Chapter 3
Logic

Greg Restall[1]

Department of Philosophy
Macquarie University
Australia

Greg.Restall@mq.edu.au

http://www.phil.mq.edu.au/staff/grestall/

Introduction

Logic is the study of good reasoning. It's not the study of reasoning as it actually occurs, because people can often reason *badly*. Instead, in logic, we study what makes good reasoning *good*. The *logic* of good reasoning is the kind of connection between the *premises* from which we reason and the conclusions at which we arrive. Logic is a *normative* discipline: it aims to elucidate how we *ought* to reason. Reasoning is at the heart of philosophy, so *logic* has always been a central concern for philosophers.

This chapter is not a comprehensive introduction to formal logic. It will not teach you how to use the *tools* and *techniques* that have been so important to the discipline in the last century. For *that* you need a textbook, the time and patience to work through it, and preferably an instructor to help.[2] The aim of this chapter is to *situate* the field of logic. We will examine some of the core ideas of formal logic, as it has developed in the past century, we will spend a significant amount of time showing connections between logic and other areas of philosophy, and most importantly, we will show how the philosophy of logic contains many open questions and areas of continued research and investigation. Logic is a living field, and a great deal of interesting and important work in that field is being done today.

This chapter has five major sections: The first, *Validity* introduces and demarcates the topic which will be the focus of our investigation: *logical validity*, or in other words, *deductive logical consequence*. The most fruitful work in logic in the 20th Century has been informed by work in the formal sciences, mathematics throughout the century, and computer science in the second half of

the century and into the 21ˢᵗ. So, the nature of *Formalism* will take up our second section. I will explain why logic as it has been studied is a *formal* discipline, and what that might mean for its techniques and applications. Work presenting formal logical systems generally proceeds in one of two ways, commonly called *syntax* and *semantics*, though I will explain below why I think that these terms are jointly a misnomer for the distinction between *Interpretations* on the one hand *Proofs* on the other. There is no doubt that *Interpretations* and *Proofs* play an exceedingly important role in logic. The relationships between the two general modes of presenting a logical system can be presented in *soundness* and *completeness* results, which are vital to logic as a discipline. Finally, a section on *Directions* will sketch where the material presented here can lead, both into open issues in logic and its interpretation, and into other areas of philosophy.

By the end of this chapter, I hope to have convinced you that logic is a vital discipline in both senses of this word — yes, it is *important* to philosophy, mathematics and to theories of computation and of language — but just as importantly, it is *alive*. The insights of logicians, from Boole, to Frege, Russell, Hilbert, Gödel, Gentzen, Tarski to those working in the present, are alive today, and they continue to inform and enrich our understanding.

We will start our investigation, then, by looking at the subject matter logicians study: logical consequence, or valid argument.

Validity

An *argument*, for philosophers, is a unit of reasoning: it is the move from premises to a conclusion. Here are some arguments that might be familiar to you. [3]

> **Nothing causes itself.**
> **There are no infinite regresses of causes.**
> **Therefore, there is an uncaused cause.**
>
> **It is a greater thing to exist both in the understanding and in reality, rather than in the understanding alone.**
> **That which is greater than all exists in the understanding.**
> **Therefore, that which is greater than all must exist not only in the understanding but in reality.**

The first argument here has two premises (the first states that nothing causes itself, the second that there is no regress of causes) and one conclusion (stating that there is a cause which is not caused by something else). Arguments may have many different virtues and vices. Some arguments are *convincing* and

others are not. Some arguments are *understandable* and others are not. Some arguments are *surprising* and others are not. None of these virtues are the prime concern in the discipline of logic. They bear not only on the argument itself and the connections between premises and conclusions, but also on important features of the *hearers* of the argument. Issues like these are very important, but they are not *logic* as it is currently conceived. The central virtue of an argument, as far as *logic* is concerned, is the virtue of *validity.* To state things rather crudely, an argument is valid just when the conclusion *follows from* the premises: that is, in stating the premises, the conclusion follows inexorably from them.

This "definition" of the term is no more than a hint. It does not tell us very much about how you might go about constructing valid arguments, and nor does it tell you how you might convince yourself (or convince others) that an argument is not valid. To do that, we need to fill out that hint in some way. *One* way to fill out the hint, which has gained widespread acceptance, is to define the concept of validity like this:

> **An argument is valid if and only if in *every* circumstance in which the premises are true, the conclusion is true too.**

This way of understanding validity clearly has *something* to do with the initial hint. If we have an argument whose premises are true in some circumstance, but whose conclusion is not true in that circumstance, then in an important sense, the conclusion tells you *more* than what is stated in the premises. On the other hand, if there is no such circumstance, the conclusion indeed does follow inexorably from the premises. No matter what possible way things are like, if the premises are true, so is the conclusion: without any exception whatsoever.[4]

This understanding of the concept of validity also at points you towards solutions to the two questions we asked. To show that an argument is *invalid* you must find a *circumstance* in which the premises are true and the conclusion is not. To convince yourself that an argument is *valid* you can do one of two things: you can convince yourself that there is no such circumstance, or you can endeavour to understand some basic arguments which preserve truth in all circumstances, and then string these basic arguments together to spell out in detail the larger argument. These two techniques for demonstrating validity will form the next two parts of this chapter. *Interpretations* provide one technique for understanding what counts as a circumstance, and techniques in logic from model theory will give techniques for constructing interpretations in which

demonstrate (and hopefully contribute to an *explanation* of) the invalidity of invalid arguments. *Proofs* are techniques to demonstrate validity of longer arguments in terms of the validity of small steps that are indubitably valid. We will see examples of both kinds of techniques in this chapter. Before this, we need to do a little more work to explain the notion of validity and its neighbours.

Validity is an all-or-nothing thing. It doesn't come in grades or shades. If you have an argument and there is just *one* unlikely circumstance in which the premises are true and the conclusion is not, the argument is *invalid*. Consider the cosmological argument, inferring the existence of an uncaused cause from the premises that nothing causes itself, and there are no infinite regresses of causes: one way to point out the invalidity of the argument as it stands is to note that a circumstance in which there are *no* causes or effects renders the premises true and the conclusion false. Then discussion about the argument can continue. We can either add the claim that something causes something else as a new premise, or we can attempt to argue that this hypothetical[5] circumstance is somehow impermissible. Both, of course, are acceptable ways to proceed: and taking either path goes some way to explain the virtues and vices of the argument and different ways we could extend or repair it.

The conclusion of a valid argument need not actually be *true*. Validity is a *conditional* concept: it is like *fragility*. Something is fragile when *if* you drop it on a hard surface it breaks. A fragile object need not be broken if it is never dropped. Similarly, an argument is valid if and only if in every circumstance where the premises are true, so is the conclusion. The conclusion need to be true, unless *this* circumstance is one in which the premises are in fact true. Another virtue of arguments, which we call *soundness*, obtains when these "activating conditions" obtain.

> **An argument is sound if and only if it is valid, and in addition, the premises are all true.**

Of course, many arguments have virtues without being valid or sound. For example, the argument

> **Christine is the mother of a five-month old son.**
> **Therefore, Christine is not getting much sleep.**

is reasonable, in the sense that we would not be making a terrible *mistake* of inferring the conclusion on the basis of the premises. However, the argument is not valid. There are circumstances in which the premise is true, but the

conclusion is not. Christine might not be looking after her son, or her son *could* be unreasonably easy to care for. However, such circumstances are out of the ordinary. This motivates another definition of a virtue of arguments:[6]

> **An argument is *strong* if and only if in *normal* circumstances in which the premises are true, the conclusion is true too.**

This definition picks out an interesting relationship, which we can use to understand ways in which arguments are good or bad. However this kind of *strength*, commonly called *inductive strength* in contrast to *deductive* validity, will not be the focus of our chapter. By far the bulk of the work in logic in the 20th Century is in studying the notion of *validity*, and in particular, in using *formal* techniques to study it. So, we will turn to this new concept. What is it that makes formal logic *formal*?

