3
deduction 1:1, 6:1, 6:4
de Morgan's laws 5:6, 5:7
derivation 7:1, 7:2, 7:3, 8:1
diagrams in reasoning 1:3
difficulty of judging correctness, consequence, and non-consequence 8:5, of quantifier logic 9:2
direct vs indirect argument 8:2, 8:3, 8:5, in quantifier logic 10:4
disjunction (**v**) 1:6, 3:2, 4:2, 6:5, 6:7 inclusive and exclusive 1:6, 5:2
disjunction rule 7:1,
disjunctive argument 8:3
disjunctive normal form 5:8
domain of a database or model 1:3, empty 11:2, finite 11:3, infinite 11:6

eclipsky 2: exercise 24
elimination of alternatives 6:6
equivalent propositions 5:6, 6:1
Euclidean geometry 11:3
excluded middle (EM) 7:3
existential quantifier (∃) 9:4

formula 5:3

hidden information games 6:2

if see conditional
independent propositions 7:5, 8:4
indexed tables 1: exercise 27, 2:9
individual 1:3
indirect argument see direct argument
inside-out 5:4, 9:7, 10:6, 10:9
intersection 2:2
iterated searches 3:8

ladder relation 10:7
library databases 3:6
linear vs branching argument 8:2
logical consequence 1:1, 6:1, 6:6, distinguished from reasonable argument 6:4 , and derivations 7:4, many ways of showing 7.5
logical independence 7:5
logically valid 6:1, 6:8

material conditional see conditional
mathematical thinking 2:3, 2: exercise 14, 3:1, 3:3, 3:8, 5:4, 7:2, 9:7, 10:6
model 1:3, 4:1, 5:6, 11;5, constructing 4:2
modus ponens 6:6, 6:7, 8:2
modus tollens 6:6, 6:7, 8:2
molecular proposition 5:1
most, logic of 10: exercise 16, 11: exercise 11

negation (**~**) 3:2, 4:2
negation rules 7:1
object and attribute table 1:3

or see disjunction
outside-in see inside-out

parallel searches 2:4
partial ordering 11: exercises 10, 11
passive verbs 10:5
premise 1:1, 6:1
prenex form 11:6
probability 5: exercise 41
problem-solving 1;1, 7:5
proof 7:2
pronoun 3:1, 8:3
proposition 4:1
propositional logic 5:1, 5:3

quantifier 9:1, context sensitivity of 9:3,10:4, multiple 10:1, 10:3 (double), 10:6 (triple), 10:8 (quadruple), 10:10 (sextuple), prefix 10:6, matrix 10:6, 10:10, scope 10:1, 11;2, illustrated graphically 10:3, 10:7, 8, exercises 5, 13, and names 10:2,11:4

query 1:4, 3:1

recursive definition 5:appendix
reductio ad absurdum   see argument by contradiction
relation 1:7, 2:9, antisymmetric 10: exercise 7, asymmetric 1:7, 10: exercise 7, reflexive 10: exercise 7, representations of 1:7, 1:10, 1:11, 2:9, symmetric 10: exercise 7, transitive 10: exercise 7
relational grid 1:7
riddle 1:4

scope 2:1, 3:3, 3:5, 10:1, 11:2
search command 1:4
search in series, parallel, branching 2:4, iterated 3:8
search on the Internet 3:6, 3:7
search tree 4:2, 6:1
semantic and syntactic 6:3, 7:2
sentence 4:1, 5:1
soundness 6:4
syllogism 11:3
syntactic, see semantic

tautology 5:6, 7:4
tense 10:5
transitive relation 10: exercises 7, 14, 11: exercises 10, 11
transitivity of the conditional (if) 8:1, 8:2
translating between English and quantifier logic 9:7, 9:8
truth function 5:2
truth table 4:3, 5:2
truth assignment 5:2
truth value 1:5, searching for 1:5, of complex propositions 5:5
turnstile (⊨) 7:2

union 2:2

universal quantifier (∀) 9:4

valid argument see logically valid
variable 3:1, 9:6, 11:3, 11: appendix
variable-binding operator 3:1, 3:3, 9:1, 9:3 , 10:5, 11:exercise 16
Venn diagram 2:2

