MPhil in Advanced Computer Science
INTRODUCTORY LOGIC

Leader: Dr. Thomas Forster
Timing: Michaelmas weeks 5-8
Prerequisites: Undergraduate Computer Science Mathematical Background
Structure: 16 lectures

AIMS

The aim of the course is to equip the people who follow it with all the background Logic they might need in the other areas to which they might turn their hand.

SYLLABUS

- Wellfoundedness, Induction and Recursion.
- Propositional Logic. Completeness, interpolation.
- Natural Deduction and Sequent Calculus for Propositional and Predicate Logic
- First-Order Theories: Completeness of first order logic. Skolem-Löwenheim. Semantics. Skolemisation.
- Complete theories and categorical theories. Decidable theories.
- Lambda-calculus and Curry-Howard.
- Other sophisticated logical syntaxes: modal logic, branching quantifiers, ϵ-terms.
- Possible world semantics for constructive and modal Logics.

OBJECTIVES

This introductory course is for Master's students, so it assumes the mathematical background a computer science B.A. will have. It covers a lot of material, some of it quite basic ("advanced revision") and the pace will be stiffer than last year.

On completion of the course students should be fluent and confident in their use of logical syntax, and understand the significance of the theoretical background.

COURSEWORK & PRACTICAL WORK

No practical work; there will be exercises in the body of the online notes.

ASSESSMENT

There will be a two-hour examination at the start of the Lent term.

RECOMMENDED READING

There will be extensive online notes linked from the lecturer's home page at `www.dpmms.cam.ac.uk/~tf/cam_only/teaching.html`, and it is the lecturer's intention that the online notes should be sufficient, so students do not need to buy any books. However, there are many books with the string 'mathematical logic' in the title, and many of them—for example E. Mendelson *Introduction to Mathematical Logic*—are suitable. The lecturer's textbook *Logic, Induction and Sets* is available in most college libraries.

Chapter 1

Two Lectures on Propositional Logic

Syntax of propositional logic. A propositional language is launched by an **alphabet**, which is a set of propositional letters, and the propositional language is the set of all formulæ built up from the propositional letters by means of whatever connectives we specify—typically $\wedge$, $\vee$, $\rightarrow$, $\neg$ and $\longleftrightarrow$. A **valuation** is a function from the alphabet to $\{0, 1\}$ (the set of truth-values). Bear in mind that our valuations are total functions.

Truth-functional connectives only. The language of Propositional Logic is a recursive datatype so we can do structural induction over it.

Here is an example of a propositional theory. We might call it the theory of adding two eight-bit words (with overflow). It has 24 propositional letters, $p_0, \ldots p_7$, $p_8, \ldots p_{15}$ and $p_{16} \ldots p_{23}$, and axioms to say that $p_{16}, \ldots p_{23}$ represent the output of an addition if p_0 to p_7 and p_8 to p_{15} represent two words of input. `true` is 1 and `false` is 0, so it contains things like $((p_0 \wedge p_8) \rightarrow \neg p_{16})$ (because an odd plus an odd is an even!).

	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0
$+$	p_{15}	p_{14}	p_{13}	p_{12}	p_{11}	p_{10}	p_9	p_8
$=$	p_{23}	p_{22}	p_{21}	p_{20}	p_{19}	p_{18}	p_{17}	p_{16}

1.1 Formal semantics

If you know what a topological space is, and what a compact space is, you might like to try the following exercise:

Exercise 1 *$\mathcal{V}$, the space of valuations, is the set of all valuations on the alphabet, P, of $\mathcal{L}$. P is infinite. We endow $\mathcal{L}$ with a topology by taking, for each finite set $P' \subset P$ and each finite map $v : P' \rightarrow \{0, 1\}$, the set*

$$\{v' : (\forall p \in P')(v(p) = v'(p))\}$$

(of all valuations that agree with v) to be a basic open set. Prove that this topology is compact.

The key to not getting lost in this enterprise is to bear in mind that the expressions of propositional logic are built up from atomic formulæ (letters) whose meaning is not reserved: they can be anything: *Anna can cancan* or *Kant can't cant*, but the symbols in the logical vocabulary—'$\wedge$', '$\vee$' and so on—emphatically *are* reserved.

A valuation [for propositional language] is a function that assigns truth-values (not meanings!) to the primitive letters of that language. We will use the letter 'v' to range over valuations. Now we define a satisfaction relation `sat` between valuations and complex expressions.

Definition 1 *A complex expression ϕ might be a propositional letter and if it is then* $\mathtt{sat}(v, \phi)$ *is just $v(\phi)$, the result of applying v to ϕ;*

If ϕ is the conjunction of ψ_1 and ψ_2 then $\mathtt{sat}(v, \phi)$ *is* $\mathtt{sat}(v, \psi_1) \wedge \mathtt{sat}(v, \psi_2)$*;*

If ϕ is the disjunction of ψ_1 and ψ_2 then $\mathtt{sat}(v, \phi)$ *is* $\mathtt{sat}(v, \psi_1) \vee \mathtt{sat}(v, \psi_2)$*;*

If ϕ is the conditional whose antecedent is ψ_1 and whose consequent is ψ_2 then $\mathtt{sat}(v, \phi)$ *is* $\mathtt{sat}(v, \psi_1) \rightarrow \mathtt{sat}(v, \psi_2)$*;*

If ϕ is the negation of ψ_1 then $\mathtt{sat}(v, \phi)$ *is* $\neg\mathtt{sat}(v, \psi_1)$ *;*

If ϕ is the biconditional whose two immediate subformulæ are ψ_1 and ψ_2 then $\mathtt{sat}(v, \phi)$ *is* $\mathtt{sat}(v, \psi_1) \longleftrightarrow \mathtt{sat}(v, \psi_2)$*.*

Notice that here i am using the letters 'ϕ' and 'ψ_1' and 'ψ_2' as variables that range over formulæ, as in the form of words "If ϕ is the conjunction of ψ_1 and ψ_2 then ...". They are not abbreviations of formulæ. There is a temptation to write things like

"If ϕ is $\psi_1 \wedge \psi_2$ then $\mathtt{sat}(v, \phi)$ is $\mathtt{sat}(v, \psi_1) \wedge \mathtt{sat}(v, \psi_2)$"

or perhaps

$$\mathtt{sat}(v, \psi_1 \wedge \psi_2) \text{ is } \mathtt{sat}(v, \psi_1) \wedge \mathtt{sat}(v, \psi_2) \tag{1.1}$$

Now although our fault-tolerant pattern matching enables us to see immediately what is intended, the pattern matching does, indeed, need to be fault-tolerant. (In fact it corrects the fault so quickly that we tend not to notice the processing that is going on.)

In an expression like '$\mathtt{sat}(v, \phi)$' the 'ϕ' has to be a name of a formula, as we noted above, not an abbreviation for a formula. But then how are we to make sense of

$$\mathtt{sat}(v, \psi_1 \wedge \psi_2) \tag{1.2}$$

The string '$\psi_1 \wedge \psi_2$' has to be the name of formula. Now you don't have to be The Brain of Britain to work out that it has got to be the name of whatever formula it is that we get by putting a '$\wedge$' between the two formulæ named by 'ψ_1' and 'ψ_2'—and this is what your fault-tolerant pattern-matching wetware (supplied by Brain-Of-Britain) will tell you. But we started off by making a fuss about the fact that names have no internal structure, and now we suddenly find ourselves wanting names to have internal structure after all!

In fact there is a way of making sense of this, and that is to use corner quotes to create an environment wherein compounds of names of formulæ (composed with connectives) name composites (composed by means of those same connectives) of the formulæ named

So 1.1 would be OK if we write it as

$$\mathtt{sat}(v, \ulcorner \psi_1 \wedge \psi_2 \urcorner) \text{ is } \mathtt{sat}(v, \psi_1) \wedge \mathtt{sat}(v, \psi_2) \tag{1.3}$$

Corner quotes were first developed in [3]. See pp 33-37. An alternative way of proceding that does not make use of corner quotes is instead to use an entirely new suite of symbols—as it might be '`and`' and '`or`' and so on, and setting up

links between them and the connectives '$\wedge$' and so on in the object language so that—for example

$$\psi_1 \texttt{ and } \psi_2 \tag{A}$$

is the conjunction of ψ_1 and ψ_2. The only drawback to this is the need to conjure up an entire suite of symbols, all related suggestively to the connectives they are supposed to name. Here one runs up against the fact that any symbols that are suitably suggestive will also be laden with associations from their other uses, and these associations may not be helpful. Suppose we were to use an ampersand instead of '`and`'; then the fact that it is elsewhere used instead of '$\wedge$' might cause the reader to assume it is just a synonym for '$\wedge$'. There is no easy way through.

Allude to Quine: ML

1.2 Eager and Lazy Evaluation

The recursive definition of `sat` in the previous section gives us a way of determining what truth-value a formula receives under a valuation. Start with what the valuation does to the propositional letters (the leaves of the parse tree) and work up the tree. Traditionally the formal logic that grew up in the 20th century took no interest in how things like $\texttt{sat}(\phi, v)$ was actually *calculated*. The recursive definition tells us uniquely what the answer must be but it doesn't tell us uniquely how to calculate it.

The way of calculating $\texttt{sat}(\phi, v)$ that we have just seen (start with what the valuation does to the propositional letters—the leaves of the parse tree—and work up the tree) is called **Eager evaluation** also known as **Strict evaluation**. But there are other ways of calculating that will give the same answer. One of them is the beguilingly named **Lazy evaluation** which we will now describe.

Consider the project of filling out a truth-table for the formula $A \wedge (B \vee (C \wedge D))$. One can observe immediately that any valuation (row of the truth-table) that makes 'A' false will make the whole formula false:

A	$\wedge$	$(B$	$\vee$	$(C$	$\wedge$	$D))$
0		0		0		0
0		0		0		1
0		0		1		0
0		0		1		1
0		1		0		0
0		1		0		1
0		1		1		0
0		1		1		1
1		0		0		0
1		0		0		1
1		0		1		0
1		0		1		1
1		1		0		0
1		1		0		1
1		1		1		0
1		1		1		1

A	$\wedge$	$(B$	$\vee$	$(C$	$\wedge$	$D))$
0	0	0		0		0
0	0	0		0		1
0	0	0		1		0
0	0	0		1		1
0	0	1		0		0
0	0	1		0		1
0	0	1		1		0
0	0	1		1		1
1		0		0		0
1		0		0		1
1		0		1		0
1		0		1		1
1		1		0		0
1		1		0		1
1		1		1		0
1		1		1		1

Now, in the remaining cases we can observe that any valuation that makes 'B' true will make the whole formula true :

A	$\wedge$	$(B$	$\vee$	$(C$	$\wedge$	$D))$
0	0	0		0		0
0	0	0		0		1
0	0	0		1		0
0	0	0		1		1
0	0	1		0		0
0	0	1		0		1
0	0	1		1		0
0	0	1		1		1
1		0		0		0
1		0		0		1
1		0		1		0
1		0		1		1
1		1	1	0		0
1		1	1	0		1
1		1	1	1		0
1		1	1	1		1

A	$\wedge$	$(B$	$\vee$	$(C$	$\wedge$	$D))$
0	0	0		0		0
0	0	0		0		1
0	0	0		1		0
0	0	0		1		1
0	0	1		0		0
0	0	1		0		1
0	0	1		1		0
0	0	1		1		1
1		0		0		0
1		0		0		1
1		0		1		0
1		0		1		1
1	1	1	1	0		0
1	1	1	1	0		1
1	1	1	1	1		0
1	1	1	1	1		1

In the remaining cases any valuation that makes 'C' false will make the whole formula false.

A	$\wedge$	$(B$	$\vee$	$(C$	$\wedge$	$D))$
0	0	0		0		0
0	0	0		0		1
0	0	0		1		0
0	0	0		1		1
0	0	1		0		0
0	0	1		0		1
0	0	1		1		0
0	0	1		1		1
1	0	0	0	0	0	0
1	0	0	0	0	0	1
1		0		1		0
1		0		1		1
1	1	1	1	0		0
1	1	1	1	0		1
1	1	1	1	1		0
1	1	1	1	1		1

A	$\wedge$	$(B$	$\vee$	$(C$	$\wedge$	$D))$
0	0	0		0		0
0	0	0		0		1
0	0	0		1		0
0	0	0		1		1
0	0	1		0		0
0	0	1		0		1
0	0	1		1		0
0	0	1		1		1
1	0	0	0	0	0	0
1	0	0	0	0	0	1
1	0	0	0	1	0	0^*
1	1	0	1	1	1	1^*
1	1	1	1	0		0
1	1	1	1	0		1
1	1	1	1	1		0
1	1	1	1	1		1

The starred '0^*' and '1^*' are the only cases where we actually have to look at the truth-value of D.

These illustrations concern evaluation in languages whose expressions evaluate to truth-values. The idea originally arose in connection with langauges whose expressions evaluate to numbers or other data objects.

`if` $x \geq 0$ `then` $f(x)$ `else` $g(x)$.

Of course you evaluate lazily. No point in calculating both $f(x)$ and $g(x)$ when you are clearly going to need only one of them! First you evaluate x to see whether it is above or below 0 and then you do whichever of $f(x)$ and $g(x)$ that it turns out you need.

Notice also in this connection that i might not have to evaluate x *completely* in order to know which way to jump. If x is presented to me as a double-precision decimal number i have 12 decimal places to evaluate, but i will know already after evaluating the first of them whether x is positive or negative.

(The difference between functional and declarative programming languages is that a functional programming language allows you to define a function, whereas a declarative one allows you to tell the computer how to evaluate it.

Eager and strict give the same result beco's all parse trees are finite and have no loops. We prove this by structural induction

1.3 Natural Deduction and Sequent Calculus

1.3.1 The Rules of Natural Deduction

In the following table we see that for each connective we have two rules: one to introduce the connective and one to eliminate it. These two rules are called the **introduction rule** and the **elimination rule** for that connective. Richard Bornat calls the elimination rules "use" rules because the elimination rule for a connective $\mathcal{C}$ tells us how to **use** the information wrapped up in a formula whose principal connective is $\mathcal{C}$.

(The idea that everything there is to know about a connective can be captured by an elimination rule plus an introduction rule has the same rather operationalist flavour possessed by the various *meaning is use* doctrines one encounters in philosophy of language. In this particular form it goes back to Prawitz, and possibly to Gentzen and Jaskowski.) references?

The rules tell us how to use the information contained in a formula (Some of these rules come in two parts.)

$\vee$-int: $\frac{A}{A \vee B}$; $\frac{B}{A \vee B}$;	$\vee$-elim(1): $\dfrac{A \vee B \qquad \overset{[A]^1}{\vdots}\ C \qquad \overset{[B]^1}{\vdots}\ C}{C}$
$\wedge$: $\frac{A \quad B}{A \wedge B}$;	$\wedge$-elim: $\frac{A \wedge B}{A}$; $\frac{A \wedge B}{B}$
$\rightarrow$-int(1) $\dfrac{\overset{[A]^1}{\vdots}\ B}{A \rightarrow B}$	$\rightarrow$-elim: $\frac{A \quad A \rightarrow B}{B}$
Ex falso sequitur quodlibet; $\frac{\bot}{A}$	Double negation $\frac{\neg\neg A}{A}$

Some small print:

N.B.: in $\rightarrow$-introduction you don't have to cancel all occurrences of the premiss: it is perfectly all right to cancel only some of them .

The Latin expression *ex falso ...* means: "From the `false` follows whatever you like".

Some of these rules look a bit daunting so let's start by cutting our teeth on some easy ones.