Formalism

The *form* of an argument is its shape or its structure. For example, the following two arguments share some important structural features:

> **If the dog ran away, then the gate was not closed.**
> **The gate was closed.**
> **So, the dog didn't run away.**
>
> **If your actions are predetermined, then you are not free.**
> **You are free.**
> **Therefore, your actions are not predetermined.**

You can see that both of these arguments are valid, and they both are valid for the same kind of reason. One way of seeing that they are both valid is to see that they both have the following *form*:

> **If *p*, then not *q*.**
> **q.**
> **Therefore, not *p*.**

We get the first argument by selecting **the dog ran away** for *p* and **the gate was closed** for *q*. We get the second argument by selecting **your actions are predetermined** for *p* and **you are free** for *q*. Whatever you choose for *p* and *q*, the resulting argument will turn out to be valid: if the premises are both true, then *p* cannot be true, because if it *were* true, then since *p* implies the falsity of *q*

we would have contradicted ourselves by agreeing that **q** is true. So **p** isn't true after all. As a result, we say that the argument form is also *valid*.

> An *argument form* is *valid* if and only if whenever you substitute statements for the letters in the argument form, you get a valid argument as a result.

The result of substituting statements for letters in an argument form is called an *instance* of the form. As a result, we could have said that an argument form is valid if and only if all of its instances are valid. Here is an example *invalid* argument form.

> **If *p*, then *q*.**
> **q.**
> **Therefore, *p*.**

This looks a lot like the previous argument form, but it not as good: it has many invalid instances. Here is one of them.

> **If it's a Tuesday, then it's a weekday.**
> **It's a weekday.**
> **Therefore, it's Tuesday.**

This is an invalid argument, because there are plenty of circumstances in which the premises are both true, but the conclusion is not. (Try Wednesday.)

We shouldn't conclude that *every* instance of this form is invalid. An invalid argument form can have valid instances. Here is one:

> **If it's a Tuesday, then it's a Tuesday.**
> **It's a Tuesday.**
> **Therefore, it's Tuesday.**

This is not a particularly informative or helpful argument, but using the definitions before us, it is most certainly a valid one. (This will hopefully make it clear, if it wasn't already, that validity is not the only virtue an argument can have.) You might object that the argument doesn't have the form requested. After all, it has the form

> **If *p*, then *p*.**
> **p.**
> **Therefore, *p*.**

Which is valid (though not informative, at least in most instances). And that is correct. An argument can be an instance of different forms. This argument is an instance of the first form by selecting **it's Tuesday** for both *p* and *q*, it is an instance of the second form by selecting **it's Tuesday** for *p*.[7]

Formal logic is the study of the validity of argument forms. Developing formal logic, then, requires giving an account of the kinds of argument forms we wish to consider. Different choices of argument forms correspond to different choices of what you wish to include and what you wish to ignore when you consider validity. Think of the shape of an argument as determined by the degree of *ignorance* you wish to exhibit when looking at arguments. In the examples considered so far, we ignore everything except for **if … then …** and **not**. One reason for this is that we can say interesting things about validity with respect to arguments formulated with these kinds of words. Another reason is that these words are an important sense, *topic neutral*. The word **not** is not about anything in particular, in the way that the word **cabbage** is about a particular kind of vegetable. We can use the word **not** when talking about anything at all, and we do not introduce *new* subject matter. Whenever I *use* the word **cabbage** I talk about vegetables.[8]

Another way to think about the choice of argument forms is to think of it as the construction of a particular language that contains only words for particular concepts we take to study, and letters or variables for the rest. This is the construction of a *formal language*. Sometimes this construction of a particular formal language comes with high philosophical expectations. An important case is Frege's *Begriffschrift* (Frege: 1972, 1984). Of course, commitment to the importance of formal logic need have no such hegemony for formal languages. Formalism may be important in gaining insight into rich natural languages, without ever endeavouring to *replace* messy natural languages by precise formal languages.

In this chapter, I will give an account of two different choices of formal languages: a smaller one (the language of *propositional logic*) and a larger one (the language *predicate logic*). Let's start with the language of prepositional logic.

Propositional logic concerns itself with propositions or statements, and the ways that we combine statements to form other statements. The words or concepts that we use to combine or modify statements are called *operators*. We have already seen two: **if … then …** combines two statements to form another, which we call a *conditional* statement. The statement **if *p* then *q*** is the conditional with *p* as the *antecedent* and *q* as the *consequent*. On the other hand, not does

not combine statements, it modifies one statement. If **p** is a statement, then so is **not-p**, and we call this the *negation* of **p**.

In formal languages, it becomes convenient to use a shorthand form of writing to represent these forms of propositions. Instead of **if p then q**, logicians write **p ⊃ q**. Instead of **not-p**, you can write **~p**. The use of symbols is may be frightening or unfamiliar, but there is nothing special in it. It is merely a shorthand convenience. It is much easier (when you get used to it) to understand

> **(p ⊃ q) ⊃ (~q ⊃ ~p)**

Than it is to understand

> If (if the first then the second) then (if it's not the case that the second then it's not the case that the first).

The second sentence is no less *formal* than the first. The *formality* of logic arises from the study of forms of arguments. The symbolism is just a convenient way of representing these forms. The use of symbols for the operators and letters for statements makes it easier to see at a glance the structure of the statement.

Other operators beyond the conditional and negation are studied in propositional logic. Two important operators are *conjunction* (**p and q** is the conjunction featuring **p** and **q** as its *conjuncts*: we write this **p & q**) and *disjunction* (**p or q** is the disjunction featuring **p** and **q** as its *disjuncts*: we write this **p ∨ q**).

Together with conditionals and negations, conjunction and disjunction can represent the structural features of a great deal of interesting and important reasoning. These operators form the basis of the *language of propositional logic*. In the following two sections, we will see two different kinds of techniques people have used to determine the kinds of arguments valid in this formal language. Before that, however, let's consider a larger language: the language of predicate logic.

If you think of the statements in propositional logic as *molecules*, then *predicate logic* introduces *atoms*. Consider the following short argument:

> **Horses are animals.**
> **Therefore, heads of horses are heads of animals.**

This is a valid argument, and it possesses a valid *form*.[9] It is a form it shares with this argument:

> **Philosophers are academics.**
> **Therefore, children of philosophers are children of academics.**

There are a number of things going on in these arguments, but nowhere inside the premises or the conclusions will you find a simpler statement. The premises and conclusions are combinations of other sorts of things. The most obvious are *predicates*. **Philosopher** is not a name: many things are philosophers, and many things are not. **Philosopher** is a term that *predicates* a property (being a philosopher) of an entity. The same goes with **horse**, **animal** and **academic**. These are all predicates. Sometimes we can use **child**, and **animal** to predicate properties too, but this is not how these terms work in our arguments. The relevant property we care about is not that this is a head, or that this is a child: it is that this is a head *of some animal*, and that that is a child *of some philosopher*. These parts of the language **head of** and **child of** in these arguments predicate *relations* between things. Just as **philosopher** divides the world into the philosophers and the non-philosophers, **child of** divides *pairs* of things into those where the first thing is a child of the second, and those where the first is not a child of the second. So, **Greg is a philosopher** is true, but **Zack is a philosopher** is not (yet); and **Zack is a child of Greg** is true but **Greg is a child of Zack** is not.

Predicates can be one-place (like **philosopher**), two-place (like **child of**), or three-place (try **... is between ... and ...**) and of higher complexity.

We have already seen one thing that you can do with predicates. You can plug in *names* that pick out individuals, and combine them with predicates to make statements. However, in our arguments we are considering, there are no names at all. Something else is going on to combine these predicates together.