well-formed formula 3:5, 5:3, 5: appendix
wheel relation 10:7
width of a query 4:5

planning a course using this book

The eleven chapters of this book have material on a variety of themes, aimed at covering the standard material of introductory symbolic logic while giving a more discursive course, engaging a greater variety of interests. You have to decide which material and which themes fit into the course that you want to give. So you must decide which chapters, and which sections, you will discuss and have the students read. Your expectations of the interests and capacities of the particular students you will be teaching are central here. The exercises for each chapter roughly mimic the order of the topics in the chapter, but if you are only covering some of these topics you must tell the students which exercises are relevant. Since some exercises for some chapters anticipate, prepare for, or refer back to topics explicitly covered in other chapters, you could look through the exercises and consider assigning some from chapters or sections of chapters that you are not explicitly covering.

In my opinion it gives students a better grounding in modern logic to cover all chapters, though, even at the price of going into some topics in less detail. Logic teachers sometimes think that they can get a one term class with no prior knowledge to manage propositional logic intensely and in detail in one term while throwing in some quantification theory at the end. I am sure this is an illusion. Whatever their results in the exam, very few of the students will remember any details six months later. And for a majority of the students the material on quantifiers will be an irritating mystery, arriving just when the burdens of the term were getting heaviest in all courses. But this is a shame: one of the main benefits of a logic course is to contrast how English and logic handle quantification. This, besides the intrinsic interest to students of issues about search, is a reason for introducing formal queries as variable binding operators early on. They prepare the ground so that when quantifiers arrive weeks later they are already half familiar.

Instructors who have given a standard logic course may worry that chapter five covers too much for one chapter. But much of the material has been covered in earlier chapters in slightly different form. (Chapter 4, which is short on formal details, makes the conceptual transition from search to deduction. It is important to get students to see how this works.) In my experience students who have been prepared by the earlier material treat most of the content of this chapter as familiar and obvious.

Throughout the text there are remarks and questions in blue, marked with >> and formatted in a distinctive way. These are meant to suggest questions to think about when reading, or to prompt class discussions. I have aimed to make the exposition clear enough that the instructor does not have to spend much class time repeating it, and can focus on discussion, problem solving, and exercises. Students should remember, though, that the instructor will not have time to discuss every interesting question, and some topics are far from the instructor's agenda for the course. If a course meets twice a week one class could be discussion and one could be problems and exercises. Some of the exercises are easy and some are hard. There are more fairly easy exercises in the early chapters, to convince students that they can handle this material. For this reason I would discourage instructors from giving difficult tests early in the course. This is the kind of material where low self-confidence can be self-fulfilling. I have always started gently, and by the time we get to more challenging material students' confidence is high and to their delight they are operating at a remarkable level. In this connection there are remarks and exercises throughout that are meant to give math-phobic students a sense of what it is to think mathematically. One of the incidental functions of a logic course is often to persuade students that a lack of facility with numbers need not exclude them from formal methods in general.

About the exercises. There are continuities between the exercises in the different chapters, for example between chapter 1 and chapter 11. One aim of this continuity is to keep central issues and techniques live in students minds throughout the course. Conversely, some exercises anticipate ideas that will appear officially in later chapters. These can be tackled with common sense and general intelligence, and are designed to demonstrate the underlying familiarity of the ideas and to begin mental preparation for the later exposition.

I have not provided answers to the exercises (those that have definite answers). The class will appreciate it if you provide them. But the work of preparing and checking them will naturally come after your decisions about which sections to cover and hence which questions to assign. If you do provide answers it is important that they be accurate, as small slips can undermine the faith in you of stronger students and undermine the confidence of weaker students. (They do not know that you have done them in a hurry the night before the first class.)

There is material in the book for a number of distinct courses, with different emphases on the formal and the conceptual side of the subject. You should read through it all and decide which sections, and consequently which exercises, you are going to require, ignore, or suggest as supplementary reading. As part of deciding what character of course you are going to give you might look at the >> remarks and think which ones you want to be prepared to have a class discussion about. There are too many for you to prompt a discussion of all of them, so your encouragement can be influential. Of course, it is of the nature of discussion that it goes where it goes.