Exercise 2 *Using just the two rules for $\wedge$, the rule for $\vee$-introduction and $\rightarrow$-elimination see what you can do with each of the following sets of formulæ:*[1]

1. A, $A \rightarrow B$;
2. A, $A \rightarrow (B \rightarrow C)$;
3. A, $A \rightarrow (B \rightarrow C)$, B;
4. A, B, $(A \wedge B) \rightarrow C$;
5. A, $(A \vee B) \rightarrow C$;
6. $A \wedge B$, $A \rightarrow C$;
7. $A \wedge B$, $A \rightarrow C$, $B \rightarrow D$;

[1] Warning: in some cases the answer might be "nothing!"

8. $A \to (B \to C)$, $A \to B$, $B \to C$;

9. A, $A \to (B \to C)$, $A \to B$;

You will probably notice in doing these questions that you use one of your assumptions more than once, and indeed that you have to *write it down* more than once (= write down more than one token!) This is particularly likely to happen with $A \wedge B$. If you need to infer both of A and B then you will have to write out '$A \wedge B$' *twice*—once for each application of $\wedge$-elimination.

If you try writing down only one token you will find that you want your sheet of paper to be made of lots of plaited ribbons. Ugh.

The two rules of *ex falso* and *double negation* are the only rules that specifically mention negation. Recall from p. **??** that $\neg B$ is $B \to \bot$, so the inference

$$\frac{A \qquad \neg A}{\bot} \tag{1.1}$$

—which *looks* like a new rule—is merely an instance of $\to$-elimination.

Finally we need the identity rule:

$$\frac{A\ B\ C \dots}{A} \tag{1.2}$$

(where the list of extra premisses may be empty) which records the fact that we can deduce A from A. Not very informative, one might think, but it turns out to be useful. After all, how else would one obtain a proof of the undoubted tautology $A \to (B \to A)$, otherwise known as 'K'? One could do something like

$$\dfrac{\dfrac{\dfrac{\dfrac{[A]^2 \qquad [B]^1}{A \wedge B}\wedge\text{-int}}{A}\wedge\text{-elim}}{B \to A}\to\text{-int }(1)}{A \to (B \to A)}\to\text{-int }(2) \tag{1.3}$$

but that is grotesque: it uses a couple of rules for a connective that doesn't even appear in the formula being proved! The obvious thing to do is

$$\dfrac{\dfrac{\dfrac{[A]^2 \qquad [B]^1}{A}\text{identity rule}}{B \to A}\to\text{-int }(1)}{A \to (B \to A)}\to\text{-int }(2) \tag{1.4}$$

If we take seriously the observation above concerning the rule of $\to$-introduction—namely that you are not required to cancel *all* occurrences of an assumption—then you infer that you can cancel none of them, and that suggests that you can cancel assumptions that aren't there—then we will not need this rule. This means we can write proofs like 1.5 below. To my taste, it seems less bizarre to discard assumptions than it is to cancel assumptions that aren't there, so I prefer 1.4 to 1.5. It's a matter of taste.

$$\dfrac{\dfrac{[A]^1}{B \to A}\to\text{-int}}{A \to (B \to A)}\to\text{-int }(1) \tag{1.5}$$

It is customary to connect the several occurrences of a single formula at introductions (it may be introduced several times) with its occurrences at elimination by means of superscripts. Square brackets are placed around eliminated formulæ, as in the formula displayed above.

There are funny logics where you are not allowed to use an assumption more than once: in these **resource logics** assumptions are like sums of money. (You will find them in section **??** if you last that long). This also gives us another illustration of the difference between an argument (as in logic) and a debate (as in rhetoric). In rhetoric it may happen that a point, albeit a good point, can be usefully made only once . . . in an ambush perhaps.

Do some very simple illustrations of compound proofs here

1.3.2 What do the rules *mean*??

One way in towards an understanding of what the rules do is to dwell on the point made by my friend Richard Bornat that elimination rules are **use** rules:

The rule of →-elimination

The rule of →-elimination tells you how to use the information wrapped up in '$A \to B$'. '$A \to B$' informs us that if A, then B. So the way to use the information is to find yourself in a situation where A holds. You might not be in such a situation, and if you aren't you might have to assume A with a view to using it up later—somehow. We will say more about this.

The rule of ∨-elimination

The rule of ∨-elimination tells you how to **use** the information in '$A \vee B$'. If you are given $A \vee B$, how are you to make use of this information without supposing that you know which of A and B is true? Well, **if** you know you can deduce C from A, and you ALSO know that you can deduce C from B, **then** as soon as you are told $A \vee B$ you can deduce C. One could think of the rule of ∨-elimination as a function that takes (1) $A \vee B$, (2) a proof of C from A and (3) a proof of C from B, and returns a proof of C from $A \vee B$. This will come in useful on page 18.

Here is an example, useful to those of you who fry your brains doing sudoku

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

There is a '5' in the top right-hand box—somewhere. But in which row? The '5' in the top left-hand box must be in the first column, and in one of the top two rows. The '5' in the fourth column must be in one of the two top cells. (It cannot be in the fifth row because there is already a '5' there, and it cannot be in the last three rows because that box already has a '5' in it.) So the '5' in the middle box on the top must be in the first column, and in one of the top two rows. These two '5's must of course be in different rows. So where is the five in the rightmost of the three top boxes? Either the '5' in the left box is on the first row and the '5' in the middle box is on the second row or the 5 in the middle box is on the first row and the '5' in the left box is on the second row. We don't know which of the possibilities is the true one, but it doesn't matter: either way the '5' in the rightmost box must be in the bottom (third) row.

There is a more general form of ∨-elimination:

$$\dfrac{\begin{array}{ccccc}[A_1]^1 & [A_2]^1 & \dots & [A_n]^1 & \\ \vdots & \vdots & & \vdots & \\ C & C & & C & A_1 \vee A_2 \vee \dots A_n\end{array}}{C} \vee\text{-elim } (1) \tag{1.1}$$

where we can cancel more than one assumption. That is to say we have a *set* $\{A_1 \dots A_n\}$ of assumptions, and the rule accepts as input a list of proofs of C: one proof from A_1, one proof from A_2, and so on up to A_n. It also accepts the disjunction $A_1 \vee \dots A_n$ of the set $\{A_1 \dots A_n\}$ of assumptions, and it outputs a proof of C.

The rule of $\vee$-elimination is a hard one to grasp so do not panic if you don't get it immediately. However, you should persist until you do.

1.3.3 Goals and Assumptions

When you set out to find a proof of a formula, that formula is your **goal**. As we have just mentioned, the obvious way to attack a goal is to see if you can obtain it as the output of (a token of) the introduction rule for its principal connective. If that introduction rule is $\rightarrow$-introduction then this will generate an **assumption**. Once you have generated an assumption you will need—sooner or later—to extract the information it contains and you will do this by means of the *elimination* rule for the principal connective of that assumption. It's actually idiotically simple:

1. Attack a **goal** with the introduction rule for its principal connective;
2. Attack an **assumption** with the elimination rule for its principal connective.

Consider (1). We have the goal $((A \rightarrow B) \rightarrow A) \underline{\rightarrow} ((A \rightarrow B) \rightarrow B)$. The principal connective of this formula is the arrow in the middle that I underlined. So we **assume** the antecedent (which is $(A \rightarrow B) \rightarrow A$) and then the consequent (which is $(A \rightarrow B) \rightarrow B$) becomes our new goal. So we have traded the old goal $((A \rightarrow B) \rightarrow A) \underline{\rightarrow} ((A \rightarrow B) \rightarrow B)$ for the new goal $((A \rightarrow B) \rightarrow B)$ and generated the new assumption $((A \rightarrow B) \rightarrow A)$.

I have noticed that beginners often treat assumptions as if they were goals. Perhaps this is because they encounter goals first and they are *perseverating*. In the example of the preceding paragraph we generated the assumption $(A \rightarrow B) \rightarrow A$. How are you going to use this assumption? Do not attempt to *prove* it; you must *use* it! And the way to use it is to whack it with the elimination rule for its principal connective—which is $\rightarrow$. The only way you can do this is if you have somehow got hold of $A \rightarrow B$—and this gives you the new goal of $A \rightarrow B$

to be continued ...

Your first step—when challenged to find a natural deduction proof of a formula—should be to identify the principal connective. (That was the point of exercise **??**.) For example, when challenged to find a proof of $(A \wedge B) \rightarrow A$, the obvious gamble is to expect that the last step in the proof was a $\rightarrow$-introduction rule applied to a proof of A with the assumption $A \wedge B$.

1.3.4 The small print

It isn't always true that you should attack an assumption (or goal) with the elimination (introduction) rule for its main connective. It might be that the goal or assumption you are looking at is a propositional letter and therefore *does not have* a principal connective! In those circumstances you have to try

something else. Your assumption might be P and if you have in your knapsack the formula $(P \vee Q) \to R$ it might be a good idea to whack the 'P' with a $\vee$-introduction to get $P \vee Q$ so you can then do a $\to$-elimination and get R. And of course you might wish to refrain from attacking your assumption with the elimination rule for its principal connective. If you assumption is $P \vee Q$ and you already have in your knapsack the formula $(P \vee Q) \to R$ you'd be crazy not to use $\to$-elimination to get R. And in so doing you are not using the elimination rule for the principal connective of $P \vee Q$.

And even when a goal or assumption does have a principal connective attacking it with their appropriate rule for that principal connective is not absolutely *guaranteed* to work. Consider the task of finding a proof of $A \vee \neg A$. (A here is a propositional letter, not a complex formula). If you attack the principal connective you will of course use $\vee$-int and generate the attempt

$$\frac{A}{A \vee \neg A} \vee\text{-int} \tag{1.1}$$

or the attempt

$$\frac{\neg A}{A \vee \neg A} \vee\text{-int} \tag{1.2}$$

and clearly neither of these is going to turn into a proof of $A \vee \neg A$, since we are not going to get a proof of A (nor a proof of $\neg A$). It turns out you have to use the rule of double negation. assume $\neg(A \vee \neg A)$ and get a contradiction. There is a pattern to at least some of these cases where attacking-the-principal-connective is not the best way forward, and we will say more about it later.

The moral of this is that finding proofs is not a simple join-up-the-dots exercise: you need a bit of ingenuity at times. Is this because we have set up the system wrongly? Could we perhaps devise a system of rules which was completely straightforward, and where short tautologies had short proofs[2] which can be found by blindly following rules like always-use-the-introduction-rule-for-the-principal-connective-of-a-goal? You might expect that, the world being the kind of place it is, the answer is a resounding 'NO!' but curiously the answer to this question is not known. I don't think anyone expects to find such a system, and i know of no-one who is trying to find one, but the possibility has not been excluded.

Connection with P=NP. NP = co-NP

[If ϕ is not a propositional tautology we can find a falsifying valuation deterministically in exponential time or nondeterministically in polynomial time. It's not at all clear whether there can be a proof system for propositional logic which will, in polynomial time, exhibit a proof if there is one.]

Get these in something like increasing order of difficulty

Exercise 3 *Find natural deduction proofs of the following tautologies:*

1. $(P \to Q) \to ((Q \to R) \to (P \to R))$;
2. $(A \to C) \to ((A \wedge B) \to C)$;
3. $((A \vee B) \to C) \to (A \to C)$;
4. $P \to (\neg P \to Q)$;
5. $A \to (A \to A)$ *(you will need the identity rule);*
6. $(((P \to Q) \to Q) \to Q) \to (P \to Q)$;
7. $((A \to B) \to A) \to ((A \to B) \to B)$;
8. $A \to ((((A \to B) \to B) \to C) \to C)$;

[2] 'short' here can be given a precise meaning

9. $(P \vee Q) \to (((P \to R) \wedge (Q \to S)) \to (R \vee S))$;
10. $(P \wedge Q) \to (((P \to R) \vee (Q \to S)) \to (R \vee S))$;
11. $\neg(A \vee B) \to (\neg A \wedge \neg B)$;
12. $A \vee \neg A$; (*)
13. $\neg(A \wedge B) \to (\neg A \vee \neg B)$; (hard!) (*)
14. $(A \wedge (B \vee C)) \to ((A \wedge B) \vee (A \wedge C))$;
15. $(A \vee (B \wedge C)) \to ((A \vee B) \wedge (A \vee C))$;
16. $A \to [(A \to C) \to ((B \to C) \to C)]$; *(for this and the next you will need the identity rule);*
17. $B \to [(A \to C) \to ((B \to C) \to C)]$; *(then put these last two together to obtain a proof of*
18. $(A \vee B) \to [(A \to C) \to ((B \to C) \to C)]$;
19. $(B \vee (B \to A)) \to A \to A$;
20. $(A \wedge B) \vee (A \wedge \neg B) \vee (\neg A \wedge B) \vee (\neg A \wedge \neg B)$. *(Hard! For enthusiasts only) (*)*

You should be able to do the first eight without breaking sweat. If you can do the first dozen without breaking sweat you may feel satisfied. The starred items will need the rule of double negation. For the others you should be able to find proofs that do not use double negation. The æsthetic into which you are being inducted is one that says that proofs that do not use double negation are always to be preferred to proofs that do. Perhaps it is a bit belittling to call it an æsthetic: there is a principled philosophical position that denies the rule of double negation, and one day you might want to engage with it.

Enthusiasts can also attempt the first two parts of exercise 23 on p. 36: they are like the exercises here but harder.

If you want to get straight in your mind the samll print around the $\to$-introduction rule you might like to try the next exercise. In one direction you willl need to cancel two occurence of an assumption, and in the other you will need the identity rule, which is to say you will need to cancel zero occurences of the assumption.

Exercise 4

1. *Provide a natural deduction proof of $A \to (A \to B)$ from $A \to B$;*
2. *Provide a natural deduction proof of $A \to B$ from $A \to (A \to B)$.*

To make quite sure you might like to try this one too

Exercise 5

1. *Provide a natural deduction proof of $A \to (A \to (A \to B))$ from $A \to B$;*
2. *Provide a natural deduction proof of $A \to B$ from $A \to (A \to (A \to B))$.*

Exercise 6 *Life is complicated on Planet Zarg. The Zarglings believe there are three truth-values:* `true`, `intermediate` *and* `false`. *Here we write them as 1, 2 and 3 respectively. Here is the truth-table for the connective $\to$ on planet Zarg:*

P	Q	$P \to Q$
1	*1*	*1*
1	*2*	*2*
1	*3*	*3*
2	*1*	*1*
2	*1*	*2*
2	*3*	*3*
3	*1*	*1*
3	*1*	*2*
3	*1*	*3*

On Zarg the truth-value of $P \vee Q$ is simply the smaller of the truth-values of P and Q; the truth-value of $P \wedge Q$ is the larger of the truth-values of P and Q.

Write out Zarg-style truth-tables for

1. $P \vee Q$;
2. $P \wedge Q$;
3. $((P \to Q) \to P) \to P$;
4. $P \to (Q \to P)$;
5. $(P \to Q) \to Q)$;

[Brief reality check: What is a tautology on Planet Earth?]

What might be a good definition of tautology *on Planet Zarg?*

According to your definition of a tautology-on-planet-Zarg, is it the case that if P and Q are formulæ such that P and $P \to Q$ are both tautologies, then Q is a tautology?

There are two possible negations on Zarg:

P	$\neg^1 P$	$\neg^2 P$
1	3	3
2	2	1
3	1	1

Given that the Zarglings believe $\neg(P \wedge \neg P)$ to be a tautology, which negation do they use?

Using that negation, do they believe the following formulæ to be tautologies?

1. $P \vee \neg P$?
2. $(\neg\neg P) \vee \neg P$?
3. $\neg\neg(P \vee \neg P)$?
4. $(\neg P \vee Q) \to (P \to Q)$?