The traditional *syllogistic* logic due to Aristotle took there to be primitive ways of combining one-place predicates (in Aristotelian jargon, these are *subjects* and *predicates*) such as **all *F* are *G***, **some *F* are *G***, **no *F* are *G*** and **some *F* are not *G***. Many arguments can be expressed using these techniques for combining predicates. However, our arguments above do even more with the language. The premises indeed do have the form **all *F* are *G***; they say that all horses are animals, and that all philosophers are academics. To put things more technically, they say that for anything you choose at all, if it is an *F* then it is a *G*. The conclusions also have this form: in the first argument it says that, for anything you choose, if it is a head of a horse it is also the head of an animal. But what is it for something to be the head of a horse? Expressing this using the two-place predicate **head of** and the one-place predicate **horse**, you can see that a thing is the head of a horse just when there is another thing that is a **horse**, and this thing is a head of that thing. There is a lot going on here. We are showing that for

anything we choose at all, if we can choose something (a horse) such that the first thing is the head of this thing, then we can choose something (an animal) such that the first thing is also the head of this thing. Making explicit the way that these choices interact is actually quite difficult. We can get by if we are happy to talk of *this thing*, *that thing*, and *the other thing* all the time. But just as we introduced letters **p**, **q** and so on to stand for statements, it is very useful to introduce letters **x**, **y** and so on to stand in the places of these pronouns. These are the *variables* in the language of predicate logic.

The only remaining pieces of the language we need to express this argument form are ways to express the kinds of choices we made for each pronoun. Sometimes we said that for *anything* we choose, if it is a horse, it is an animal. At other times we said that there was *something* we could choose which was a horse and had our other thing as a head. We have two ways of choosing and stating things: either we say that *everything* has some property, or we say that *something* has that property. These two ways of choosing are called *quantifiers*. Each comes with a variable: The *universal quantifier* **(All x)** — symbolised as **(∀x)** — indicates that our statement is true for any choice for **x** at all. The *existential quantifier* **(Some x)** — symbolised as **(∃x)** — indicates that our statement is true for some choice for **x**. (You must be careful here: in English we almost always read "some" as meaning "a few".) Given the language of predicates, variables and quantifiers, our arguments then have the following form:

> **(All *x*)(if *Fx* then *Gx*)**
> **So, (All *y*)(if (Some *z*)(*Fz* and *Hyz*) then (Some *z*)(*Gz* and *Hyz*))**

Or, using the symbols at our disposal:

> **(∀*x*)(*Fx* ⊃ *Gx*)**
> **So, (∀*y*)((∃*z*)(*Fz* & *Hyz*) ⊃ (∃*z*)(*Gz* & *Hyz*))**

This notation makes very explicit the dependencies between the choices made in quantifiers. For example, it makes clear the two different claims: *someone robbed everyone*, and *everyone was robbed by someone*.[10]

> **(∃*x*)(∀*y*)*Rxy***   **(∀*y*)(∃*x*)*Rxy***

In *someone robbed everyone*, the choice of *someone* happens first, and we state that that person robbed everyone. In *everyone was robbed by someone*, we consider each individual person first, and for each person, we state that someone

robbed this person. The person who robbed this person might not have robbed anyone else.

Some predicates have special properties. A very special predicate, from the point of view of logic, is the *identity* predicate, most often depicted by the "=" sign. A statement of the form **a = b** is true if and only if the names **a** and **b** denote the same object. A language with identity can state much more than a language without it. In particular, in the language of predicate logic with identity, we are able to *count* objects. For example, we can say that there is exactly one object with property **F** by with the expression

$$(\exists x)(Fx \ \& \ (\forall y)(Fy \supset x = y))$$

which states that something is **F** and that if anything at all is **F** it is the same object as the first object we chose.

There is much more that you can say about formal languages, but we must stop here. It is not clear that *all* of the structure relevant to determining the validity of arguments can be explained using the techniques we have seen so far. Some arguments utilise notions of possibility and necessity (It *could* rain and the game *has* to be played, therefore we *could* be playing in the rain), predicate *modifiers* (She walked very quickly, therefore she walked) and quantification over *properties* as well as *objects* (The evening star has some property that the morning star doesn't, so the evening star and the morning star are different objects). Various *extensions* of this formal language have been considered and used to attempt to uncover more about the forms of these kinds of arguments.

Interpretations

Given a formal language, we have a precise grasp of the kinds of assertions that can be made and the features we need to understand in order to give an account of validity. Recall that we take an argument to be valid if and only if in any circumstance in which the premises are true, so is the conclusion. As a result, to establish which arguments are valid in a formal language, it suffices to give an account of what these circumstances might be, and what it takes for a sentence in our formal language to be true in a circumstance.

Of course, in our *formal* languages, sentences are not genuinely true or false, because they do not *mean* anything. If we were to be precise, formulas such as **p & q** or **(∀x)(Fx ⊃ Gx)** have a meaning only when meanings are given to their constituents. However, we can think of formulas as *derivatively* true or false, because they might be used to stand for true or false sentences in the language we

are interested in expressing arguments. With this in mind, let's continue with the fiction of thinking of formulas as the kinds of things that might be true or false.

One way to think of circumstances appropriate for the analysis of arguments in *propositional* logic is to think of what they must do. A circumstance must decide the truth or otherwise of formulas. The language of propositional logic makes this task easy, because many of the operators of propositional logic interact with truth and falsity in special ways. The simplest case is negation: if a formula *p* is true, then its negation *~p* is false, because it "says" that *p* is not true — which is wrong, because *p is* true. On the other hand, if *p* is false, then its negation *~p* is true, because it "says" that *p* is not true, and this time this is correct. This small piece of reasoning can be presented in a table, which we call a *truth table*.

p	*~p*
0	1
1	0

Here, the number **1** represents truth and the number **0** represents falsehood. The two rows represent two different circumstances. In the first, *p* is false, and as a result, *~p* is true. (We can say that in this circumstance the truth *value* of *p* is **0** and the value of *~p* is **1**.) In the second, *p* is true, and as a result, *~p* is false.

The fact that the truth value of a negation depends only on the truth value of the proposition negation means that negation is *truth functional*. More involved truth tables can be given for the other operators of propositional logic, because they are truth functional in exactly the same way.

We can present truth tables for other operators, by giving rows for the different circumstances — now we have *four* because there are four different combinations of truth or falsity among *p* and *q* — and a column for each complex formula: conjunction, disjunction and the conditional.

p	*q*	*p & q*	*p ∨ q*	*p ⊃ q*
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	1

Two of the three columns are straightforward: a conjunction is true if and only if both conjuncts are true, and a disjunction is true if and only if either disjunct is true. Here, disjunction is *inclusive*: *p* ∨ *q* is true when *p* and *q* are true. Another operator, *inclusive* disjunction, is false when both disjuncts are true. The column that causes controversy belongs to the conditional. According to this column, a conditional *p* ⊃ *q* is false only when *p* is true and *q* is false. While it is clear that a conditional with a true antecedent and false consequent is false (we learn that **if it is cloudy it is raining** is false if it is a cloudy day without rain, for example) it is no means as certain that it this is the *only* way that a conditional can be false. (After all, **if it is cloudy it is raining** certainly *seems* false even on a fine day when it is neither cloudy nor raining.) However, if we are to give a truth table for a conditional, it must be this one. (A conditional must be true if the antecedent and consequent have the same truth value, if *p* ⊃ *q* is to always be true. A conditional must also be true if the antecedent is false and the consequent true, if **(*p* & *q*) ⊃ *p*** also is to be true when *p* is true but *q* is false.) Much has been said both in favour and against this analysis of the conditional. We will see a little of it in a later section. For now, it is sufficient to note that these rules for the conditional define *some* kind of **if ... then ...** operator, which happens to suffice for many arguments involving conditional constructions.[11]

It is also common to define another operator in terms of the conditional. The *biconditional **p ≡ q*** can be defined as **(*p* ⊃ *q*) & (*q* ⊃ *p*)**, that is, it says that if *p* is true, so is *q* and vice versa. Equivalently, you can define the biconditional by requiring that *p* ≡ *q* gets the value true if and only if *p* and *q* get the same truth value.