When writing the book I had an eleven week term in mind, so there are eleven chapters. (Often there is an extra class at the beginning of term or an extra class at the end.) Some courses will have twelve or more weeks. If this is the case, or if the course concentrates on some topics by skipping others, some chapters are designed to break into two. The chapter text and the exercises can be easily divided over two weeks. This is true of chapters 5, 7, and 8. And there are the appendices to chapters 5 and 11. The first of these could occupy a week if well-formedness and the play between syntax and semantics is something you want to discuss. And the second could occupy any amount of time, depending on how much emphasis you want to put on it.

I have sometimes included term papers in the final grade. Sometimes these have been compulsory and sometimes optional. (But in the second case grading them fairly against test results is difficult.) The C exercises and the >> remarks are sources of topics. Term papers make the course more like the courses that many students are used to, and they restore a balance between reflective and quick-witted mentalities.

When I have taught this material I have given regular tests, and counted them towards the final grade. In recent years I have not given them in class time but let students do them on the internet. As protection against having one's math major sister take the test, I have also had a final exam, with the proviso that the final grade is not an average of the tests and the exam, but is rather a non-linear function which rewards improvement but also penalizes large downward disparity. If a student's test average is A and their exam is B+ then they get an A-, but if the test average is A+ and the exam is C then the result is a C. (If you have got your sister to take the online tests for you then you will not learn the stuff, and you are setting yourself up for a C.) Of course this scheme should be announced to the class at the beginning, and it is best to put it in the written syllabus, to protect against later professions of ignorance.

You may edit the text to insert or remove material, as long as you acknowledge your debt to this work, and as long as you do not make a profit. (So no submitting it to a publisher!) One area where additions would be helpful is with diagrams, especially animated diagrams. I make some suggestions about these in the appendix that contains the full Creative Commons statement.

Good luck. Give a course that both you and the students enjoy.

copyright and changes

You may use this material in any way that fits your teaching. You may add to it, and if you do I encourage you to post and circulate the result so that others can use it. By using it you commit to the conditions specified more precisely in the Creative Commons wording that follows.

Promising areas for additions are:

– *exercises*: if your emphasis in teaching is different from mine you may want additional exercises reinforcing the points that are central for you.

– *answers*: I have not supplied solutions to the exercises. In any case they should be circulated separately to encourage students to complete their answers before checking. I think it is important to avoid even slight mistakes in answers, as the confidence of weaker students in themselves is adversely acted when they think they have got an answer wrong, and the confidence of stronger students in the instructor is adversely affected when they find even small errors. (Although it is a sign of a good student that she finds ways of improving what you have done.)

– *diagrams*. Some people learn best from pictures, and with an online text we do not have to worry about the space they take. We also do not have to worry about printing expensive colours. So the right diagram in the right place is always helpful.

– *animated diagrams*. One advantage of an electronic text is the possibility of moving diagrams. These are better than still diagrams for explaining many things. (See the diagrams in Wikipedia found in the articles on Maxwell's equations, Quicksort [particularly relevant], Solar system, Wave equation, Watt's linkage. Some of these articles are fairly technical: the content of the text does not matter; what is important is what can be done with a moving diagram.) In the version of these notes that I used for several classes I inserted fairly primitive moving diagrams in the text.

These were powered by Flash, which is generally avoided these days. So I invite someone with the needed skills to make more and better.

Places where animated diagrams would fit well:

- ch 2 end of sec 4, discussing the difference between filters in parallel and filters in sequence. The idea is to see the names of the individuals move through the filters or be blocked by them.

-ch 2 end of sec 8. The idea would be to see the arrow diagrams for two relations turn into an arrow diagram for one relation, the result of combining the two 2-place relations In the indicated way.

– Chapter 3, middle of section 1. The idea would be to show the variable, perhaps represented by a little machine, going through the domain and picking up things fitting the criterion, then doing something with them. It could count them, or collect them in one place, or look for an exception falsifying a universal quantification, or for a single case verifying an existential quantification. Ideally it would do all of these, with the student able to choose which, and as a result the student would see that they are all processes that are well symbolized by a variable binding operator.

– Chapter 5, end of section 7. The de Morgan flip: See the static diagram for the basic idea and then make individuals pass through the filters in their various configurations.

These could be inserted in a version of the text, or put on a separate web page. It would be in the spirit of the project if these too were made generally available.