Exercise 7 *Annotate the following proofs, indicating which rules are used where and which premisses are being cancelled when.*

$$\dfrac{\dfrac{\dfrac{P \qquad P \to Q}{Q}}{(P \to Q) \to Q}}{P \to ((P \to Q) \to Q)} \tag{1.3}$$

$$\dfrac{\dfrac{\dfrac{P \wedge Q}{Q}}{P \vee Q}}{(P \wedge Q) \to (P \vee Q)} \tag{1.4}$$

$$\dfrac{\dfrac{P \qquad \neg P}{\bot}}{P \to Q} \tag{1.5}$$

$$\dfrac{P \vee Q \qquad \dfrac{P \qquad P \to R}{R} \qquad \dfrac{Q \qquad Q \to R}{R}}{\dfrac{R}{(P \vee Q) \to R}} \tag{1.6}$$

$$\dfrac{\dfrac{\dfrac{A \qquad B}{A \wedge B}}{B \to (A \wedge B)}}{A \to (B \to (A \wedge B))} \tag{1.7}$$

$$\dfrac{\dfrac{\dfrac{(A \to B) \to B \qquad A \to B}{B}}{((A \to B) \to B) \to B}}{(A \to B) \to (((A \to B) \to B) \to B)} \tag{1.8}$$

Exercise 8 *Now find sequent proofs for the formulæ in exercise 3 (page 11). For the starred formulæ you should expect to have to have two formulæ on the right at some point.*

Be sure to annotate your proofs by recording at each step which rule you are using. That makes it easier for you to check that you are constructing the proofs properly.

Exercise 9 *Find a proof of the sequent:*

$$(A \to B) \to B \vdash (B \to A) \to A$$

Exercise 10 *Go back to Zarg (exercise 6 p. 12)*

1. *Using the truth-table for $\neg$ that you decided that the Zarglings use—check that the Zarglings do not believe axiom T to be a tautology.* $(\neg A \to B) \to ((\neg A \to \neg B) \to A)$
2. *Do the Zarglings believe S to be a tautology?*

The empty conjunction and the empty disjunction

Since a conjunction or disjunction can have more than two disjuncts, it's worth asking if it can have fewer...

As we have just seen, '$\vee$' and '$\wedge$' have uppercase versions '$\bigvee$' and '$\bigwedge$' that can be applied to sets of formulæ: $\bigvee\{p, q\}$ is obviously the same as $p \vee q$ for example, and $\bigwedge\{p, q\}$ is $p \wedge q$ by the same token.

Slightly less obviously $\bigwedge\{p\}$ and $\bigvee\{p\}$ are both p. But what is $\bigvee\emptyset$? (the disjunction of the empty set of formulæ). Does it even make sense? Yes it does, and if we are brave we can even calculate what it is.

If X and Y are sets of formulæ then $\bigvee(X \cup Y)$ had better be the same as $\bigvee X \vee \bigvee Y$. Now what if Y is $\emptyset$, the empty set? Then

$$\bigvee X$$
$$= \bigvee(X \cup \emptyset)$$

(because $X = X \cup \emptyset$)

$$= (\bigvee X) \vee (\bigvee \emptyset)$$

so

$$(\bigvee X) \vee (\bigvee \emptyset) \;=\; (\bigvee X) \tag{1.9}$$

and this has got to be true for all sets X of formulæ. This compels $\bigvee \emptyset$ to be the `false`. If it were anything else then a situation might arise in which the left-hand side of (1.9) were true and the right-hand-side false.

Similarly $\bigwedge \emptyset$ is the `true`.

In general if $*$ is an (associative) operation $* : V \times V \to V$ then $*$ is defined on lists and $*$ of the empty list will be the unit for $*$. (You will no doubt remember from your early education that $x^0 = 1$. Equally you can explain why this is correct!)

Thus the sum of the empty set of (naturals, reals, complexes ...) will be 0 (of the appropriate type); the product of the empty set of (naturals, reals, complexes ...) will be 1 (of the appropriate type).

1.4 CNF and DNF

Definition of CNF and DNF. Normal form theorem: every formula is equivalent to something in CNF and also to something in DNF. We prove this using the de Morgan identities.

Exercise 11 *Put the formula "*`if` p `then` q `else` r*" into DNF and also into DNF.*

Exercise 12 *The* **dual** $\widehat{A}$ *of a propositional formula A is the result of replacing every propositional letter in A by its negation.*

(Thus if p and q are letters then $\widehat{p \vee q}$ is $\neg p \vee \neg q$; $\widehat{\neg p \wedge q}$ is $p \wedge \neg q$ and so on. Notice that $\widehat{A}$ is typically not logically equivalent to $\neg A$! A formula A is **self-dual** *if it is logically equivalent to its own dual: that is to say that $A \longleftrightarrow \widehat{A}$ is a tautology. For example: p* `XOR` *q is self-dual—'p' and 'q' being literals—even tho' A* `XOR` *B is not self-dual in general.)*

A `XOR` *$(B$* `XOR` *$C)$ is not self-dual: it is in fact dual to its negation. But $(A$* `XOR` *$D)$* `XOR` *$(B$* `XOR` *$C)$ is self-dual.*

1. *Show that the hat commutes with all connectives:*
   $\widehat{A \vee B} \longleftrightarrow (\widehat{A} \vee \widehat{B})$
   $\widehat{A \wedge B} \longleftrightarrow (\widehat{A} \wedge \widehat{B})$
   $\widehat{A \to B} \longleftrightarrow (\widehat{A} \to \widehat{B})$
   $\widehat{\neg A} \longleftrightarrow \neg \widehat{A}$

2. *Show that any propositional formula of the form $A \longleftrightarrow \widehat{A}$ is self-dual;*

3. *Show that if A is a self-dual formula so is $\neg A$;*

4. *Show that whenever A is a self-dual formula there is a formula B such that A is logically equivalent to $B \longleftrightarrow \widehat{B}$; In how many ways can this be done?*

5. *Is there a similar result for Predicate Calculus?*

1.4.1 CNF and Resolution

Let us say that a **clause** is a disjunction of atomics and negatomics. The theorem that says that every formula is equivalent to one in CNF says that every expression is equivalent to a conjunction of clauses. If we have a finite set of assumptions in a propositional language we can put each assumption into CNF and thereby obtain a set of cluases which is logically equivalent to our initial set of assumptions. This is an advantageous move because clauses are much easier things to work with than are arbitrary formulæ. In particular we can use a rule commonly called **resolution**. Suppose $\mathcal{A}$ is a clause one of whose disjuncts is the letter P, and $\mathcal{B}$ is a clause one of whose disjuncts is the negatomic $\neg P$. We can think of $\mathcal{A}$ and $\mathcal{B}$ as $A \vee P$ and $B \vee \neg P$ respectively. When we think of them like that, we can see immediately that we can infer $A \vee B$ from them.

Being a real pedant i cannot help noticing that the conclusion $A \vee B$ that we have inferred isn't a clause (unless both A and B are atomics-or-negatomics). We would need to flatten it in order to obtain a clause. So the appropriate data structure for inputs to the resolution rule is **set** (each input should be a set of atomics-and-negatomics) rather than **formula** (a disjunction of atomics-and-negatomics).

1.5 Soundness and completeness of the natural deduction rules

The rules of natural deduction are **sound**: every formula we can prove using natural deduction is a tautology. The rules preserve truth: if you reason using these rules from true premisses your conclusions will be true as well. Whatever our logical machinery (and it might be deliberately over-simplified, as it is when we start off with propositional logic) we want to be sure that the rules that we decide on for reasoning with that machinery are sound in this sense.

Completeness is a feature complementary to soundness. Not only are the rules sound, but they exhaust the possible modes of truth-preserving reasoning (in this language) in the sense that any truth-preserving inference can be captured by reasoning according to these formulations. We say the rules are **complete**. We prove this in section **??**. It is impossible to overstate the significance of this fact. There is a finite system of rules of inference which captures **all** truth-preserving reasoning expressible in this syntax. The power of this simplification is incalculable and has impressed generations of ogicians. There is a tradition in modern logic that holds that a body of principles of reasoning that cannot be finitely codified is simply not part of Logic at all. Not everybody believes this, but it is a widely held view.

In the case of propositional logic we have truth-tables, which enable us to decide quite quickly when a formula is valid (or when a principle of reasoning is truth-preserving aka sound). This is so convenient that one tends to forget that there is actually a method of *generating* all the valid principles (and all the tautologies aka valid formulæ) over and above a method of recognising them when they pop up. In fact there are several ways of doing this, and we will see some of them, and we will prove that they do this: that is, that they are complete.

The rules are sound in that they preserve truth: in any token of the rule if the premisses are true then the conclusions are true too. For the rules like $\wedge$-introduction, $\vee$-introduction, $\wedge$-elimination, $\rightarrow$-elimination ... it's obvious what is meant: for any valuation v if the stuff above the line is true according to v then so is the stuff below the line.

What I am planning to convince you is that any complex proof made up by

composing lots of tokens of $\wedge$-int, $\rightarrow$-elim and so on has the property that any valuation making all the premisses true also makes the conclusion true. That is to say, we claim that all complex proofs are **truth-preserving**. Notice that this has as a special case the fact that any complex proof with no premisses has a conclusion that is logically valid. Every valuation making all the premisses true will make the conclusion true. Now since there are no premisses, every valuation makes all the premisses true, so every valuation makes the conclusion true. So the conclusion is valid! Rephrase this

However this way of thinking about matters doesn't enable us to make sense of $\rightarrow$-introduction and $\vee$-elimination. To give a proper description of what is going on we need to think of the individual (atomic) introduction and elimination rules as gadgets for making new complex proofs out of old (slightly less complex) proofs.

That is to say you think of the rule of $\wedge$-introduction as a way of taking a complex proof $\mathcal{D}_1$ of A and a complex proof $\mathcal{D}_2$ of B and giving a complex proof $\mathcal{D}_3$ of $A \wedge B$. We are trying to show that all complex deductions are truth-preserving.

The fact that $\wedge$-introduction is truth-preserving in the sense of the previous paragraph now assures us that it has the new property that:

If

- $\mathcal{D}_1$ is a truth-preserving deduction of A (that is to say, any valuation making the premisses of $\mathcal{D}_1$ true makes A true); and
- $\mathcal{D}_2$ is a truth-preserving deduction of B (that is to say, any valuation making the premisses of $\mathcal{D}_2$ true makes A true);

Then
the deduction $\mathcal{D}_3$:

$$\dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2 \\ \vdots & \vdots \\ A & B\end{array}}{A \wedge B}\ \wedge\text{-int} \tag{1.1}$$

too, is truth-preserving in the sense that any valuation making the premisses of $\mathcal{D}_3$ true—and they are just (the premisses of $\mathcal{D}_1$) $\cup$ (premisses of $\mathcal{D}_2$)—makes $A \wedge B$ true too.

This sounds like a much more complicated way of thinking of $\wedge$-introduction as truth-preserving than the way we started out with, but we need this way of seeing things when we come to consider the rules that involve cancelling assumptions, namely $\rightarrow$-introduction and $\vee$-elimination. Let us now consider these two.

$\rightarrow$-introduction

Suppose we have a deduction $\mathcal{D}$ of B from A, $C_1 \ldots C_n$, and that $\mathcal{D}$ is truth-preserving. That is to say, any valuation making all of A, $C_1 \ldots C_n$ true will also make B true. Now consider the deduction $\mathcal{D}'$ (of $A \rightarrow B$ from $C_1 \ldots C_n$) that is given us by an application of $\rightarrow$-introduction. We want this to be truth-preserving as well, that is to say, we want any valuation making $C_1 \ldots C_n$ true to make $A \rightarrow B$ true too. Let's check this. Let v be a valuation making $C_1 \ldots C_n$ true. Then either picture here

(i) it makes A true in which case—beco's $\mathcal{D}$ was truth-preserving—it makes B true as well and thereby makes $A \rightarrow B$ true

Or

> (ii) it makes A false. Any valuation making A false makes $A \to B$ true.

Remember: you don't have to cancel all occurrences of the premiss. (see page 7.)

$\vee$-elimination

We can tell a similar story about $\vee$-elimination. Suppose we have a truth-preserving deduction $\mathcal{D}_1$ of C from A (strictly: from A and a bag of extra assumptions like the $C_1 \ldots C_n$ of the previous paragraph) and a truth-preserving deduction $\mathcal{D}_2$ of C from B (and extra assumptions). That is to say that any valuation making A (and the extra assumptions) true makes C true, and any valuation making B (and the extra assumptions) true makes C true. Now, any valuation making $A \vee B$ (and the extra assumptions) true will make one of A and B true. So the new proof

$$\dfrac{\begin{array}{ccc}[A]^1 & [B]^1 & \\ \vdots & \vdots & \\ \mathcal{D}_1 & \mathcal{D}_2 & \\ \vdots & \vdots & \\ C & C & A \vee B\end{array}}{C}\ \vee\text{-elim } (1) \tag{1.2}$$

—that we make from $\mathcal{D}_1$ and $\mathcal{D}_2$ by applying $\vee$-elim to it—is truth-preserving as well.

In excruciating detail: let v be a valuation that makes $A \vee B$ (and the extra assumptions) true. Since v makes $A \vee B$ true, it must either (i) make A true, in which case we conclude that C must be true beco's of $\mathcal{D}_1$; or (ii) make B true, in which case we conclude that C must be true beco's of $\mathcal{D}_2$. Either way it makes C true.

1.6 Harmony and conservativeness

1.6.1 Conservativeness

Recall the discussion on page 8 about the need for the identity rule, and the horrendous proof of K that we would otherwise have, that uses the rules for $\wedge$.

Notice that the only proof of Peirce's Law that we can find uses rules for a connective ($\neg$, or $\bot$ if you prefer) that does not appear in the formula being proved. (Miniexercise: find a proof of Peirce's law). This rule is the rule of double negation of course. No-one is suggesting that this is illicit: it's a perfectly legal proof; however it does violate an æsthetic. (As does the proof of K that uses the rules for $\wedge$ instead of the identity rule). The æsthetic is *conservativeness*: every formula should have a proof that uses only rules for connectives that appear in the formula. Quite what the metaphysical force of this æsthetic is is a surprisingly deep question. It is certainly felt that one of the points in favour of the logic without the rule of double negation (which we will see more of below) is that it respects this æsthetic.

Plonk and tonk

The point of exercise 6 part 3 p. 12 was to establish that there can be no proof of Peirce's law using just the rules for '$\to$'.

See section 1.6.6

1.6.2 Harmony

A further side to this æsthetic is the thought that, for each connective, the introduction and elimination rule should complement each other nicely. What might this mean, exactly? Well, the introduction rule for a connective $\pounds$ tells us how to parcel up information in a way represented by the formula $A \pounds B$, and the corresponding elimination ("use"!) rule tells us how to use the information wrapped up in $A \pounds B$. We certainly don't want to set up our rules in such a way that we can somehow extract more information from $A \pounds B$ than was put into it in the first place. This would probably violate more than a mere æsthetic, in that it could result in inconsistency. But we also want to ensure that all the information that was put into it (by the introduction rules) can be extracted from it later (by the use rules). If our rules complement each other neatly in this way then something nice will happen. If we bundle information into $A \pounds B$ and then immediately extract it, we might as well have done nothing at all. Consider

$$\dfrac{\dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2 \\ \vdots & \vdots \\ A & B\end{array}}{A \wedge B}\,\wedge\text{-int}}{B}\,\wedge\text{-elim} \tag{1.1}$$

where we wrap up information and put it inside $A \wedge B$ and then immediately unwrap it. We can clearly simplify this to:

$$\begin{array}{c}\mathcal{D}_2 \\ \vdots \\ B\end{array} \tag{1.2}$$

This works because the conclusion $A \wedge B$ that we infer from the premisses A and B is the strongest possible conclusion we can infer from A and B and the premiss $A \wedge B$ from which we infer A and B is the *weakest* possible premiss which will give us both those conclusions. If we are given the $\wedge$-elimination rule, what must the introduction rule be? From $A \wedge B$ we can get both A and B, so we must have had to put them in in the first place when we were trying to prove $A \wedge B$ by $\wedge$-introduction. Similarly we can infer what the $\wedge$-elimination rule must be once we know the introduction rule.