Given this understanding of the interaction between operators and truth values, it is a very short step to using it to evaluate argument forms. After all, in any *circumstance* statements receive truth values. So, to evaluate an argument form involving these operators, it suffices to consider all of the possible combinations of truth values to the constituent atomic formulas that make up the argument form. For then, we can spot precisely the kinds of circumstances in which the statements are true, and those in which they are false. Since validity amounts to the preservation of truth in circumstances, we have a technique for testing validity of argument forms. Let's examine this technique by way of an example. Consider the argument form

> ***p* or *q* therefore if *q* then both *p* and *q***

with one premise and one conclusion. We can test it for validity by considering each of the possible combinations of truth values for *p* and *q*, and then using the

truth table rules for each operator to establish the truth values of the premise and the conclusion, given each choice for *p* and *q*. This data can be presented in a table.

p	*q*	*p*	∨	*q*	*q*	⊃	(*p*	&	*q*)
0	0	0	0	0	0	1	0	0	0
0	1	0	1	1	1	0	0	0	1
1	0	1	1	0	0	1	1	0	0
1	1	1	1	1	1	1	1	1	1

The two columns to the side list all the possible different combinations of truth values to *p* and *q*. As before, each row represents a different kind of circumstance. For example, the first row of truth values represents a kind of circumstance in which *p* and *q* are both false. Then, the other two sections of the table present the values of the premise and the conclusion in each of these rows. The values are computed *recursively*: In the first row, for example, the value of *p* & *q* is **0** because *p* and *q* are both **0**: the value is written under the ampersand, the primary operator of this formula. The value of *q* ⊃ (*p* & *q*) is **1** because *q* and *p* & *q* both have the value **0**. The other values in the table are computed in the same fashion.

In the table, the values of the premise and the conclusion are found in the two shaded columns. So, for example, in the first row, the premise is false and the conclusion is true. In the second (shaded) row, the premise is true but the conclusion is false. In the last two rows, the premise and conclusion are both true. Each row is an *interpretation* of the formulas, sufficient to determine their truth values. This information helps us evaluate the argument form. Since there is an interpretation in which the premise is true and the conclusion is not (that given by the second row) the argument form is *invalid*.

This is a sketch of the truth table technique for determining the validity of argument forms in the language of propositional logic. You can see that for any argument form, featuring *n* basic proposition letters, a truth table with 2^n rows is sufficient to determine the validity of this form. This means that we have a process that we can use to tell us whether or not an argument is valid. Validity in propositional logic is *decidable*. We will see more about decidability in the final section of this chapter.

There are a number of ways to that these techniques can be extended further. One way we will not pursue here is to extend the class of truth values from the straightforward *true* and *false*. A popular extension is to admit a third value for statements that at least *appear* to be neither true nor false (think of borderline cases such as *Max is bald*, when Max is not hairless yet not hairy; or think of paradoxical sentences such as *this very statement is false*): you can think of adding extra values as extra *truth* values. However, this is not the only way to view modifications of this simple two-valued scheme. We may use more than two values to evaluate statements, without thinking of those values as extra truth values. Perhaps more values are needed to encode different kinds of semantic information. At any rate, *many-valued logic* is an active research area to this day (Urquhart: 1986).

Another way that interpretations such as truth tables can be extended is to incorporate the predicates, names, variables and quantifiers of *predicate* logic. Here, we need to do a lot more work than with truth tables. To interpret the language of predicate logic we need to decide how to interpret *names*, *predicates*, *variables* and *quantifiers*. I will spend the rest of this section sketching the most prevalent way for providing interpretations for each semantic category. This technique is fundamentally due to Alfred Tarski, who pioneered and made precise the kinds of *models* for predicate logic in widespread use today (Tarski: 1956).

In the case of truth tables, an interpretation of an expression was a simple affair: we distribute a truth value to each basic proposition letter, and we get the truth or falsity of every complex expression as defined out of them. We need something similar in the case of the language of predicate logic: we need enough to determine the truth or falsity of each expression of the language of predicate logic. But how can we do this? There are connections between different expressions such as *Fa*, *(∀x)Fx* and *(∃x)Fx*. It will not suffice to distribute truth values among them. We need some way to understand the connections between them, and this will require understanding the ways that names, predicates, variables and quantifiers function, and how they contribute to truth.

A natural way to interpret names is to pair each name with an *object*, which we interpret the name as *denoting*. The collection of objects that *might* be used to denote names we will call the *domain* of the interpretation. Then, to interpret a *predicate*, we need at the very least something that will give us a truth value for every object we wish to consider. Given that we know what *a* denotes, we wish to know if it has the *property* picked out by *F*. This is the very least we need in order to tell whether or not *Fa* is true. So, to interpret a one-place predicate, we

require a *rule* that gives us a truth value (true or false; or equivalently, **1** or **0**) for each object in the domain. The same thing goes for two-place predicates, except that do tell whether or not *Rab* is true, we need a truth value corresponding to the *pair* of *a* and *b*: so a two-place predicate is interpreted by a rule that gives a truth value for every *pair* of objects in the domain.

Names and predicates are the straightforward part of the equation. The difficulty in interpreting the language of predicate logic is caused by quantifiers and variables. The major cause of the difficulty is in the behaviour of variables. Variables by themselves are meaningless. Even if we have an interpretation of each of the names and predicates in our language, variables don't mean anything in particular. In fact, variables don't have any interpretation at all in isolation. If we know what *F* means, it does not help in answering the question of what *Fx* means, and in particular, whether or not it is true. Variables have meanings by themselves only in the context of *quantifiers*. For example, given an interpretation, *(∃x)Fx* is true if and only if *something* in the domain of that interpretation has property *F* (as given by that interpretation). What we need is a rule that can tell us whether or not a quantified formula is true, in general. There are two general ways to do this. One, due to Tarski, keeps variables in the formula and adds exactly what you need to interpret them. The other, expands the language to incorporate names for every object in the domain, and dispenses with variables when it comes to evaluating formulas involving quantifiers.

Tarski's approach notes that you can interpret variables "unbound" by quantifiers, like the *x* in *Fx* if you already know in advance what we take *x* to denote. If we proceed with the fiction that *variables* can denote, we can say whether or not *Fx* is true. The way of maintaining the fiction is to introduce an *assignment* of values to variables. An *assignment* α is a rule that picks out an object in the domain for every variable in the language. Then we can say that a formula like *(∀y)(∃x)Rxy* is true, relative to the assignment α when the inside formula *(∃x)Rxy* is true *for every value for* *y* — which now means that it's true for every assignment β that agrees with α, except that it is allowed to vary the value assigned to the variable *y*. Tarski said that a universally quantified formula is *satisfied* by an assignment, just when the inside formula is satisfied by every *variant* assignment. Similarly, an existentially quantified formula is satisfied by as assignment just when the inside formula is satisfied by *some* variant assignment, as it says that *something* in the domain has the required property.

Another way to go in interpreting a formula like *(∀y)(∃x)Rxy* is to ignore assignments of variables, and to make sure that our language contains a name for

every object in the domain. Then **(∀y)(∃x)Rxy** is true just when **(∃x)Rxa**, **(∃x)Rxb** **(∃x)Rxc** and all other *instances* of **(∀y)(∃x)Rxy** are true. Similarly, **(∃x)Rxa** is true just when *some* instance such as **Raa**, **Rba**, or **Rca** is true.

Both techniques result in exactly the same answers for each formula, given an interpretation, and each technique has its own advantages. Tarski's technique assigns meanings to each expression in terms of the meanings of its constituent parts, without resorting to any *other* formula outside the original expression. The other technique, however, is decidedly simpler, especially when applied to interpretations where the domain is *finite*.

The rules we have discussed here suffice to fix a truth value for every complex expression in the language of predicate logic, once you are given an interpretation for every predicate and name in the language. Predicates, names, variables and quantifiers are interpreted using these techniques, and the operators of propositional logic are interpreted as before. Therefore, we have a technique for determining validity of arguments in the language of predicate logic. An argument form is valid if and only if every interpretation that makes the premises true, also makes the conclusion true.

Let's see how this technique works in a simple example. We will show that the argument form

$$\textbf{(∀y)(∃x)Rxy} \textbf{ therefore } \textbf{(∃x)(∀y)Rxy}$$

is invalid, by exhibiting an interpretation which makes the first formula true, but the second one false. (To guide your intuitions here, think of **R** as "is related to": The premise says that for everyone you choose, there's someone related to them. The conclusion says that someone is related to everyone.) Consider a simple domain with just two objects *a* and *b*. There are no names in the language of the argument, but we will use the two names *a* and *b* in the language to pick out the objects *a* and *b* respectively. To interpret the two-place predicate **R** we need to have a rule that gives us a truth value for every *pair* of objects in the domain. I will present a rule doing just that in a table, like this

R	*a*	*b*
a	**0**	**1**
b	**1**	**0**

This table tells us that **Raa** and **Rbb** are false, but **Rab** and **Rba** are true. As an example of how to read tables for two-place predicates it is not particularly good,

since I haven't told you which of the **1**s is the value for **Rab** and which is the value for **Rba**. The answer is this: The **1** in the first row and second column is the value for **Rab**.

Now, in this interpretation, **(∀y)(∃x)Rxy** is true, since **(∃x)Rxa** and **(∃x)Rxb** are both true. (Why are these true? Well, **(∃x)Rxa** is true since **Rba** is, and **(∃x)Rxb** is true because **Rab** is.) However, in this interpretation **(∃x)(∀y)Rxy** is not true, since **(∀y)Ray** and **(∀y)Rby** are both not true. (Why are these not true? **(∀y)Ray** is not true because **Raa** is not true, and **(∀y)Rby** is not true because **Rbb** is not true.) Therefore, the argument is invalid.

This very short example gives you a taste of how interpretations for predicate logic can be used to demonstrate the invalidity of argument forms. Of course, doing this at this level isn't necessarily an advance over simply thinking **(∀y)(∃x)Rxy** could be true if every object is related by **R** to some other object: it doesn't follow that **(∃x)(∀y)Rxy** because objects could be paired up, so that nothing is related by **R** to everything. Demonstrating invalidity by dreaming up hypothetical examples still has its place: the techniques of formal logic don't replace thought, they simply expose the structure of what we were already doing, and help us see how the techniques might apply in cases where our imagination or intuition give out.

You may notice that interpretations of the language of predicate logic differ from truth tables in one very important respect. We could easily *list* all of the possible different interpretations of an argument in propositional logic. In predicate logic this is no longer possible. We could interpret an argument in a domain of *one* object, of *two* objects, of *three*, or of 3088, or of a million, or of an infinite number. There is no limit to the number of different interpretations of the language of predicate logic. This means that our definition of validity for expressions in predicate logic does not give us a recipe for determining validity in practice. If we chance on a counterexample showing that an argument is invalid, we might be able to verify that the argument is invalid.[12] But what if there is no counterexample? How could we *verify* that there isn't one? Going one-by-one through an infinite list is not going to help. Finding an alternative way to *demonstrate* validity requires a different approach. We need to go back to square one, and examine an alternative analysis of validity.

Proofs

Interpretations are one way to do *semantics*: to give an account of the *significance* of an expression. In doing this, models work from the *inside*, *out*. In

truth tables for prepositional logic, the truth value of a complex proposition is determined by the truth values of its constituents. In models for predicate logic, the satisfaction of a complex formula in a model is determined by the satisfaction of its constituents in that model. In other models for other kinds of logics, the same features hold.

This is not the only way to determine significance. Another technique turns this on its head: you can work *outside*, *in*. We may determine the significance of a complex expression in terms of its surface structure. Let's start with an example. Consider the argument form:

p therefore (if *q* then both *p* and *q*)

One way to deal with the argument is to enumerate the different possibilities for *p* and *q* and consider what this expression might amount to in each of them, in terms of these possibilities. This is proceeding *inside out*.

On the other hand, we might work *outside in* by *supposing* that the premise is true, and then seeing if we can show that the conclusion is true. We might ask what we can do with the conclusion, an **if … then …** statement, by asking how we could show it to be true (or in general, how we could show that it follows from some collection of assumptions). It is a plausible thought that **if … then …** statements can be proved by *assuming* the antecedent (in this case, *q*) and then by *deducing* the consequent (in this case: **both *p* and *q***). To prove this on the basis of the assumptions we have, it suffices to look back and see that we have assumed both *p* and *q*. So, our argument seems valid: we have shown that the conclusion **if *q* then both *p* and *q*** follows from the premise *p*.

What we have just done is a *proof*: it is what is called a *natural deduction* proof. We can present that proof in a diagram. Here is one way of presenting what has gone on in that paragraph.

$$\frac{\dfrac{\quad}{}}{} $$

$$\frac{p \qquad \overline{q}}{\dfrac{p \mathbin{\&} q}{q \supset (p \mathbin{\&} q)}}$$

We start by *assuming **p*** and ***q*** at the top of the diagram. Then we deduce the conjunction ***p* & *q*** at the next line. Then in the last step, we *discharge* the assumption of *q* to deduce the conclusion ***q* ⊃ (*p* & *q*)**.

This is a *natural deduction proof* in the style of Prawitz (1965). There are many other different styles of presenting proofs like this (Fitch: 1952, Lemmon: 1965),

and it is most common to be taught logic by means of one of these kinds of systems of proof. Natural deduction proofs have a virtue of being very close to a natural style of reasoning we already use.
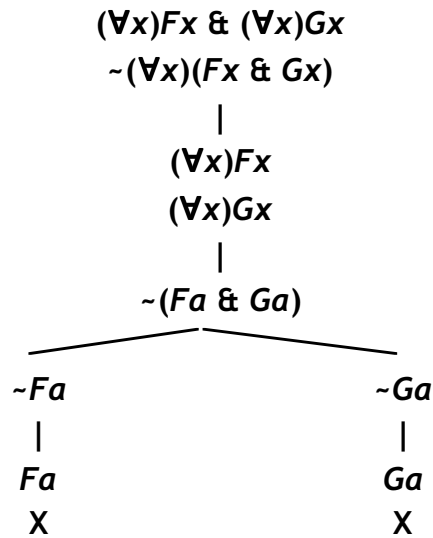
The rules for each operator can be supplemented with rules for quantifiers, which results in a proof system for the whole of predicate logic. For example, here is a proof showing **(∀x)(Fx & Gx)** that follows from **(∀x)Fx & (∀x)Gx**.

$$
\frac{\dfrac{\dfrac{(\forall x)Fx \,\&\, (\forall x)Gx}{(\forall x)Fx} \qquad \dfrac{(\forall x)Fx \,\&\, (\forall x)Gx}{(\forall x)Gx}}{\dfrac{Fa \qquad\qquad Ga}{Fa \,\&\, Ga}}}{(\forall x)(Fx \,\&\, Gx)}
$$

The interesting moves in this proof are those involving quantifiers. In the left branch we move from **(∀x)Fx** to **Fa**: from a universal quantifier to some instance of it. Similarly in the right branch, we move from **(∀x)Gx** to **Ga**. Then, after deducing **Fa & Ga** from the two conjuncts, we deduce the final conclusion **(∀x)(Fx & Gx)**. This move is valid not because **(∀x)(Fx & Gx)** follows from **Fa & Ga** — it doesn't — but because we proved **Fa & Ga** without assuming anything about **a**. The only assumption the proof made was **(∀x)Fx & (∀x)Gx**. So, **a** was arbitrary. What holds for **a** holds for anything at all, so we can conclude **(∀x)(Fx & Gx)**.

There are other kinds of proof system beyond natural deduction. Hilbert-style proof theories typically have number of *axioms* and a small number of rules, and construct proofs in a similar way to natural deduction systems. Tableaux or tree proof theories are somewhat different: instead of attempting to *demonstrate* statements, tableaux systems aim to show that statements are *satisfiable* or *unsatisfiable*. They are still *decompositional* theories, decomposing statements into their constituents, but instead of asking "how can I *prove X*?" you ask "what *follows from X*?" You show that an argument is valid, using a tableaux system by showing that you cannot make the premises true and the conclusion false: that is, the premises and the negation of the conclusion, considered together, are *unsatisfiable*.