The same goes for $\vee$ and $\rightarrow$. Given that the way to prove $A \rightarrow B$ is to assume A and deduce B from it, the way to use $A \rightarrow B$ must be to use it in conjunction with A to deduce B; given that the way to use $A \rightarrow B$ is to use it in conjunction with A to infer B it must be that the way to prove $A \rightarrow B$ is to assume A and deduce B from it. That is why it's all right to simplify

$$\dfrac{\dfrac{\begin{array}{c}[A] \\ \vdots \\ B\end{array}}{A \rightarrow B}\,\rightarrow\text{-int} \qquad A}{B}\,\rightarrow\text{-elim} \tag{1.3}$$

to

$$\begin{array}{c}A \\ \vdots \\ B\end{array} \tag{1.4}$$

And given that the way to prove $A \vee B$ is to prove one of A and B, the way to use $A \vee B$ must be to find something that follows from A and that also—separately—follows from B; given that the way to use $A \vee B$ is to find something

that follows from A and that also—separately and independently—follows from B, it must be that the way to prove $A \vee B$ is prove one of A and B. That is why we can simplify

$$\dfrac{\begin{array}{ccc}[A_1]^1 & [A_2]^1 & \\ \vdots & \vdots & \dfrac{A_1}{A_1 \vee A_2}\vee\text{-int} \\ C & C & \end{array}}{C}\vee\text{-elim }(1) \tag{1.5}$$

to

$$\begin{array}{c} A_1 \\ \vdots \\ C \end{array} \tag{1.6}$$

Definition 2 *We say a pair of introduction-plus-elimination rules for a connective £ is* **harmonious** *if (i) A£B is the strongest thing we can infer from the premisses for £-introduction and (ii) A£B is the weakest thing that (with the other premisses to the £-elimination rule, if any[3]) implies the conclusion of the £-elimination rule.*

What we have shown above is that the rules for $\rightarrow$, $\wedge$ and $\vee$ are harmonious.

1.6.3 Maximal Formulæ

. . . [*for enthusiasts only*!]

The first occurrence of '$A \rightarrow B$' in proof 1.3 above is a bit odd. It's the result of a $\rightarrow$-introduction and at the same time the (major) premiss of an $\rightarrow$-elimination. (We say such a formula is *maximal*.). That feature invites the simplification that we showed there. Presumably this can always be done? Something very similar happens with the occurrence of '$A_1 \vee A_2$' in proof 1.5. One might think so, but the situation is complex and not entirely satisfactory. One way into this is to try the following exercise:

Exercise 13 *Deduce a contradiction from the two assumptions $p \rightarrow \neg p$ and $\neg p \rightarrow p$. (These assumptions are of course really $p \rightarrow (p \rightarrow \bot)$ and $(p \rightarrow \bot) \rightarrow p$). Try to avoid having a maximal formula in your proof.*

1.6.4 Completeness

Completeness is harder. When we say that the system of rules of natural deduction is complete we mean that it provides proofs of every tautology.

A **row** is a conjunction of atomics and negatomics in which every propositional letter appears precisely once. There is an obvious correlation between rows and valuations.

For A a propositional formula let A^* be the disjunction of all the rows that make A come out true.

We write '$\vdash \phi$' for "there is a natural deduction proof of ϕ".

Lemma 3 *For all propositional formulæ A, there is a natural deduction proof of $A \longleftrightarrow A^*$.*

Proof:

By structural induction on formulæ. The base case concerns individual propositional letters and $\bot$, the false. If A is a propositional letter or $\bot$ then A^* is just A. Clearly $\vdash A \longleftrightarrow A^*$.

There is an induction step for each of $\wedge$, $\vee$ and $\rightarrow$.

[3] Do not forget that the elimination rule for £ might have premisses in addition to $A£B$: $\rightarrow$-elimination and $\vee$-elimination do, for example.

$\wedge$ $(A \wedge B)^*$ is the disjunction of those rows common to A^* and B^*, and is therefore interdeducible with $A^* \wedge B^*$. By induction hypothesis A is interdeducible with A^* and B is interdeducible with B^* so $A^* \wedge B^*$ (which we have just seen is interdeducible with $(A \wedge B)^*$) is interdeducible with $A \wedge B$.

$\vee$ $(A \vee B)^*$ is the disjunction of those rows appearing in the truth-table for $A \vee B$, and is therefore interdeducible with $A^* \vee B^*$. By induction hypothesis A is interdeducible with A^* and B is interdeducible with B^* so $A^* \vee B^*$ (which we have just seen is interdeducible with $(A \vee B)^*$) is interdeducible with $A \vee B$.

$\longrightarrow$ $(A \to B)^*$ is of course the disjunction of all rows that make A false or B true. We prove the two directions separately.

$\vdash (A \to B)^* \to (A \to B)$

Let r be one of the rows of $(A \to B)^*$.

(i) If r is a row that makes B true, then it is a disjunct of B^* so $\vdash r \to B^*$ whence $\vdash r \to B$ by induction hypothesis. So definitely $\vdash r \to (A \to B)$.

(ii) If r is a row that makes A false, then it is inconsistent with every row that makes A true, so it is inconsistent with their disjunction—which is A^*. A and A^* are interdeducible by induction hypothesis, so $\vdash r \to (A \to \bot)$. But $\vdash (A \to \bot) \to (A \to B)$, so $\vdash r \to (A \to B)$.

Either way, if r is a row of $(A \to B)$, $\vdash r \to (A \to B)$. $(A \to B)^*$ is the disjunction of all the rows of $A \to B$ so, by $\vee$-elimination, $\vdash (A \to B)^* \to (A \to B)$.

$\vdash (A \to B) \to (A \to B)^*$.

Assume $A \to B$ and $\neg(A \to B)^*$. We will deduce the false. $\neg(A \to B)^*$ denies every row in B^*, so refutes B^* and therefore refutes B (by induction hypothesis). $\neg B$ gives $\neg A$ by *modus tollens*. Now by induction bypothesis on A we can refute every disjunct in A^* (every row that makes A true). But our denial of $(A \to B)^*$ refuted every row that made A false. So we have refuted *all* rows! Recall that we can prove the disjunction of all the rows. $(A \vee \neg A)^*$ is provable. This gives us the contradiction we seek. Then we use the rule of classical negation to deduce $(A \to B)^*$. We now use $\to$-introduction to obtain a proof of $(A \to B) \to (A \to B)^*$.

■

We can now prove

THEOREM 4 *Every truth-table tautology has a natural deduction proof*

Proof: Suppose that A is a truth-table tautology. Observe that, if $a_1 \ldots a_n$ are the propositional letters that appear in A, then we can prove the disjunction of the 2^n rows to be had from $a_1 \ldots a_n$. We do this by induction on n. Since A is a truth-table tautology this disjunction is in fact A^*. Lemma 3 tells us that there is a natural deduction proof of $A \longleftrightarrow A^*$ so we conclude that there is a natural deduction proof of A. ■

If you are still less than 100% happy about this, attempt the following exercise:

Exercise 14

1. *Find a natural deduction proof of $A \vee \neg A$ (case $n = 1$);*
2. *Find a natural deduction proof of $(A \wedge B) \vee (\neg A \vee B) \vee (A \wedge \neg B) \vee (\neg A \vee \neg B)$ (case $n = 2$);*
3. *Explain the induction step.*

Admittedly this seems excessively laborious but the result is important even if the proof isn't. Important too, is the experience of discovering that soundness proofs are easy and completeness proofs are hard(er)!

1.6.5 What is a Completeness Theorem?

The completeness+soundness result we have just seen for the rules of natural deduction and the concept of a propositional tautology connects two sets. One set is defined by a semantical property (being satisfied by all valuations) and the other is defined by a syntactic property (being generated by a set of rules. Indeed the property of being generated by a set of rules is equivalent to being what is called in the literature a recursively enumerable ("r.e.") set or (more illuminatingly) a **semidecidable** set. We say a set X is semidecidable if there is a procedure $\mathcal{P}$ that will authenticate its members (so whenever a candidate for membership is in fact a member this will be confirmed in finite time). Notice that this does not require that the method $\mathcal{P}$ will reject any unsuiitable candididate in finite time. If there is a method that will reject any unsuiitable candididate in finite time then the complement of X is semidecidable and we say X is **decidable** ("recursive" is the old terminology).

So typically a completeness theorem is an assertion about two sets X and Y where X is a set defined semantically (as it might be, the set of tautologies) and Y is a semidecidable set defined by a syntactic criterion (as it might be the set of strings that have natural deduction proofs) and says that $X = Y$.

You may have felt tempted to say that the completeness theorem for propositional logic was no big deal. So we have this set of tautologies ... well, cook up some rules that generate them all. What's the problem? The problem is that there might be no such set of rules. We will see later that there are Logics which cannot be captured by a set of rules in this way: every set of rules either generates things it shouldn't or fails to generate some things it should. (Trakhtenbrot's theorem; second-order logic)

There is a completeness theorem for predicate logic, as we shall see. There is also a completeness theorem for constructive logic.

1.6.6 Interpolation

Exercise 15 *Suppose A is a propositional formula and 'p' is a letter appearing in A. Show that there are formulæ A_1 and A_2 not containing 'p' such that A is semantically equivalent to $(A_1 \wedge p) \vee (A_2 \wedge \neg p)$. [Hint: consider how you might be able to simplify A on coming to know the truth-value of 'p'.]*

Exercise 16 *Find an interpolant Q for*

$$(A \wedge B) \vee (\neg A \wedge C) \quad \vdash \quad (B \to C) \to (D \to C)$$

and supply proofs (in whatever style you prefer) of

$$(A \wedge B) \vee (\neg A \wedge C) \quad \to \quad Q$$

and

$$Q \quad \rightarrow \quad ((B \rightarrow C) \rightarrow (D \rightarrow C))$$

1.7 Sequent Calculus

Imagine you are given the task of finding a natural deduction proof of the tautology

$$(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r)).$$

Obviously the first thing you do is to attack the principal connective, and claim that $(p \rightarrow q) \rightarrow (p \rightarrow r)$ is obtained by an $\rightarrow$-introduction as follows:

$$\frac{\begin{array}{c} p \rightarrow (q \rightarrow r) \\ \vdots \end{array}}{(p \rightarrow q) \rightarrow (p \rightarrow r)} \rightarrow\text{-int} \tag{1.1}$$

in the hope that we can fill the dots in later. Notice that we don't know at this stage how many lines or how much space to leave .... At the second stage the obvious thing to do is try $\rightarrow$-introduction again, since '$\rightarrow$' is the principal connective of '$(p \rightarrow q) \rightarrow (p \rightarrow r)$'. This time my proof sketch has a conclusion which looks like

$$\frac{\dfrac{\begin{array}{c}\vdots\\ p \rightarrow r\end{array}}{\rightarrow\text{-int}}}{(p \rightarrow q) \rightarrow (p \rightarrow r)} \rightarrow\text{-int} \tag{1.2}$$

and we also know that floating up above this—somewhere—are the two premisses $p \rightarrow (q \rightarrow r)$ and $p \rightarrow q$. But we don't know where on the page to put them!

This motivates a new notation. Record the endeavour to prove

$$(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$$

by writing

$$\vdash (p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r)).$$

using the new symbol '$\vdash$'.[4] Then stage two (which was formula 1.1) can be described by the formula

$$p \rightarrow (q \rightarrow r) \vdash ((p \rightarrow q) \rightarrow (p \rightarrow r)).$$

which says that $(p \rightarrow q) \rightarrow (p \rightarrow r)$ can be deduced from $p \rightarrow (q \rightarrow r)$. Then the third stage [which I couldn't write down and which was formula 1.2, which said that $p \rightarrow r$ can be deduced from $p \rightarrow q$ and $p \rightarrow (q \rightarrow r)$] comes out as

$$p \rightarrow (q \rightarrow r),\ p \rightarrow q \ \vdash\ p \rightarrow r$$

This motivates the following gadgetry.

A **sequent** is an expression $\Gamma \vdash \psi$ where Γ is a set of formulæ and ψ is a formula. $\Gamma \vdash \psi$ says that there is a deduction of ψ from Γ. In sequent calculus one reasons not about formulæ—as one did with natural deduction—but instead about sequents, which are assertions about deductions between formulæ.

> **Programme: sequent calculus is natural deduction with control structures! A sequent proof is a program that computes a natural deduction proof.**

[4] For some reason this symbol is called 'turnstile'

Capital Greek letters denote sets of formulæ and lower-case Greek letters denote formulæ.

We accept any sequent that has a formula appearing on both sides. Such sequents are called **initial sequents**. Clearly the allegation made by an initial sequent is correct!

There are some obvious rules for reasoning about these sequents. Our endeavour to find a nice way of thinking about finding a natural deduction proof of

$$(p \to (q \to r)) \to ((p \to q) \to (p \to r))$$

gives us something that looks in part like

$$\dfrac{\dfrac{\dfrac{\underline{p \to (q \to r), (p \to q), p \vdash r}}{\underline{p \to (q \to r), (p \to q) \vdash (p \to r)}}}{p \to (q \to r) \vdash (p \to q) \to (p \to r)}}{\vdash (p \to (q \to r)) \to ((p \to q) \to (p \to r))}$$

and this means we are using a rule

$$\frac{\Gamma, A \ \vdash \ B}{\Gamma \ \vdash \ A \to B} \to \mathrm{R} \tag{1.3}$$

Of course there are lots of other rules, and here is a summary of them:

$\vee L$: $\dfrac{\Gamma, \psi \vdash \Delta \quad \Gamma', \phi \vdash \Delta'}{\Gamma \cup \Gamma', \underline{\psi \vee \phi} \vdash \Delta \cup \Delta'}$ $\qquad$ $\vee$ R: $\dfrac{\Gamma \vdash \Delta, \phi}{\Gamma \vdash \Delta, \underline{\psi \vee \phi}}$ $\qquad$ $\vee$ R: $\dfrac{\Gamma \vdash \Delta, \psi}{\Gamma \vdash \Delta, \underline{\psi \vee \phi}}$

$\wedge L$: $\dfrac{\Gamma, \psi, \phi \vdash \Delta}{\Gamma, \underline{\psi \wedge \phi} \vdash \Delta}$ $\qquad$ $\wedge R$: $\dfrac{\Gamma \vdash \Delta, \psi \quad \Gamma' \vdash \Delta', \phi}{\Gamma \cup \Gamma' \vdash \Delta \cup \Delta', \underline{\psi \wedge \phi}}$

$\neg L$: $\dfrac{\Gamma \vdash \Delta, \psi}{\Gamma, \underline{\neg\psi} \vdash \Delta}$ $\qquad$ $\neg R$: $\dfrac{\Gamma, \psi \vdash \Delta}{\Gamma \vdash \Delta, \underline{\neg\psi}}$

$\to L$: $\dfrac{\Gamma \vdash \Delta, \phi \quad \Gamma', \psi \vdash \Delta'}{\Gamma \cup \Gamma', \underline{\phi \to \psi} \vdash \Delta \cup \Delta'}$ $\qquad$ $\to R$: $\dfrac{\Gamma, \psi \vdash \Delta, \phi}{\Gamma \vdash \Delta, \underline{\psi \to \phi}}$

Weakening-L: $\dfrac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}$; Weakening-R: $\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, B}$;

Contraction-L: $\dfrac{\Gamma, \psi, \psi \vdash \Delta}{\Gamma, \underline{\psi} \vdash \Delta}$; $\quad$ Contraction-R: $\dfrac{\Gamma \vdash \Delta, \psi, \psi}{\Gamma \vdash \Delta, \underline{\psi}}$;

Cut: $\dfrac{\Gamma \vdash \Delta, \underline{\psi} \quad \Gamma', \underline{\psi} \vdash \Delta'}{\Gamma \cup \Gamma' \vdash \Delta, \Delta'}$.