Here, for example, is a tableaux proof showing that **(∀x)(Fx & Gx)** follows from **(∀x)Fx & (∀x)Gx**: that is, that **(∀x)Fx & (∀x)Gx** and **~(∀x)(Fx & Gx)** cannot be true together.

$$
\begin{array}{c}
\textbf{(∀x)Fx \& (∀x)Gx} \\
\textbf{~(∀x)(Fx \& Gx)} \\
| \\
\textbf{(∀x)Fx} \\
\textbf{(∀x)Gx} \\
| \\
\textbf{~(Fa \& Ga)}
\end{array}
$$

~Fa	**~Ga**
|	|
Fa	**Ga**
X	**X**

In this proof, as in the natural deduction proof, we deduce **(∀x)Fx** and **(∀x)Gx** from **(∀x)Fx & (∀x)Gx**. However, from **~(∀x)(Fx & Gx)** we deduce that there must be *some* object which doesn't have both properties **F** and **G**. We call this object **a**. Now, since **~(Fa & Ga)** is true, it follows that *either* **~Fa** or **~Ga**. Our way of representing this is by *branching* the tree into two possibilities. But now in the left branch we can use **(∀x)Fx** to deduce **Fa** which conflicts with **~Fa**, and in the right we can use **(∀x)Gx** to deduce **Ga** which conflicts with **~Ga**. Neither case is satisfiable in an interpretation, so there is no interpretation at all in which **(∀x)Fx & (∀x)Gx** and **~(∀x)(Fx & Gx)** are true together: the argument is valid.

Proofs like these are one way to demonstrate conclusively that an argument in the language of predicate logic is indeed *valid*. To use proofs as a technique for determining validity of arguments, where validity is defined in terms of *interpretations* you need to have some connection between proofs and interpretations. Such a connection is provided by *soundness* and *completeness* results. A system of proofs is *sound* if any argument you can show to be valid using the proof system is indeed valid (according to the definition in terms of interpretations.) You can show that a system of proofs is sound by going through every rule in the proof system, showing that they are valid, and by checking that stringing together valid arguments in the ways licensed by the proof system results in more valid arguments. Soundness results are generally straightforward (if tedious) to demonstrate.

A system of proofs is *complete* if any argument that is indeed valid can be shown to be valid by some particular proof. Completeness results are typically much more difficult to prove. It is usually possible to demonstrate the equivalent result that if some argument cannot be proved by a particular proof system, then there is some interpretation that makes the premises true and the conclusion false. The techniques for demonstrating completeness are beyond the scope of this chapter, but they are some of the most important techniques in 20th Century logic.

Directions and Issues

I have attempted to give a general overview of some of the motivations, tools and techniques that have been important in the study of logic in the last century. In this remaining section, I will consider a few issues that arise on the basis of this foundation. These issues each form the core of distinct research programmes that are alive and flourish today.

Decidability and Undecidability: We have already seen that the classes of interpretations appropriate for propositional and predicate logic differ in one important respect. To check a propositional argument form for validity, you need only check finitely many interpretations. To check a predicate logic form for validity you may need to check *infinitely* many interpretations. It has been shown that this is not an artefact of the way that interpretations have been defined. There is *no* recipe or algorithm for determining whether or not an argument form of predicate logic is valid or invalid. To be sure, some valid arguments can be shown to be valid, and some invalid arguments can be shown to be invalid. However, there is no single process which, when given any predicate logic argument form, will determine whether or not the argument form is valid. This result follows from Gödel's celebrated *incompleteness* theorem, which I will discuss below. To defend the claim that there is no algorithm determining validity, however, you need to have a precise account of what it is for something to *be* an algorithm. The clarification of this notion is another of the highlights of 20th Century logic. It has been shown that different explanations of what it is for a process to be an algorithm — a mathematical definition in terms of *recursive functions*, and different concrete implementations of algorithms in terms of *Turing machines* and *register machines* — are all equivalent: they pick out the same class of processes. This lends support to *Church's Thesis*: all computable processes are computable by Turing machines (Boolos and Jeffrey: 1989). Given such a precise identification of the range of the computable, it follows — after some work (Boolos and Jeffrey: 1989) — that there is no algorithm for determining validity in predicate logic. This means that no computer can be

programmed to decide validity for arguments. Any theorem-proving software expressive enough to handle arguments in predicate logic, there will be arguments that it will not be able to determine for validity.

Contemporary work in computer science aims to understand not only the border between the computable and the uncomputable, but also different grades of computability. Distinctions may still be drawn between problems that are solvable by algorithm. Some problems, such as evaluating propositional validity, are decidable by *exponentially difficult* algorithms. Here, the number of cases to check grows exponentially in terms of the length of the problem itself. (A problem in 3 sentence letters requires checking $2^3 = 8$ cases. A problem in 100 sentence letters requires checking 2^{100} cases. That is, you must check

$$1,267,650,600,228,229,401,496,703,205,376$$

cases, which is a great deal more. (Checking a billion cases a second would still leave you with 4×10^{13} years of work to complete your task.) So, an exponentially difficult problem like this can in practice be impossible to carry out.

Completeness and Incompleteness: The insight that validity in predicate logic is *undecidable* followed from Kurt Gödel's groundbreaking work showing that elementary arithmetic is *incomplete*. That is, he showed that any collection of premises about numbers (expressed in the language of predicate logic, with enough vocabulary to express identity, addition and multiplication) would be *incomplete* in the sense that not every truth about the natural numbers (the *whole* numbers 0, 1, 2, 3 ...) would follow from those premises. This is not quite right as it stands, of course, because you could take the premises to be the collection of *all* of the truths about whole numbers, and this would be trivially complete, at the cost of being uninteresting. No, Gödel showed that provided you had an *algorithm* for determining whether or not something was a premise for your theory of numbers, then your theory of numbers (the collection of conclusions provable from your premises) could not be complete, at least, not if it was consistent. If it were consistent, it would always leave *some* truth about numbers out.

Gödel's technique for demonstrating this is his celebrated encoding of statements about *proofs* and other *statements* into statements about *numbers*: what we now call a *Gödel* numbering. Arithmetic happens to be expressive enough to ensure that any statements about proofs check with an algorithm can be encoded into statements about numbers. (The technique is not altogether different from the encoding of a document on a computer into a string of digits in ASCII or in Unicode.) Then statements about statements and proofs can be manipulated in a

theory designed to talk only about numbers. In particular, you can get a theory *T* of arithmetic to express a statement *G* that would be true if and only if the theory *T* cannot prove that *G* is true. (The statement *G* basically "says of itself" that it is not provable in *T*.) Now consider what the theory *T* can say about *G*. If the theory *T* can prove *G*, then the theory *T* proves some falsehoods, since *G* is true if and only if it *cannot* be proved in *T*. So, if the theory cannot prove *G* then *G* must be true (because it is true if and only if it cannot be proved). So the theory *T* is incomplete.

This technique is general and it applies to other theories beyond arithmetic. An important branch of mathematics studies the relationships between different mathematical theories of increasing strength, and ways to extend incomplete theories in natural directions.

Gödel's results also have important philosophical consequences. Hilbert's program of finding a philosophically acceptable *foundation* of mathematics in terms of logical consistency ran aground because logical consequence and consistency was shown by Gödel to not be a finitary algorithmically-checkable notion. Logic, in its complexity, remains useful as a tool for the analysis of arguments. It is less appealing as a straightforward foundation of mathematics or any other discipline.

Which are the logical constants*?* We have shown that a lot can be done by focussing on the propositional operators and quantifiers. These expressions are logical *constants*. Their interpretation is held fixed in every model. We can vary propositional letters, names or predicates, but an interpretation is not allowed to model *conjunction* as *disjunction*. Is this simply a matter of convenience, or is this a matter deeply embedded in the notion of logical consequence? There is no settled answer to this question. Some take the logical constants to be privileged symbols in our language; because of some special feature they have, such as truth functionality or some analogue (see, for example Quine: 1970). Others take the distinction between the logical and non-logical vocabulary to be a conventional one (Etchemendy: 1990).