In this box I have followed the universal custom of writing 'Γ, ψ' for '$\Gamma \cup \{\psi\}$'; I have not so far followed the similarly universal custom of writing 'Γ, Δ' instead of '$\Gamma \cup \Delta$' but from now on I will.

You might find useful the terminology of **eigenformula**. The eigenformula of an application of a rule is the formula being attacked by that application. In each rule in the box above I have underlined the eigenformula.

There is no rule for the biconditional: we think of a biconditional $A \longleftrightarrow B$ as a conjunction of two conditionals $A \to B$ and $B \to A$.

The two rules of $\vee$-R give rise to a derived rule which makes good sense when we are allowed more than one formula on the right, it is

$$\frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B}$$

$$\frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B}$$

I shall explain soon (section **??**) why this is legitimate.

A word is in order on the two rules of contraction. Whether one needs the contraction rules or not depends on whether one thinks of the left and right halves of sequents as sets or as multisets. Both courses of action can be argued for. If one thinks of them as multisets then one can keep track of the multiple times one exploits an assumption. If one thinks of them as as sets then one doesn't need the contraction rules. It's an interesting exercise in philosophy of mathematics to compare the benefits of the two ways of doing it, and to consider the sense in which they are equivalent. Since we are not hell-bent on rigour we will equivocate between the two approaches: in all the proofs we consider it will be fairly clear how to move from one approach to the other and back.

A bit of terminology you might find helpful. Since premisses and conclusion are the left and right parts of a sequent, what are we going to call the things above and below the line in a sequent rule? The terminology **precedent** and **succedent** is sometimes used. I'm not going to expect you to know it: I'm offering it to you here now because it might help to remind you that it's a different distinction from the premiss/conclusion distinction. I think it is more usual to talk about the **upper sequent** and the **lower sequent**.

You will notice that I have cheated: some of these rules allow there to be more than one formula on the right! There are various good reasons for this, but they are quite subtle and we may not get round to them. If we are to allow more than one formula on the right, then we have to think of $\Gamma \vdash \Delta$ as saying that every valuation that makes everything Γ true also makes something in Δ true. We can't correctly think of $\Gamma \vdash \Delta$ as saying that there is a proof of something in Δ using premisses in Γ because:

$$A \vdash A$$

is an initial sequent. so we can use $\neg-\mathrm{R}$ to infer

$$\vdash A, \neg A.$$

So $\vdash A, \neg A$ is an OK sequent. Now it just isn't true that there is always a proof of A or a proof of $\neg A$, so this example shows that it similarly just isn't true that a sequent can be taken to assert that there is a proof of something on the right using only premisses found on the left—unless we restrict matters so that there is only one formula on the right. This fact illustrates how allowing two formulæ on the right can be useful: the next step is to infer the sequent

$$\vdash A \vee \neg A$$

and we can't do that unless we allow two formulæ on the right.

However, it does help inculcate the good habit of thinking of sequents as metaformulæ, as things that formalise facts about formulæ rather than facts of the kind formalised by the formulæ.

One thing you will need to bear in mind, but which we have no space to prove in this course, is that sequent proofs with more than formula on the right correspond to natural deduction proofs using the rule of double negation. N.B.: Display this properly commas on the left of a sequent mean 'and' while commas on the right-hand

side mean 'or'! This might sound odd, but it starts to look natural quite early, and you will get used to it easily.

A summary of what we have done so far with Natural Deduction and Sequent Calculus.

- A sequent calculus proof is a log of attempts to build a natural deduction proof.
- So a sequent is telling you that there is a proof of the formula on the right using as premisses the formulæ on the left.
- But we muck things up by allowing more than one formula on the right so we have to think of a sequent as saying if everything on the left is true then something on the right is true.
- Commas on the left are **and**, commas on the right are **or**.

1.8 Hilbert-style Proofs

In this style of proof we have only three axioms

K: $A \to (B \to A)$

S: $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$

T: $(\neg A \to B) \to ((\neg A \to \neg B) \to A)$

and the rules of *modus ponens* and substitution. 'K' and 'S' are standard names for the first two axioms. There is a good reason for this, which we will see in chapter **??**. The third axiom does not have a similarly standard name.

Notice that only two connectives appear here: $\to$ and $\neg$. How are we supposed to prove things about $\wedge$ and $\vee$ and so on? The answer is that we define the other connectives in terms of $\to$ and $\neg$, somewhat as we did on page **??**—except that there we defined our connectives in terms of a different set of primitives.

Here is an example of a proof in this system:

1. $A \to ((A \to A) \to A)$	Instance of K
2. $(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$	Instance of S
3. $(A \to (A \to A)) \to (A \to A)$	Modus Ponens (1) and (2)
4. $A \to (A \to A)$	Instance of K:
5. $A \to A$	Modus Ponens (3) and (4)

I thought I would give you an illustration of a proof before giving you a definition. Here is the definition.

Definition 5 *A Hilbert-style proof is a list of formulæ wherein every formula is either an axiom or is obtained from earlier formulæ in the list by modus ponens or substitution.*

Some comments at this point.

1. We can do without the rule of substitution, simply by propagating the substitutions we need back to the axioms in the proof and ruling that a substitution instance of an axiom is an axiom.

2. We can generalise this notion to allow assumptions as well as axioms. That way we have—as well as the concept of an outright (Hilbert)-proof—the concept of a *Hilbert-proof of a formula from a list of assumptions.*

3. An initial segment of a Hilbert-style proof is another Hilbert-style proof—of the last formula in the list.

4. Hilbert-style proofs suffer from not having the subformula property, as the boxed proof (above, page 26) shows.

Exercise 17 *You have probably already found natural deduction proofs for K and S. If you have not done so, do it now. Find also a natural deduction proof of T, the third axiom. (You will need the rule of double negation).*

Exercise 18 *Go back to Zarg (exercise 6 p. 12) and—using the truth-table for $\neg$ that you decided that the Zarglings use—check that the Zarglings do not believe axiom T to be a tautology.*

I will spare you the chore of testing whether or not the Zarglings believe S to be a tautology. One reason is that it would involve writing out a truth-table with a dispiritingly large number of rows. How many rows exactly?

Exercise 19 *[For enthusiasts only]*
Find Hilbert-style proofs of the following tautologies

(a) $B \to \neg\neg B$.
(b) $\neg A \to (A \to B)$.
(c) $A \to (\neg B \to \neg(A \to B))$.
(d) $(A \to B) \to ((\neg A \to B) \to B)$.

Notice how easy it is to prove that the Hilbert-style proof system is sound! After all, every substitution-instance of a tautology is a tautology, and if $A \to B$ and A are tautologies, so is B.

This needs massive expansion

1.8.1 The Deduction Theorem

In this Hilbert-style proof system the only rules of inference are *modus ponens* and substitution. Establishing that $A \to A$ is a theorem—as we did above—is quite hard work in this system. If we had a derived rule that said that if we have a Hilbert-style proof of A using a premiss B then we have a Hilbert-style proof of $A \to B$ then as a special case we would know that there was a Hilbert-proof of $A \to A$.

To justify a derived rule that says that if we have a Hilbert-proof of A from B then there is a Hilbert-proof of $A \to B$ we will have to show how to transform a proof of B with an assumption A in it into a proof of $A \to B$. Let the Hilbert-proof of B be the list whose ith member is B_i. The first thing we do is replace every B_i by $A \to B_i$ to obtain a new list of formulæ. This list isn't a proof, but it is the beginnings of one.

Suppose B_k had been obtained from B_i and B_j by *modus ponens* with B_i as major premiss, so B_i was $B_j \to B_k$. This process of whacking '$A \to$' on the front of every formula in the list turns these into $A \to (B_j \to B_k)$ and $A \to B_j$. Now altho' we could obtain B_k from B_j and $B_j \to B_k$ by *modus ponens* we clearly can't obtain $A \to B_k$ from $A \to B_j$ and $A \to (B_j \to B_k)$ quite so straightforwardly. However we can construct a little Hilbert-style proof of $A \to B_k$ from $A \to B_j$ and $A \to (B_j \to B_k)$ using S. When revising you might like to try covering up the next few formulæ and working it out yourself.

1. $(A \to (B_j \to B_k)) \to ((A \to B_j) \to (A \to B_k))$	S
2. $A \to (B_j \to B_k)$	
3. $(A \to B_j) \to (A \to B_k)$	*modus ponens* (1), (2)
4. $A \to B_j$	
5. $A \to B_k$	*modus ponens* (3), (4)

Lines (2) and (4) I haven't labelled. Where did they come from? Well, what we have just seen is an explanation of how to get $A \to B_k$ from $A \to (B_j \to B_k)$ and $A \to B_j$ given that we can get B_k from B_j and $B_j \to B_k$. What the box shows us is how to rewrite any **one** application of *modus ponens*. What we have to do to prove the deduction theorem is to do this trick to every occurrence of *modus ponens*.

Revise this: it isn't correct

If we apply this process to:
$A \to ((A \to B) \to B)$
$A, A \to B \vdash B$
$A \vdash ((A \to B) \to B)$
we obtain

1. $(A \to B) \to (((A \to B) \to (A \to B)) \to (A \to B))$ Instance of K
2. $((A \to B) \to (((A \to B) \to (A \to B)) \to (A \to B))) \to (((A \to B) \to ((A \to B) \to (A \to B))) \to ((A \to B) \to (A \to B)))$ Instance of S
3. $((A \to B) \to ((A \to B) \to (A \to B))) \to ((A \to B) \to (A \to B))$ Modus Ponens (1) and (2)
4. $(A \to B) \to ((A \to B) \to (A \to B))$ Instance of K:
5. $(A \to B) \to (A \to B)$ Modus Ponens (3) and (4)
6. $((A \to B) \to (A \to B)) \to (((A \to B) \to A) \to ((A \to B) \to B)$ Instance of S
7. $((A \to B) \to A) \to ((A \to B) \to B)$ Modus ponens (6), (5)
8. A Assumption
9. $A \to ((A \to B) \to A)$ Instance of K.
10. $(A \to B) \to A$ Modus ponens (9), (8).
11. $(A \to B) \to B$ modus ponens (10), (7).

(Of course the annotations at the beginning and end of the lines are not part of the proof but are part of a commentary on it. That's the language-metalanguage distinction again.)

Say more about this

The deduction theorem is a pleasing phenomenon: it is an echo in the object language of something going on in the metlanguage

More to do here

Theorem 6 *If* $\Gamma, A \vdash B$ *then* $\Gamma \vdash A \to B$

Chapter 2

Curry-Howard

The Curry-Howard trick is to exploit the possibility of using the letters 'A', 'B' *etc.* to be dummies not just for propositions but for sets. This means reading the symbols '$\rightarrow$', '$\wedge$', '$\vee$' etc. as symbols for operations on sets as well as on formulæ. The ambiguity we will see in the use of '$A \rightarrow B$' is quite different from the ambiguity arising from the two uses of the word 'tank'. Those two uses are completely unrelated. In contrast the two uses of the arrow in '$A \rightarrow B$' have a deep and meaningful relationship. The result is a kind of cosmic pun. Here is the simplest case.

Altho' we use it as a formula in propositional logic, the expression '$A \rightarrow B$' is used by various mathematical communities to denote the set of all functions from A to B. To understand this usage you don't really need to have decided whether your functions are to be functions-in-intension or functions-in-extension; either will do. The ideas in play here work quite well at an informal level. A function from A to B is a thing such that when you give it a member of A it gives you back a member of B.

2.1 Decorating Formulæ

2.1.1 The rule of $\rightarrow$-elimination

Consider the rule of $\rightarrow$-elimination

$$\frac{A \qquad A \rightarrow B}{B} \rightarrow\text{-elim} \tag{2.1}$$

If we are to think of A and B as sets then this will say something like "If I have an A (abbreviation of "if i have a member of the set A") and an $A \rightarrow B$ then I have a B". So what might an $A \rightarrow B$ (a member of $A \rightarrow B$) be? Clearly $A \rightarrow B$ must be the set of functions that give you a member of B when fed a member of A. Thus we can decorate 2.1 to obtain

$$\frac{a : A \qquad f : A \rightarrow B}{f(a) : B} \rightarrow\text{-elim} \tag{2.2}$$

which says something like: "If a is in A and f takes As to Bs then $f(a)$ is a B.[1] This gives us an alternative reading of the arrow: '$A \rightarrow B$' can now be read ambiguously as either the conditional "if A then B" (where A and B are propositions) or as a notation for the set of all functions that take members of A and give members of B as output (where A and B are sets).

[1] So why not write this as '$a \in A$' if it means that a is a member of A? There are various reasons, some of them cultural, but certainly one is that here one tends to think of the denotations of the capital letters 'A' and 'B' and so on as predicates rather than sets.

These new letters preceding the colon sign are **decorations**. The idea of Curry-Howard is that we can decorate *entire proofs*—not just individual formulæ—in a uniform and informative manner.

We will deal with $\rightarrow$-int later. For the moment we will look at the rules for $\wedge$.

2.1.2 Rules for $\wedge$

2.1.2.1 The rule of $\wedge$-introduction

Consider the rule of $\wedge$-introduction:

$$\frac{A \qquad B}{A \wedge B}\ \wedge\text{-int} \tag{2.1}$$

If I have an A and a B then I have a . . . ? thing that is both A and B? No. If I have one apple and I have one banana then I don't have a thing that is both an apple and a banana; what I do have is a sort of plural object that I suppose is a pair of an apple and a banana. (By the way I hope you are relaxed about having compound objects like this in your world. Better start your breathing exercises *now*.) The thing we want is called an **ordered pair**: $\langle a, b\rangle$ is the ordered pair of a and b. So the decorated version of 2.1 is

$$\frac{a : A \qquad b : B}{\langle a, b\rangle : A \times B}\ \wedge\text{-int} \tag{2.2}$$

Say something about how we use $\times$ here . . .

What is the ordered pair of a and b? It might be a kind of funny plural object, like the object consisting of all the people in this room, but it's safest to be entirely *operationalist*[2] about it: all you know about ordered pairs is that there is a way of putting them together and a way of undoing the putting-together, so you can recover the components. Asking for any further information about what they *are* is not cool: they are what they do. Be doo be doo. That's operationalism for you.

2.1.2.2 The rule of $\wedge$-elimination

If you can do them up, you can undo them: if I have a pair-of-an-A-and-a-B then I have an A and I have a B.

$$\frac{\langle a, b\rangle : A \wedge B}{a : A} \qquad\qquad \frac{\langle a, b\rangle : A \wedge B}{b : B}$$

$A \times B$ is the set $\{\langle a, b\rangle : a \in A \wedge b \in B\}$ of[3] pairs whose first components are in A and whose second components are in B. $A \times B$ is the **Cartesian product** of A and B.

(Do not forget that it's $A \times B$ not $A \cap B$ that we want. A thing in $A \cap B$ is a thing that is both an A and a B: it's not a pair of things one of which is an A and the other a B; remember the apples and bananas above.)

2.1.3 Rules for $\vee$

To make sense of the rules for $\vee$ we need a different gadget.

$$\frac{A}{A \vee B} \qquad\qquad \frac{B}{A \vee B}$$

If I have a thing that is an A, then I certainly have a thing that is either an A or a B—namely the thing I started with. And in fact I know which of A and

[2] Have a look at chapter 1

[3] If you are less than 100% happy about this curly bracket notation have a look at the discrete mathematics material on my home page

B it is—it's an A. Similarly If I have a thing that is a B, then I certainly have a thing that is either an A or a B—namely the thing I started with. And in fact I know which of A and B it is—it's a B.

Just as we have cartesian product to correspond with $\wedge$, we have **disjoint union** to correspond with $\vee$. This is not like the ordinary union you may remember from school maths. You can't tell by looking at a member of $A \cup B$ whether it got in there by being a member of A or by being a member of B. After all, if $A \cup B$ is $\{1, 2, 3\}$ it could have been that A was $\{1, 2\}$ and B was $\{2, 3\}$, or the other way round. Or it might have been that A was $\{2\}$ and B was $\{1, 3\}$. Or they could both have been $\{1, 2, 3\}$! We can't tell. However, with disjoint union you *can* tell.

The disjoint union $A \sqcup B$ of A and B is obtained by making copies of everything in A and marking them with wee flecks of *pink* paint and making copies of everything in B and marking them with wee flecks of *blue* paint, then putting them all in a set. We can put this slightly more formally, now that we have the concept of an ordered pair: $A \sqcup B$ is

$$(A \times \{\texttt{pink}\}) \quad \cup \quad (B \times \{\texttt{blue}\}),$$

where `pink` and `blue` are two arbitrary labels.

(Check that you are happy with the notation: $A \times \{\texttt{pink}\}$ is the set of all ordered pairs whose first component is in A and whose second component is in $\{\texttt{pink}\}$ which is the singleton of[4] `pink`, which is to say whose second component *is* `pink`. Do not ever confuse any object x with the set $\{x\}$—the set whose sole member is x! So an element of $A \times \{\texttt{pink}\}$ is an ordered pair whose first component is in A and whose second component is `pink`. We can think of such an ordered pair as an object from A labelled with a pink fleck.)

Say something about $A \sqcup B = B \sqcup A$

$\vee$-introduction now says:

$$\frac{a : A}{\langle a, \texttt{pink}\rangle : A \sqcup B} \qquad \frac{b : B}{\langle b, \texttt{blue}\rangle : A \sqcup B}$$

$\vee$-elimination is an action-at-a-distance rule (like $\rightarrow$-introduction) and to treat it properly we need to think about:

2.2 Propagating Decorations

The first rule of decorating is to decorate each assumption with a variable, a thing with no syntactic structure: a single symbol.[5] This is an easy thing to remember, and it helps guide the beginner in understanding the rest of the gadgetry. Pin it to the wall:

Decorate each assumption with a variable!

How are you to decorate formulæ that are not assumptions? You can work that out by checking what rules they are the outputs of. We will discover through some examples what extra gadgetry we need to sensibly extend decorations beyond assumptions to the rest of a proof.

[4] The singleton of x is the set whose sole member is x.

[5] You may be wondering what you should do if you want to introduce the same assumption twice. Do you use the same variable? The answer is that if you want to discharge two assumptions with a single application of a rule then the two assumptions must be decorated with the same variable.

2.2.1 Rules for $\wedge$

2.2.1.1 The rule of $\wedge$-elimination

$$\frac{A \wedge B}{B}\ \wedge\text{-elim} \tag{2.1}$$

We decorate the premiss with a variable:

$$\frac{x : A \wedge B}{B}\ \wedge\text{-elim} \tag{2.2}$$

... but how do we decorate the conclusion? Well, x must be an ordered pair of something in A with something in B. What we want is the second component of x, which will be a thing in B as desired. So we need a gadget that when we give it an ordered pair, gives us its second component. Let's write this '`snd`'.

$$\frac{x : A \wedge B}{\texttt{snd}(x) : B}$$

By the same token we will need a gadget '`fst`' which gives the first component of an ordered pair so we can decorate[6]

$$\frac{A \wedge B}{A}\ \wedge\text{-elim} \tag{2.3}$$

to obtain

$$\frac{x : A \wedge B}{\texttt{fst}(x) : A}$$

2.2.1.2 The rule of $\wedge$-introduction

Actually we can put these proofs together and whack an $\wedge$-introduction on the end:

$$\frac{\dfrac{x : A \wedge B}{\texttt{snd}(x) : B} \quad \dfrac{x : A \wedge B}{\texttt{fst}(x) : A}}{\langle \texttt{snd}(x), \texttt{fst}(x) \rangle : B \wedge A}$$

2.2.2 Rules for $\rightarrow$

7.2.2.1 The rule of $\rightarrow$-introduction

Here is a simple proof using $\rightarrow$-introduction.

$$\frac{\dfrac{[A \rightarrow B]^1 \qquad A}{B}\ \rightarrow\text{-elim}}{(A \rightarrow B) \rightarrow B}\ \rightarrow\text{-int } (1) \tag{2.1}$$

We decorate the two premisses with single letters (variables): say we use 'f' to decorate '$A \rightarrow B$', and 'x' to decorate 'A'. (This is sensible. 'f' is a letter traditionally used to point to functions, and clearly anything in $A \rightarrow B$ is going to be a function.) How are we going to decorate 'B'? Well, if x is in A and f is a function that takes things in A and gives things in B then the obvious thing in B that we get is going to be denoted by the decoration '$f(x)$':

$$\frac{\dfrac{f : [A \rightarrow B]^1 \quad x : A}{f(x) : B}}{??? : (A \rightarrow B) \rightarrow B}$$

[6] Agreed: it's shorter to write 'x_1' and 'x_2' than it is to write 'fst(x)' and 'snd(x)' but this would prevent us using 'x_1 and x_2' as variables and in any case I prefer to make explicit the fact that there is a function that extracts components from ordered pairs, rather than having it hidden it away in the notation.

So far so good. But how are we to decorate '$(A \to B) \to B$'? What can the '???' stand for? It must be a notation for a thing (a function) in $(A \to B) \to B$; that is to say, a notation for something that takes a thing in $A \to B$ and returns a thing in B. What might this function be? It is given f and gives back $f(x)$. So we need a notation for a function that, on being given f, returns $f(x)$. (Remember, we decorate all assumptions with variables, and we reach for this notation when we are discharging an assumption so it will always be a variable). We write this

$$\lambda f.f(x)$$

This notation points to the function which, when given f, returns $f(x)$. In general we need a notation for a function that, on being given x, gives back some possibly complex term t. We will write:

$$\lambda x.t$$

for this. Thus we have

$$\dfrac{\dfrac{f : [A \to B]^1 \qquad x : A}{f(x) : B} \to\text{-elim}}{\lambda f.f(x) : (A \to B) \to B} \to\text{-int } (1) \tag{2.2}$$

Thus, in general, an application of $\to$-introduction will gobble up the proof

$$\dfrac{\begin{array}{c}\underline{x : A} \\ \vdots \\ t : B\end{array}}{}$$

and emit the proof

$$\dfrac{\begin{array}{c}\underline{[x : A]} \\ \vdots \\ t : B\end{array}}{\lambda x.t : A \to B}$$

This notation—$\lambda x.t$—for a function that accepts x and returns t is incredibly simple and useful. Almost the only other thing you need to know about it is that if we apply the function $\lambda x.t$ to an input y the output must be the result of substituting 'y' for all the occurrences of 'x' in t. In the literature this result is notated in several ways, for example $[y/x]t$ or $t[y/x]$.

Go over a proof of S at this point

2.2.3 Rules for $\vee$

We've discussed $\vee$-introduction but not $\vee$-elimination. It's very tricky and—at this stage at least—we don't really need to. It's something to come back to—perhaps! [7]

[7] For any gluttons for punishment out there here is a message from my former student Nick Benton of Microsoft Research:

"$\vee$-elim goes to a generalization of `if-then-else` called "`case`":

$$\dfrac{\vdash E : A + B \qquad x : A \vdash M : C \qquad y : B \vdash N : C}{\vdash \texttt{case}\ E\ \texttt{of}\ \mathrm{inl}(x) \Rightarrow M | \mathrm{inr}(y) \Rightarrow N : C}$$

Note 'x' bound in M, 'y' in N. The operational behaviour is to evaluate E, see if it turns into $\mathrm{inl}(a)$ for some $a \in A$ and—if it does—evaluate M with 'x' bound to a, otherwise the symmetric thing. `if-then-else` is morally the special case where A and B are both just 1, the one element type, though binding a variable to a value of type 1 is a bit of a waste of time, so we simplify the syntax.

Haskell, ML etc have `case in`, and that's what it's called there too, but they generalize the forms of pattern matching somewhat."

Exercise 20 *Go back and look at the proofs that you wrote up in answer to exercise 2, and decorate those that do not use '$\vee$'.*

2.2.4 Remaining Rules

2.2.4.1 Identity Rule

See [**?**]: Semantical Archæology.

Here is a very simple application of the identity rule.

$$\dfrac{\dfrac{\dfrac{A \quad B}{B}}{B \to A}}{A \to (B \to A)}$$

Can you think of a function from A to the set of all functions from B to A? If I give you a member a of A, what function from B to A does it suggest to you? Obviously the function that, when given b in B, gives you a.

This gives us the decoration

$$\dfrac{\dfrac{\dfrac{a : A \quad b : B}{b : B}}{\lambda b.a : B \to A}}{\lambda a.(\lambda b.a) : A \to (B \to A)}$$

Show how do do this using the option of cancelling non-existent assumptions.

The function $\lambda a.\lambda b.a$ has a name: K for Konstant. (See section 1.8.)

2.2.4.2 The *ex falso*

The *ex falso sequitur quodlibet* speaks of the propositional constant $\bot$. To correspond to this constant *proposition* we are going to need a constant *set*. The obvious candidate for a set corresponding to $\bot$ is the empty set. Now $\bot \to A$ is a propositional tautology. Can we find a function from the empty set to A which we can specify without knowing anything about A? Yes: the empty function! (You might want to check very carefully that the empty function ticks all the right boxes: is it really the case that whenever we give the empty function a member of the empty set to contemplate it gives us back one and only one answer? Well yes! It has never been known to fail to do this!! Look again at page **??**.) That takes care of $\bot \to A$, the *ex falso*.

2.2.4.3 Double Negation

What are we to make of $A \to \bot$? Clearly there can be no function from A to the empty set unless A is empty itself. What happens to double negation under this analysis?

$$((A \to \bot) \to \bot) \to A$$

- If A is empty then $A \to \bot$ is the singleton of the empty function and is not empty. So $(A \to \bot) \to \bot$ is the set of functions from a nonempty set to the empty set and is therefore the empty set, so $((A \to \bot) \to \bot) \to A$ is the set of functions from the empty set to the empty set and is therefore the singleton of the empty function, so it is at any rate nonempty.

- However if A is nonempty then $A \to \bot$ is empty. So $(A \to \bot) \to \bot$ is the set of functions from the empty set to the empty set and is nonempty—being the singleton of the empty function—so $((A \to \bot) \to \bot) \to A$ is the set of functions from a nonempty set to the empty set and is therefore empty.

So $((A \to \bot) \to \bot) \to A$ is not reliably inhabited. This is in contrast to all the other truth-table tautologies we have considered. Every other truth-table tautology that we have looked at has a lambda term corresponding to it.

continued

A final word of warning: notice that we have not provided any λ-gadgetry for the quantifiers. This can in fact be done, but there is no spacetime here to do it properly.

2.3 Exercises

In the following exercises you will be invited to find λ terms to correspond to particular wffs—in the way that the lambda term $\lambda a.\lambda b.a$ (aka 'K') corresponds to $A \to (B \to A)$ (also aka 'K'!) You will discover very rapidly that the way to find a λ-term for a formula is to find a proof of that formula: λ-terms encode proofs!

Exercise 21 *Find λ-terms for*

1. $(A \wedge B) \to A$;
2. $((A \to B) \wedge (C \to D)) \to ((A \wedge C) \to (B \wedge D))$;
3. $(A \to B) \to ((B \to C) \to (A \to C))$;
4. $((A \to B) \to A) \to ((A \to B) \to B)$;
5. $(A \to (B \to C)) \to (B \to (A \to C))$;
6. $(A \to (B \to C)) \to (B \wedge A) \to C))$;
7. $((B \wedge A) \to C)) \to (A \to (B \to C))$;

Finding λ-terms in exercise 21 involves of course first finding natural deduction proofs of the formulæ concerned. A provable formula will always have more than one proof. (It won't always have more than one *sensible* proof!) For example the tautology $(A \to A) \to (A \to A)$ has these proofs (among others)

$$\dfrac{\dfrac{\dfrac{[A \to A]^1}{A \to A}\,\text{identity rule}}{(A \to A) \to (A \to A)}\,{\to}\text{-int }(1)} \tag{2.1}$$

$$\dfrac{\dfrac{\dfrac{[A]^1 \qquad [A \to A]^2}{A}\,{\to}\text{-elim}}{A \to A}\,{\to}\text{-int }(1)}{(A \to A) \to (A \to A)}\,{\to}\text{-int }(2) \tag{2.2}$$

$$\dfrac{\dfrac{\dfrac{[A]^1 \qquad [A \to A]^2}{A}\,{\to}\text{-elim} \qquad [A \to A]^2}{A}\,{\to}\text{-elim}}{A \to A}\,{\to}\text{-int }(1)}{(A \to A) \to (A \to A)}\,{\to}\text{-int }(2) \tag{2.3}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A]^1 \qquad [A \to A]^2}{A}\to\text{-elim} \qquad [A \to A]^2}{A}\to\text{-elim} \qquad [A \to A]^2}{A}\to\text{-elim}}{\dfrac{A}{A \to A}\to\text{-int }(1)}}{(A \to A) \to (A \to A)}\to\text{-int }(2) \tag{2.4}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A]^1 \qquad [A \to A]^2}{A}\to\text{-elim} \qquad [A \to A]^2}{A}\to\text{-elim} \qquad [A \to A]^2}{A}\to\text{-elim} \qquad [A \to A]^2}{A}\to\text{-elim}}{\dfrac{\dfrac{A}{A \to A}\to\text{-int }(1)}{(A \to A) \to (A \to A)}\to\text{-int }(2)} \tag{2.5}$$

Exercise 22 *Decorate all these proofs with λ-terms. If you feel lost, you might like to look at the footnote*[8] *for a HINT.*

2.3.1 Need a section here on Combinators and Hilbert proofs

Exercise 23 *Provide,* **without** *using the rule of double negation,*

1. *a natural deduction proof of* $\neg\neg((\neg A \to B) \to ((\neg A \to \neg B) \to A))$*;*
2. *a natural deduction proof of* $\neg\neg B$ *from* $\neg\neg A$ *and* $\neg\neg(\neg\neg A \to \neg\neg B)$*;*

Provide sequent proofs of the following, respecting the one-formula-on-the-right constraint.

1. $\vdash \neg\neg((\neg A \to B) \to ((\neg A \to \neg B) \to A))$*;*
2. $\neg\neg A, \neg\neg(\neg\neg A \to \neg\neg B) \vdash \neg\neg B$*.*

2.3.2 Some advanced exercises for enthusiasts

Life on Planet Zarg taught us that Peirce's law does not follow from K and S alone: we seem to need the rule of double negation. In fact Peirce's law, in conjunction with K and S, implies all the formulæ built up only from $\to$ that we can prove using the rule of double negation.

Exercise 24 *Observe that* $(P \to Q) \to Q$ *has the same truth-table as* $P \vee Q$*. Construct a natural deduction proof of* R *from the premisses* $(P \to Q) \to Q$*,* $P \to R$ *and* $Q \to R$*. You may additionally use as many instances of Peirce's law as you wish.*[9]

[8] Notice that in each proof of these proofs all the occurrences of '$A \to A$' are cancelled simultaneously.. Look at the footnote on page 31.

[9] I am endebted to Tim Smiley for this amusing fact.

Chapter 3

Lectures three and four

Predicate calculus syntax. Natural deduction and sequent calculus. Epsilon terms, completeness. Cut-elimination?

All the apparatus for constructing formulæ in propositional logic works too in this new context: If A and B are formulæ so are $A \vee B$, $A \wedge B$, $\neg A$ and so on. However we now have new ways of creating formulæ, new gadgets which we had better spell out:

There is really an abuse of notation here: we should use quasi-quotes ...