Conditionality and Modality: When we considered the truth table for the conditional, we saw that it appears to leave a lot to be desired. After all, if I assert

<div align="center">

If it's Sunday, it's a weekday

</div>

on a Wednesday, we would not be inclined to say that my statement is *true* just because the antecedent is false and the consequent is true. We'd much more likely think that if it *were* a Sunday, it would be a weekend, and not a weekday,

and so, the statement is actually *false*. What we have done in considering this statement in this way is to consider *alternative circumstances*. We have asked ourselves what the world *would* have been like had the antecedent been true, and in particular, we consider if the consequent would have been true in those circumstances. In doing this, we need to move beyond the simple evaluation of propositions just in terms of their truth or falsity to consider their truth or falsity in other circumstances. We have done this already to a small extent in the evaluation of arguments (which is, after all, a *conditional* notion: we want to know whether or not *if* the premises are true, *then* the conclusion is true). A *modal* account of the conditional says that the same must be done for (at least some) *if ... then ...* statements. A conditional statement depends not only on the truth or falsity of the antecedent and consequent here and now, but also on the connections between them in *alternative* circumstances.

This makes the conditional a kind of *modal* operator like *necessity* and *possibility*. It is *necessary* that **p** if and only if **p** is true in all alternative circumstances (so **p** is not only true, but it is in some sense *unavoidable*). It is *possible* that **p** if and only if **p** is true in some alternative circumstances (so **p** might not be true, but were things to turn out like *that* circumstance **p** *would be* true). Similar logical features are displayed by other operators that pay attention to *context* such as *temporal* operators (*always* **p** is true at some time if and only if **p** is true at all times) and *location* operators (*around here* **p** is true at some location if and only if **p** is true at all locations near that location). In each case, we see that the semantic value of an expression depends not only on its truth or falsity, but its truth or falsity in some kind of *context*. The study of these kinds of operators has flowered in the latter part of the 20th Century. See the further reading list for some places to pursue this material.

Relevance: Some arguments seem to be invalid, despite coming out as valid according to the definitions we have given. Some particularly tricky arguments are the *fallacies of relevance*:

$$p \text{ therefore } q \text{ or not } q \qquad p \text{ and not } p \text{ therefore } q$$

The first argument turns out to be valid because *every* interpretation makes **q or not q** true, whether or not **p** is true. So in every interpretation in which **p** is true, so is **q or not q**. Similarly, *no* interpretation makes **p and not p** true, so there is no interpretation in which **p and not p** is true but **q** isn't, so there is no counterexample to the validity of the second argument. A minority tradition in 20th Century logic has argued that something is mistaken in the dominant account, because it does not pay heed to the norms of *relevance*. These

arguments are invalid because the premise has nothing to do with the conclusion in each case. In the first argument, the conclusion is true (and necessarily so) but it need not *follow from* the premise. In the second case the premise is false (and necessarily so) but again, the conclusion need not follow from it (Read 1995). *Relevant* logics attempt to formalise a notion of validity that take these considerations into account.

Vagueness: Another issue with the standard account of propositional logic arises from the assumption that every interpretation assigns exactly one of the values *true* and *false* to every expression. This is certainly less than obviously true when it comes to *vague* expressions. Consider a strip of colour shading evenly from fire engine red on the one side to lemon yellow on the other. Consider the statement "that's red" expressed while pointing to parts of the strip, going from left to right. The statement is true when you start, and false when you finish. Where along the strip did it change from true to false? It is difficult to say. This is one way to think of the problems of *vagueness*. One option is to say that logic has nothing to do with vagueness. Logical distinctions apply only when we have precise notions and not vague ones. This seems like an unpalatable option, because our languages are *riddled* with vague notions, and there seems to be no way to eliminate them in favour of precise ones. It would be dire indeed to conclude that there is *no* distinction between validity and invalidity for arguments involving vague notions.

So, let's consider options for considering how logic might apply in the context of vagueness. (Williamson (1994) gives a good overview of the issues here. Read (1996) supplies a helpful shorter account.) One option is to say that logic applies but that the standard *classical* two-valued account does not apply to vague predicates. A vague predicate is not interpreted as a rule giving just *true* or *false* for every object: it must do more. One simple account is to say that a vague predicate supplies the value *true* to every object definitely within the extension of the predicate, *false* to every object definitely outside it, and a third value, *neither*, to the rest. This might be appealing, but it almost certainly doesn't give the right answer in general. One problem plaguing this three-valued approach is the way that it trades in *one* sharp borderline (between the *true* and the *false*) for *two* sharp borderlines (between the *true* and the *neither* and between the *neither* and the *false*). If there is no sharp borderline between the red and the non-red in the strip of colours, there is no sharp borderline between the definitely red and the neither definitely red nor definitely non-red, and there is no sharp borderline between the neither definitely red nor definitely non-red and the definitely non-

red. The distinctive behaviour of the strip of colours is that there appears to be *no* sharp borderline, not that there are *two*.

A more popular modification of classical predicate logic to deal with vagueness is to take predicates to be interpreted by *degrees*. The predicate *red* is interpreted not as dividing things into the red and non-red, and not as dividing things into the definitely red, definitely non-red and the neither, but rather as assigning a number between 0 and 1 to every object: its *degree* of redness. Canonical red things get degree 1, canonical non-red things get degree 0, and other things in between get an in-between value, such as 1/2, 0.33, or any other number in the interval [0, 1]. This is the approach of *fuzzy logic*. If we leave things as I have explained them, we are in no better situation than in the three-valued case: we have traded in one or two borderlines for *infinitely many*. Now there is not only a sharp difference between things which are definitely red (red to degree 1) and not quite definitely red (red to some degree less than 1), there is also the sharp difference between things which are more red than not (red to a degree greater than 1/2) and things which are at least as non-red as they are red (red to a degree no greater than 1/2), and infinitely more borderlines besides. Again, the strip of colours doesn't seem to exhibit *this* structure. Proponents of fuzzy logic respond that the assignment of values to objects is *itself* a matter of degree, and not an all-or-nothing matter. Whether this can be made coherent or not, is a matter of some debate.

The alternative is to think that the classical two-valued approach still works in the case of vagueness, but that it must be interpreted carefully. There are two major traditions here. The first, *supervaluational* approach, due to van Fraassen and Fine (see Williamson's book for references) takes an interpretation to still assign one the two truth values to each predicate/object pair, but vagueness means that our language doesn't pick out one interpretation as the *right* one. There are a number of ways that you could acceptably draw the borderline between the red and the non-red. Something is *true* if it is true in any acceptable interpretation. A *supervaluation* is a class of interpretations (or valuations). Again, supervaluational approaches face a similar problem with borderlines. The very notion of something being an acceptable interpretation seems to be a vague notion. Another kind of problem for supervaluational approaches is the fact that they seem to undercut their own position: it is true on any acceptable interpretation that there is a last red spot on the spectrum: red patch were but the very next patch is not red.[13] For any interpretation at all draws a line between the red and the non-red. So according to supervaluations there is a sharp borderline, but there is no line such that *that* is the line between the red and the non-red.

The major alternative approach is to bite the bullet and conclude that there *is* a borderline between the red and the non-red. However, the fact that the predicate is vague means that the borderline is impossible to discern. While there is one correct interpretation of the predicate *red*, the class of acceptable interpretations might be the best we can do in actually *determining* what the extension of *red* might be. Vagueness, then, is a limitation in our *knowledge* rather than our *semantics*. The meaning of the term *red* is picked out precisely, but our capacity for recognising that meaning is not complete. So goes the response of the *epistemicist* about vagueness. Williamson's book (1994) is a spirited defence of epistemicism.

Meaning: Vagueness is just one phenomenon alerting us to the fact that the interpretation of logical techniques is fraught with philosophical issues. So is the relationship between *proofs* and *models*. A number of deep philosophical concerns about *meaning* hang on the way we ought to analyse and understand the meanings of statements. Broadly *realist* approaches enjoin us to analyse meaning in terms of truth conditions (Devitt: 1991) or what we might see as *models* or *interpretations*. Broadly *anti-realist* approaches enjoin us to take *inference* or *proof* as primary (Brandom: 1994, Dummett: 1991). It is not my place here to determine the virtues or vices of either side in this debate (to do so would take us away into concerns in the philosophy of language). Committed realists will take models or interpretations as primary, and view proof theory as derivative. Committed anti-realists will take proof theory as primary, and take models as derivative.[14] *Logicians* who start with a thorough grounding in the techniques of models and of proofs, and who know that they can be shown to be equivalent, might ask different questions. In what sense might proofs be primary? In what sense might models be primary? In what sense might they do exactly the same job?