Constants and variables

Constants tend to be lower-case letters at the start of the latin alphabet ('a', 'b' ...) and variables tend to be lower-case letters at the end of the alphabet ('x', 'y', 'z' ...). Since we tend to run out of letters we often enrich them with subscripts to obtain a larger supply: 'x_1' etc.

Predicate letters

are upper-case letters from the latin alphabet, usually from the early part: 'F' 'G' .... They are called *predicate* letters because they arise from a programme of formalising reasoning about predicates and predication. '$F(x, y)$' could have arisen from 'x is fighting y'. Each predicate letter has a particular number of terms that it expects; this is the **arity** of the letter. If we feed it the correct number of terms—so we have an expression like $F(x, y)$—we call the result an **atomic formula.**

The equality symbol '=' is a very special predicate letter: you are not allowed to reinterpret it the way you can reinterpret other predicate letters. We in the Information Technology fraternity say of strings that cannot be assigned meanings by the user that they are **reserved**. It is said to be **part of the logical vocabulary**. The equality symbol '=' is the only relation symbol that is reserved. In this respect it behaves like '$\wedge$' and '$\forall$' and the connectives, all of which are reserved in this sense.

Unary predicates have one argument, **binary** predicates have two; n**-ary** have n. Similarly functions.

Atomic formulæ can be treated the way we treated literals in propositional logic: we can combine them together by using '$\wedge$' '$\vee$'and the other connectives.

Finally we can bind variables with **quantifiers**. Ther are two: $\exists$ and $\forall$. We can write things like

$$(\forall x)F(x)$$

everything is a frog;

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

everybody loves everyone

we might write this second thing as

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

to save space

The syntax for quantifiers is variable-preceded-by quantifier enclosed in brackets, followed by stuff inside brackets:

$(\exists x)(\ldots)$ and $(\forall y)()$. We sometimes omit the pair of brackets to the right of the quantifier when no ambiguity is caused thereby.

The difference between variables and constants is that you can bind variables with quantifiers, but you can't bind constants. The meaning of a constant is fixed.

complete this explanation; quantifiers are connectives too

. . . free

For example, in a formula like

$$(\forall x)(F(x) \to G(x))$$

the letter 'x' is a variable: you can tell because it is bound by the universal quantifier. The letter 'F' is not a variable, but a predicate letter. It is not bound by a quantifier, and cannot be: the syntax forbids it. In a first-order language you are not allowed to treat predicate letters as variables: you may not bind them with quantifiers. Binding predicate letters with quantifiers (treating them as variables) is the tell-tale sign of **second-order** Logic.

We also have

Function letters

These are lower-case latin letters, typically 'f' 'g' 'h'. We apply them to variables and constants, and this gives us **terms**: $f(x)$, $g(a,y)$ and suchlike. In fact we can even apply them to terms: $f(g(a,y))$, $g(f(g(a,y),x))$ and so on. So a term is either a variable or a constant or something built up from variables-and-constants by means of function letters. What is a function? That is, what sort of thing do we try to capture with function letters? We have seen an example: *father-of* is a function: you have precisely one father; *son-of* is not a function. Some people have more than one, or even none at all.

3.1 Exercises

Exercise 25 *In each formula circle the principal connective. (This requires more care than you might think! Pay close attention to the brackets)*

In each of the following pairs of formulæ, determine whether the two formulæ in the pair are (i) logically equivalent or are (ii) negations of each other or (iii) neither. The last two are quite hard.

$(\exists x)(F(x))$;	$\neg\forall x\neg F(x)$
$(\forall x)(\forall y)F(x,y)$;	$(\forall y)(\forall x)F(x,y)$
$(\exists x)(F(x) \vee G(x))$;	$\neg(\forall x)(\neg F(x) \vee \neg G(x))$
$(\forall x)(\exists y)(F(x,y))$;	$(\exists y)(\forall x)(F(x,y))$
$(\exists x)(F(x)) \to A$;	$(\forall x)(F(x) \to A)$
$(\exists x)(F(x) \to A)$;	$(\forall x)(F(x)) \to A$

(In the last two formulæ 'x' is not free in A)

Exercise 26 *Find proofs of the following sequents:*

1. $\neg\forall x\phi(x) \vdash \exists x\neg\phi(x)$;

2. $\neg\exists x\phi(x) \vdash \forall x\neg\phi(x)$;
3. $\phi \wedge \exists x\psi(x) \vdash \exists x(\phi \wedge \psi(x))$;
4. $\phi \vee \forall x\psi(x) \vdash \forall x(\phi \vee \psi(x))$,
5. $\phi \to \exists x\psi(x) \vdash \exists x(\phi \to \psi(x))$,
6. $\phi \to \forall x\psi(x) \vdash \forall x(\phi \to \psi(x))$,
7. $\exists x\phi(x) \to \psi \vdash \forall x(\phi(x) \to \psi)$
8. $\forall x\phi(x) \to \psi \vdash \exists x(\phi(x) \to \psi)$,
9. $\exists x\phi(x) \vee \exists x\psi(x) \vdash \exists x(\phi(x) \vee \psi(x))$,
10. $\forall x\phi(x) \wedge \forall x\psi(x) \vdash \forall x(\phi(x) \wedge \psi(x))$,

In this exercise ϕ and ψ are formulæ in which 'x' is not free, while $\phi(x)$ and $\psi(x)$ are formulæ in which 'x' may be free.

Exercise 27 *Prove the following sequents. The first one is really quite easy. (It is Russell's paradox of the set of all sets that are not members of themselves. The second one underlines the fact that you do not need a biconditional in the definition of 'symmetric'.*

1. $\vdash \neg(\exists x)(\forall y)(P(y,x) \longleftrightarrow \neg(P(y,y)))$
2. $\forall x\forall y(R(x,y) \to R(y,x)) \vdash \forall x\forall y(R(x,y) \longleftrightarrow R(y,x))$;
3. $\vdash \neg(\exists x)(\forall y)(P(y,x) \longleftrightarrow (\forall z)(P(z,y) \to \neg P(y,z)))$

   This formula concerns the modified paradox of Russell concerning the set of those sets that are not members of any member of themselves.

   It is noticeably harder, and is recommended mainly for enthusiasts. You will certainly need to "keep a copy"! You will find it much easier to find a proof that uses cut. Altho' there is certainly a proof that never has more than one formula on the right you might wish to start off without attempting to respect this constraint.

Exercise 28 *Find a proof of the following sequent:*

$$(\forall x)[P(x) \to P(f(x))] \vdash (\forall x)[P(x) \to P(f(f(x)))]$$

For this you will definitely need to keep a copy. (On the left, as it happens)

Exercise 29 *Using the natural deduction rules derive a contradiction from the two assumptions $\neg(\forall x)(\neg\phi(x))$ and $\neg(\exists x)(\phi(x))$.*

3.2 Equality and Substitution

Frege gave a definition of equality in higher-order logic. Equality is a deeply deeply problematic notion so it was really quite brave of Frege to even attempt to define it. His definition of equality says that it is the intersection of all reflexive relations. Recall from definition **??** that a binary relation R is reflexive if $R(w,w)$ holds for all w: (That's what the '$(\forall w)(R(w,w))$' is doing in the formula 3.2 below.) So Frege's definition is

$$x = y \quad \text{iff} \quad (\forall R)[(\forall w)(R(w,w)) \to R(x,y)] \tag{3.1}$$

The first thing to notice is that this definition is second-order! You can tell that by the '$(\forall R)$' and the fact that the 'R' is obviously a predicate letter because of the '$R(w, w)$'.

Notice that this definition is not circular (despite what you might have expected from the appearance of the word 'reflexive') since the *definiendum* does not appear in the *definiens*.

3.2.1 Substitution

Consider the binary relation "every property that holds of x holds also of y and *vice versa*". This is clearly reflexive! If x and y are equal then they stand in this relation (because two things that are equal stand in every reflexive relation, by definition) so they have the same properties. This justifies the rule of substitution. (If you have good French have a look at [**?**]).

$$\frac{A(t) \qquad t = x}{A(x)}\ \text{subst} \tag{3.1}$$

In the rule of substitution you are not obliged to replace *every* occurrence of 't' by 'x'. (This might remind you of the discussion on page 7 where we consider cancelling premisses.)

The following example is a perfectly legitimate use of the rule of substitution, where we replace only the *first* occurrence of 't' by 'x'. In fact this is how we prove that equality is a symmetrical relation!

$$\frac{t = t \qquad t = x}{x = t}\ \text{subst} \tag{3.2}$$

Given that, the rule of substitution could more accurately be represented by

$$\frac{A[t/x] \qquad t = x}{A}\ \text{subst} \tag{3.3}$$

. . . the idea being that A is some formula or other—possibly with free occurrences of 'x' in it—and $A[t/x]$ is the result of replacing all free occurrences of 'x' in A by 't'. This is a bit pedantic, and on the whole our uses of substitution will look more like 3.1 than 3.3.

However we will definitely be using the $A[t/x]$ notation in what follows, so be prepared. Sometimes the $[t/x]$ is written the other side, as

$$[t/x]A. \tag{3.4}$$

This notation is intended to suggest that $[t/x]$ is a function from formulæ to formulæ that is being applied to the formula A.

One thing that may cause you some confusion is that sometimes a formula with a free variable in it will be written in the style "$A(x)$" making the variable explicit. Sometimes it isn't made explicit. When you see the formula in 3.2.1 it's a reasonable bet that the variable 'x' is free in A, or at least could be: after all, there wouldn't be much point in substituting 't' for 'x' if 'x' weren't free, now would it?!

3.2.2 Leibniz's law

"The identity of indiscernibles". This is a principle of second-order logic:

$$(\forall xy)((\forall R)(R(x) \longleftrightarrow R(y)) \rightarrow x = y) \tag{3.1}$$

The converse to 3.1 is obviously true so we can take this as a claim about the nature of equality: $x = y$ iff $(\forall R)(R(x) \longleftrightarrow R(y))$

It's not 100% clear how one would infer that x and y are identical in Frege's sense merely from the news that they have the same *monadic* properties: Frege's definition talks about reflexive relations, which of course are *binary*. The claim that 3.1 characterises equality is potentially contentions. It is known as **Leibniz's Law**.

3.3 Prenex Normal Form

There is a generalisation of CNF and DNF to first-order logic: it's called **Prenex normal form**. The definition is simplicity itself. A formula is in Prenex normal form if it is of the form

$$(Qv_1)(Qv_2)\cdots(Qv_n)(....)$$

where the Qs are quantifiers, and the dots at the end indicate a purely propositional formula: one that contains no quantifiers, and is in conjunctive normal form. All quantifiers have been "pulled to the front".

Exercise 30 *Which of the following formulæ are in Prenex normal form?* Insert some formulae here!!

Theorem 7 *Every formula is logically equivalent to one in PNF.*

To prove this we need to be able to "pull all quantifiers to the front". What does this piece of italicised slang mean? Let's illustrate:

$$(\forall x)F(x) \wedge (\forall y)G(y)$$

is clearly equivalent to

$$(\forall x)(\forall y)(F(x) \wedge G(y))$$

(If everything is green and everything is a frog then everything is both green and a frog, and *vice versa*).

In exercise 26 the point in each case is that in the formula being deduced the scope of the quantifier is larger: it has been "pulled to the front". If we keep on doing this to a formula we end up with something that is in PNF.]

. . . and explain to your flatmates what this has to do with theorem 7.

Explain why PNF is important—why normal form theorems are important in general. It imposes a linear order on the complexity of formulæ.

3.4 Soundness again

At this point we should have a section analogous to section 1.5 where we prove the soundness of natural deduction for propositional logic and section **??** where we prove the soundness of sequent calculus for propositional logic.

Work to be done here

You will discover that it's nowhere near as easy to test predicate calculus formulæ for validity as it is to test propositional formulæ: there is no easy analogue of truth-tables. Despite this there is a way of generating all the truth-preserving principles of reasoning that are expressible with this syntax, and we will be seeing them, and I hope to prove them complete

You must get used to the idea that all notions of logical validity, or of sound inference, can be reduced to a finite set of rules in the way that propositional logic can and predicate calculus can. Given that—as we noted on p **??**—the validity of an argument depends entirely on its syntactic form, perhaps we should not be surprised to find that there are finite mechanical methods for recognising valid arguments. However this holds good only for arguments of a particularly simple kind. If we allow variables to range over predicate letters then things start to go wrong. Opinion is divided on how important is this idea of completeness. If we have something that looks like a set of principles of reasoning but discover that it cannot be generated by a finite set of rules, does that mean it isn't part of logic?

Mention here other notions of validity: true in all finite

In contrast to soundness, *completeness* is hard. See section 3.6

3.5 Hilbert-style systems for first-order Logic

At this point there should be a section analogous to section 1.8. However I think we can safely omit it.

3.6 Semantics for First-order Logic

We arrived at the formulæ of first-order logic by a process of codifying what was logically essential in some scenario or other. Semantics is the reverse process: picking up a formula of LPC and considering what situations could have given rise to it by the kind of codification that we have seen in earlier exercises such as **??**.

A valid formula is one that is true in all models. We'd better be clear what this means! So let's define what a model is, and what it is for a formula to be true in a model.

Signatures, structures, carrier set. Then we can explain again the difference between a first-order theory and a higher-order theory.

The obvious examples of structures arise in mathematics and can be misleading and in any case are not really suitable for our expository purposes here. We can start off with the idea that a structure is a set-with-knobs on. Here is a simple example that cannot mislead anyone.

The carrier set is the set {Beethoven, Handel, Domenico Scarlatti} and the knobs are (well, '*is*' rather than '*are*' because there is only one knob in this case) the binary relation "is the favourite composer of". We would obtain a different structure by adding a second relation: "is older than" perhaps.

Need some formal semantics here!

Now we have to give a rigorous explanation of what it is for a formula to be true in a structure.

Definition 8
A **theory** *is a set of formulæ closed under deduction.*

We say T **decides** *ψ if $T \vdash \psi$ or $T \vdash \neg\psi$.*

Let us extend our use of the '$\mathcal{L}$' notation to write '$\mathcal{L}(T)$' for the language to which T belongs.[1]

A theory T is **complete** *if T decides every closed ϕ in $\mathcal{L}(T)$.*

Definition 9 *A Logic is a theory closed under uniform substitution.*

We need one more technicality: the concept of a **countable language**. A first-order language with a finite lexicon has infinitely many expressions in it, but the set of those expressions is said to be *countable*: that is to say we can count the expressions using the numbers 1, 2, 3, 4 . . . which are sometimes called the *counting numbers* and sometimes called the *natural numbers*.

Insert discussion of countability here

3.6.1 Countably presented rectypes are countable

Mathematicians should be warned that logicians often use the word 'countable' to mean 'countably infinite'. The symbol used for the cardinal number of countably infinite sets is '$\aleph_0$'.

[1] For sticklers:

$$\mathcal{L}(T) =: \bigcup_{s \in T} \mathcal{L}(s)$$

where $\mathcal{L}(s)$ is as defined in the second part of definition **??** on page **??**.

The prime powers trick

"Countably presented" is slang, but in this case we mean that the rectype has countably many founders and countably many functions all of finite arity.

THEOREM 10 *Countably presented rectypes are countable.*

Sketch of proof: The key observation is that the set of finite sets of naturals and the set of finite sequences of naturals are both countable. The function $\lambda x.\Sigma_{n\in x}2^n$ maps finite sets of natural numbers 1-1 to natural numbers. (Make a note of this for later use in finding models of ZF without the axiom of infinity.) We map finite sequences of naturals to naturals by sending–for example–the tuple $\langle 1,8,7,3\rangle$ to $2^{1+1}\cdot 3^{8+1}\cdot 5^{7+1}\cdot 7^{3+1}$). This is the **prime powers trick**.