Questions

1. Is validity always a matter of logical form? Can you think of any genuinely *valid* argument that doesn't exhibit any kind of *form* or *structure* responsible for its validity?

2. What is special about the logical operators of conjunction, disjunction, negation and implication? What other operators, if any, have any of these special features?

3. What is special about the *existential* and *universal* quantifiers? Do other quantifiers like *most* (think: *most* of the beer is gone, instead of *all* of the beer is gone) have interesting logical properties?

4. We have seen proof systems like *natural deduction*, which aim to *demonstrate* formulas, and *tableaux*, which aim to *satisfy* formulas. Is there any other kind of goal a proof system might have?

5. Which has priority: proofs or interpretations? Or can one technique have priority in *one* sense, and the other in a *different* sense?

Bibliography of works cited

Boolos, George and Richard C. Jeffrey. (1989) *Computability and Logic*, third edition, Cambridge: Cambridge University Press.

Brandom, Robert. (1994) *Making it Explicit*, New Haven: Harvard University Press.

Devitt, Michael. (1991) *Realism and Truth*, second edition, Oxford: Blackwell.

Dummett, Michael. (1991) *The Logical Basis of Metaphysics*, New Haven: Harvard University Press.

Etchemendy, John. (1990) *The Concept of Logical Consequence*, Cambridge, Massachusetts: Harvard University Press.

Fitch, F. B. (1952) *Symbolic Logic*, New York, Roland Press.

Frege, Gottlob. (1972) *Conceptual Notation and Related Articles*. Translated and edited with a biography and introduction by Terrell Ward Bynum, Oxford: Oxford University Press.

Frege, Gottlob. (1984) *Collected Papers on Mathematics, Logic and Philosophy*, edited by Brian McGuinness, translated by Max Black, V. H. Dudman, Peter Geach, Hans Kaal, E−H. W. Kluge, Brian McGuinness and R. H. Stoothoff, Oxford: Blackwell.

Lemmon, E. J. (1965) *Beginning Logic*, London: Nelson.

Prawitz, Dag (1965) *Natural Deduction*, Stockholm: Almqvist and Wiksell.

Priest, Graham. (1999) "Validity," in *European Review of Philosophy* volume 4: *The Nature of Logic*, pages 183–206, edited by Achille C. Varzi, CSLI Publications.

Tarski, Alfred (1956) *Logic, Semantics, Metamathematics: papers from 1923 to1938*, translated by J. H. Woodger, Oxford: Clarendon Press.

Quine, W. V. O. (1970) *Philosophy of Logic*, Englewood Cliffs, N.J.: Prentice-Hall.

Read, Stephen. (1995) *Thinking about Logic*, Oxford: Oxford University Press

Urquhart, Alasdair. (1986) "Many-Valued Logics," in the *Handbook of Philosophical Logic*, volume 3, pages 71–116, edited by Dov. M. Gabbay and Franz Günthner, Dordrecht: Reidel.

Williamson, Timothy. (1994) *Vagueness*, London: Routledge.

Recommended Reading

There are many books you could use to further your study of logic. This list of recommended reading contains just a small sampling of what you can use to get started on more.

Introductions to Logic

Hodges, Wilfrid. (1977) *Logic*, New York: Penguin.

Forbes, Graeme. (1994) *Modern Logic*, Oxford: Oxford University Press.

Howson, Colin. (1996) *Logic with Trees*, London: Routledge.

Restall, Greg. (200+) *Logic*, London: UCL Press.

Shand, John. (2000) *Arguing Well*, London: Routledge.

Philosophy of Logic

Etchemendy, John. (1990) *The Concept of Logical Consequence*, Cambridge, Massachusetts: Harvard University Press.

Priest, Graham. (2000) *A Very Short Introduction to Logic*, Oxford: Oxford University Press.

Read, Stephen. (1995) *Thinking about Logic*, Oxford: Oxford University Press.

Sainsbury, Mark. (1991) *Logical Forms*, Oxford: Blackwell.

Quine, W. V. O. (1970) *Philosophy of Logic*, Englewood Cliffs, N.J.: Prentice-Hall.

Special Topics

Bell J. L., David deVidi and Graham Solomon (2001) *Logical Options: an introduction to classical and alternative logics*, Peterborough: Broadview Press.

Boolos, George and Richard C. Jeffrey. (1989) *Computability and Logic*, third edition, Cambridge: Cambridge University Press.

Chellas, Brian. (1980) *Modal Logic*, Cambridge: Cambridge University Press.

Dummett, Michael. (1977) *Elements of Intuitionism*, Oxford: Oxford University Press.

Hughes, George and Max Cresswell. (1996) *A New Introduction to Modal Logic*, London: Routledge.

Girle, Roderic. (2000) *Modal Logics and Philosophy*, Teddington: Acumen.

Jeffrey, Richard C. (1991) *Formal Logic: its scope and its limits*, New York: McGraw–Hill.

Priest, Graham. (2001) *An Introduction to Non-Classical Logic*, Oxford: Oxford University Press.

Restall, Greg. (2000) *An Introduction to Substructural Logics*, London: Routledge.

Shapiro, Stewart (1991) *Foundations without Foundationalism: a case for second-order logic*, Oxford: Oxford University Press.

Endnotes

[1] Thanks to my students in logic classes at Macquarie University for their enthusiastic responses when I have to taught logic and shared some of why it is so interesting. Thanks too, to John Shand for detailed comments on an earlier draft of this chapter.

[2] Of course, I recommend my textbook *Logic* in this series. However, the recommended reading list contains a number of books that also serve as excellent introductions to the techniques of formal logic.

[3] These are, of course, two arguments purporting to demonstrate the existence of God. The first is found in the second of Aquinas' *five ways*, and the second is Anselm's ontological argument.

[4] "Circumstance" here, is *very* broadly construed. In fact, it is so broadly construed as to incorporate any consistent circumstance at all. As a result, we could also think of valid arguments as ones in which the premises are *inconsistent with* the denial of the conclusion.

[5] For the circumstance is indeed hypothetical, because presumably there *are* causes and effects. Hypothetical or non-actual circumstances are always within the remit of our discussion, even if they seem crazy or unexpected. The way we demonstrate the invalidity of arguments is to ask "but what would happen *if...?*" One does not have to argue that there *actually* are no causes or effects to show that this argument is invalid.

[6] I am indebted to Graham Priest and his paper "Validity" (Priest: 1999) for this way to consider these kinds of *non-monotonic* virtues of arguments.

[7] There is no more problem for picking the one thing to instantiate different *variables* in logic than there is in mathematics. After all, if $f(x,y) = x + 2y$, we want to be able to say that $f(3,3) = 9$, but in doing that, we're instantiating the variables x and y both to 3.

[8] However, presumably I can *mention* the word cabbage without talking about vegetables.

[9] The validity of these arguments cannot be shown using Aristotle's *Syllogistic*, and it was arguments like these that helped fuel the development of modern predicate logic.

[10] Interestingly, the grammar checker in the software package I used to write this chapter does not know the difference: it prompts me to replace the passive voice in "everyone was robbed by someone" with the active-voiced "someone robbed everyone." If "someone" were grammatically a *name* this would be acceptable. The grammar checker does not understand the difference between names and quantifiers.

[11] It also seems to suffice for the kinds of conditionals used in mathematical reasoning, which perhaps explains why logicians such as Frege and Russell saw this understanding of the conditional as adequate.

[12] Even this might be difficult, if the domain is infinite. Consider statements about numbers. It might be that the domain of the natural numbers is a counterexample to

Goldbach's conjecture (that every even number is the sum of two primes) but verifying this fact, if indeed it is a fact, is a difficult mathematical problem.

[13] Let's presume that the spectrum is divided into a finite but very large number of patches, such that each patch looks indiscernibly different from the patches immediately to its left and to its right.

[14] And in fact, some take constraints on an acceptable proof theory to be so stringent as to motivate us away from classical predicate logic to some weaker logic, such as *intuitionistic* predicate logic (Dummett: 1991).