The elements of a finitely presented (indeed countably presented) rectype can obviously be represented by finite sequences of symbols, and so the prime powers trick is enough to show that every finitely presented rectype is countable. ■

A meal is often made of the fact that (the syntax of) every human language is a rectype–unlike the syntax of any animal language–and that therefore the repertory of possible expressions is infinite in a way that the repertory of meaningful calls available to animals of other species is not. Quite how useful this recursive structure is to those who wish to drive a wedge between human language and animal language is not entirely clear, but it is immensely useful when dealing with artificial languages, since it enables us to exploit structural induction in proving facts about them.

We can enumerate the wffs[2] of a language and then *sequences* of wffs of a language (which is to say, Gödel-proofs, which we will meet on page **??**). This enables us to arithmetise proof theory and eventually to prove the incompleteness theorem. Since this was first done by Gödel with precisely this end in view, any enumeration of formulæ (or register machines or Turing machines, as in chapter **??**, or anything else for that matter) tends to be called **Gödel numbering** or **gnumbering** for short: the 'g' is silent.)

The way to crystallise the information contained in theorem 10 is to develop a nose for the difference between what one might call *finite precision objects* and *infinite precision objects*. Members of finitely presented rectypes are finite precision objects: one can specify them uniformly with only finitely many symbols. In contrast, the reals, for example, are infinite precision objects: there is no way of uniformly notating reals using only finitely many symbols for each real. There is a sort of converse to theorem 10: if a set is countable, then there will be a way of thinking of it (or at least there will be a notation for its members) as a finitely presented rectype. This gives us a rule of thumb: if X is a set that admits a uniform notation for its members, where each member has a finite label, then X is countable, and conversely. This is not the *definition* of countable, but it is the most useful way to tell countable sets from uncountable sets.

EXERCISE 31 *Which of the following sets are countable and which are uncountable, using the above test? The set of*
(i) permutations of $\mathbb{N}$ *that move only finitely many things;*
(ii) permutations of $\mathbb{N}$ *of finite order;*
(iii) algebraic numbers;
(iv) partitions of $\mathbb{N}$ *into finitely many pieces;*
(v) partitions of $\mathbb{N}$ *all of whose pieces are finite;*
(vi) partitions of $\mathbb{N}$ *containing a cofinite piece;*

[2]Logicians' slang, which I shall frequently lapse into. It is an acronym: Well-Formed Formula.

(vii) increasing functions $\mathbb{N} \to \mathbb{N}$*;*
(viii) nonincreasing functions $\mathbb{N} \to \mathbb{N}$ *(*$f(n+1) \leq f(n)$*).*

It is often quite hard to provide explicit bijections between two things that are the same size: a good example is the naturals and the rationals. However, it commonly happens that we find sets X and Y where–although there is no obvious bijection between them–there are nevertheless obvious injections $X \hookrightarrow Y$ and $Y \hookrightarrow X$. For this reason we often have recourse to the Schröder-Bernstein theorem (theorem **??**), which tells us that in those circumstances there is actually a bijection between X and Y after all, and it can even be given explicitly if we are patient.

However, proving that the uncountable sets mentioned in this exercise are all the same size is quite hard even with the aid of theorem **??**!

Exercise 32 *Exhibit injections from the rationals into the naturals, and vice versa.*

The set of natural numbers is usually written with a capital 'N' in a fancy font, for example $\mathbb{N}$. There is some small print to do with the fact that we might have an infinite supply of variables .... After all, there is no limit on the length of expressions so there is no limit on the number of variables that we might use, so we want to be sure we will never run out. The best way to do this is to have infinitely many variables to start with. We can achieve this while still having a finite alphabet by saying that our variables will be not 'x', 'y' ... but 'x', 'x'', 'x''' ... the idea being that you can always make another variable by plonking a '$'$' on the right of a variable. (Notice that the systematic relation that holds between a variable and the new variable obtained from it by whacking it on the right with a '$'$' has no semantics: the semantics that we have cannot see through into the typographical structure of the variables.)

Theorem 11 *Every theory in a countable language can be extended to a complete theory.*

Proof: Suppose T is a theory in a language $\mathcal{L}(T)$ which is countable. Then we count the formulæ in $\mathcal{L}(T)$ as ϕ_1, ϕ_2 ... and define a sequence of theories T_i as follows.

$T_0 = T$ and thereafter

T_{i+1} is to be T_i if T_i decides ϕ_i and is $T_i \cup \{\phi_i\}$ otherwise.

The theory $T_\infty =: \bigcup\{T_i : i \in \mathbb{N}\}$ is now a complete theory. ■

There is no suggestion that this can be done effectively

3.6.2 Completeness

$\in$-terms

For any theory T we can always add constants to $\mathcal{L}(T)$ to denote witnesses to $\exists^n$ sentences in T.

$\exists^n$ sentence? 'witness' eo existential quantifiers not explained yet

Suppose $T \vdash (\exists x)(F(x))$. There is nothing to stop us adding to $\mathcal{L}(T)$ a new constant symbol 'a' and adding to T an axiom $F(a)$. Clearly the new theory will be consistent if T was. Why is this? Suppose it weren't, then we would have a deduction of $\bot$ from $F(a)$. But T also proves $(\exists x)(F(x))$, so we can do a $\exists$-elimination to have a proof of $\bot$ in T. But T was consistent.

picture here

Notice that nothing about the letter 'a' that we are using as this constant tells us that a is a thing which is F. We could have written the constant 'a_F' or something suggestive like that. Strictly it shouldn't matter: variables and constant symbols do not have any internal structure that is visible to the

language[3], and the 'F' subscript provides a kind of spy-window available to anyone *mentioning* the language, but not to anyone merely *using* it. The possibility of writing out novel constants in suggestive ways like this will be useful later.

Exercise 33

1. *Find a proof of the sequent* $\vdash (\exists x)(\forall y)(F(y) \to F(x))$
2. *Find a natural deduction proof of* $(\exists x)(\forall y)(F(y) \to F(x))$
3. *Find a proof of the sequent* $\vdash (\exists x)(F(x) \to (\forall y)(F(y))$
4. *Find a natural deduction proof of* $(\exists x)(F(x) \to (\forall y)(F(y))$

The first item tells us that for any F with one free variable we can invent a constant whose job it is to denote an object which has property F as long as anything does. If there is indeed a thing which has F then this constant can denote one of them, and as long as it does we are all right. If there isn't such a thing then it doesn't matter what it denotes. There is a similar argument for the formula in parts 3 and 4. The appeal to the law of excluded middle in this patter should alert you to the possibility that this result is not constructively correct. (So you should expect to find that you need to have to use the rule of double negation in parts 2 and 4 and will have two formulæ on the right at some point in the proof of parts 1 and 3.

Explain constructively correct

This constant is often written $(\epsilon x)F(x)$. Since it points to something that has F as long as there is something that has F, we can see that

$$(\exists x)(F(x)) \qquad \text{and} \qquad F((\epsilon x)F(x))$$

are logically equivalent. So we have two rules

$$\frac{(\exists x)(F(x))}{F((\epsilon x)F(x))} \qquad \text{and} \qquad \frac{F((\epsilon x)F(x))}{(\exists x)(F(x))}$$

The right-hand one is just a special case of $\exists$-introduction but the left-hand one is new, and we call it ϵ-introduction. In effect it does the work of $\exists$-elimination, because in any proof of a conclusion ϕ using $\exists$-elimination with an assumption $(\exists x)F(x)$ we can replace the constant (as it might be) 'a' in the assumption $F(a)$ being discharged by the ϵ term '$(\epsilon x)F(x)$' to obtain a new proof of ϕ, thus:

$$\dfrac{\begin{array}{c}[A(t)]^{(1)} \\ \vdots \\ C\end{array} \qquad (\exists x)(A(x))}{C}\ \exists\text{-elim}(1) \tag{3.1}$$

with

$$\begin{array}{c}\dfrac{(\exists x)(A(x))}{A((\epsilon x)(A(x)))}\ \epsilon\text{-int} \\ \vdots \\ C\end{array} \tag{3.2}$$

... where, in the dotted part of the second proof, 't' has been replaced by '$(\epsilon x)(A(x))$'

Notice that this gives us an equivalence between a formula that definitely belongs to predicate calculus (co's it has a quantifier in it) and something that

[3] Look again at formula ?? on page ?? and the discussion on page ??

appears not to[4]. Hilbert was very struck by this fact, and thought he had stumbled on an important breakthrough: a way of reducing predicate logic to propositional logic. Sadly he hadn't, but the ϵ-terms are useful gadgets all the same, as we are about to see.

Theorem 12 *Every consistent theory has a model.*

Proof:

Let T_1 be a consistent theory in a countable language $\mathcal{L}(T_1)$.

Now we do the following things

1. Add axioms to T_1 to obtain a complete extension;
2. Add ϵ terms to the language.

Notice that when we add ϵ-terms to the language we add new formulæ: if '$(\epsilon x)F(x))$' is a new ϵ-term we have just added then '$G((\epsilon x)F(x)))$' is a new formula, and T_1 doesn't tell us whether it is to be true or to be false. That is to say $\mathcal{L}(T_1)$ doesn't contain '$(\epsilon x)F(x)$' or '$G((\epsilon x)F(x)))$'. Let $\mathcal{L}(T_2)$ be the language obtained by adding to $\mathcal{L}(T_1)$ the expressions like '$(\epsilon x)F(x)$' and '$G((\epsilon x)F(x)))$'.

We extend T_1 to a new theory in $\mathcal{L}(T_2)$ that decides all these new formulæ we have added. This gives us a new theory, which we will—of course—call T_2.

It's worth thinking about what sort of formulæ we generate. We added terms like $(\epsilon x)(F(x))$ to the language of T_1. Notice that if H is a two-place predicate in $\mathcal{L}(T)$ then we will find ourselves inventing the term $(\epsilon y)H(y,(\epsilon x)F(x))$ which is a term of—one might say—*depth* 2. And there will be terms of depth 3, 4 and so on as we persist with this process. All atomic questions about ϵ terms of depth n are answered in T_{n+1}.

Repeat and take the union of all the theories T_i we obtain in this way: call it T_∞. (Easy to see that all the T_i are consistent—we prove this by induction). T_∞ is a theory in a language $\mathcal{L}_\infty$, and it will be complete. The model for T_∞ will be the structure whose carrier set is the set of ϵ terms we have generated *en route*. All questions about relations between the terms in the domain are answered by T_∞. Does this make it a model of T? Might not T_∞ assert that there is a wombat but yet deny of each term that it is a wombat? This is what we fear ...

The key fact is that if T_∞ asserts that there is a wombat, then that is because one of the T_i asserts that there is a wombat. But if T_i does this, then at stage $i+1$ we add a term and tell it to be a wombat. So there is a term asserted by T_∞ to be a wombat. This means that the thing we fear cannot happen. ■

This is a result of fundamental importance. Any theory that is not actually self-contradictory is a description of *something*. It's important that this holds only for first-order logic. It does not work for second-order logic, and this fact is often overlooked. (If you want a discussion of this, look at appendix **??**). A touching faith in the power of the completeness theorem is what lies behind the widespread error of reifying possibilities into possible worlds. See [**?**].

Notice that this proof gives us something slightly more than I have claimed. If the consistent theory T we started with was a theory in a countable language then the model we obtain by the above method is also countable. It's worth recoding this fact:

Corollary 13 *Every consistent theory in a countable language has a countable model.*

Also Schütte's proof

[4]The 'ϵ' is not a quantifier, but it is a *binder*: something that binds variables. '$\exists$' and '$\forall$' are binders of course, and so is 'λ' which we will meet in chapter **??**

3.7 Interpolation

There is a precise analogue in predicate calculus of the interpolation lemma for propositional logic of section **??**.

THEOREM 14 *The Interpolation Lemma*
If $A \to B$ is a valid formula of first-order logic then there is a formula C containing only predicate letters that appear in **both** *A and B such that $A \to C$ and $C \to B$ are both valid formulæ of first-order logic.*

A proof of this fact is beyond the scope of this course. The proof relies on the **subformula property** mentioned earlier. The disjoint-vocabulary case is intuitively obvious, but it's not at all clear how to do the induction.

Close attention to the details of the proof of the completeness theorem will enable us to prove it and get bounds on the complexity of the interpolating formula. These bounds are not very good!

The interpolation lemma is probably the most appealing of the consequences of the completeness theorem, since we have very strong intuitions about irrelevant information. Hume's famous dictum that one cannot derive an "ought" from an "is" certainly arises from this intuition. The same intuition is at work in the hostility to the *ex falso sequitur quodlibet* that arises from time to time: if there has to be a connection in meaning between the premisses and the conclusion, then an empty premiss—having no meaning—can presumably never imply anything.

3.8 Compactness

Recall section **??** at this point.

3.9 Skolemisation

EXERCISE 34 *Find a proof of the sequent*

$$\forall x \exists y R(x,y) \vdash (\forall x_1)(\exists y_1)(\forall x_2)(\exists y_2)(R(x_1,y_1) \wedge R(x_2,y_2) \wedge (x_1 = x_2 \to y_1 = y_2))$$

[Fit this in somewhere: take a formula of the Σ^1 fragment of second-order logic. Delete the existential quantifiers. The result is a formula in 1st order logic with function letters. If it is refutable then so was the Σ^1 formula we started with. So there is a refutation procedure for the Σ^1 fragment of second-order logic.

Similarly there is a refutation procedure for the set of formulæ true in all finite structures.]

Chapter 4

Lectures Five and Six

4.1 Assorted First-order Theories

1. every poly of odd degree has a root;
2. 0 is not a sum of nontrivial squares;
3. either x or $-x$ has a square root.

Can introduce an order by $x \leq y$ iff $(\exists z)(z^2 + x = y)$. See Chang and Keisler sec 5.4.

A typical way for a theory to arise is as the set of things true in a given structure $\mathcal{M}$. We write this $Th(\mathcal{M})$. Thus $Th(\mathcal{M}) = \{\phi : \mathcal{M} \models \phi\}$. Theories that arise in this way, as the set of things true in a particular structure, are of course complete—simply because of excluded middle.

A related typical way in which a theory can arise is as *the set of all sentences true in a given class of structures.*

Theories that arise in the first way are obviously going to be complete! Surprisingly some theories that arise in the second way can be complete too: DLO is the theory of dense linear orders. It is expressed in a language $\mathcal{L}$(DLO) with equality and one two-place predicate $<$. Its axioms say that $<$ is transitive and irreflexive, and that between any two things there is a third, and that there is no first or last element.

Exercise 35 *Write out the axioms of DLO. Can there be a finite model of DLO?*

It's not hard to show that this theory is complete, using a famous construction of Cantor's. We do this below.

A famous example of an incomplete theory is the theory known as *Peano Arithmetic.* Its incompleteness was proved by Gödel.

4.2 Categoricity

4.2.1 Countably categorical theories and Back-and-forth

DLO; the canonical random graph.

4.2.2 Uncountably categorical theories

Here is a rather nice example of a theory that is *un*countably categorical It has two axioms:

$(\forall x)(\exists! y)(R(x, y))$ and $(\forall y)(\exists! z)(R(x, y))$.

Chapter 5

Possible World Semantics

First we illustrate why constructivists repudiate the law of excluded middle. Some readers may already know the standard horror story about $\sqrt{2}^{\sqrt{2}}$. For those of you that don't—yet—here it is.

Suppose you are given the challenge of finding two irrational number α and β auch that α^β is rational. It is in fact the case that both e and $log_e(2)$ are transcendental but this is not easy to prove. Is there an easier way in? Well, one thing every schoolchild knows is that $\sqrt{2}$ is irrational, so how about taking both α and β to be $\sqrt{2}$? This will work if $\sqrt{2}^{\sqrt{2}}$ is rational. Is it? As it happens, it isn't (but that, too, is hard to prove). If it isn't, then we take α to be $\sqrt{2}^{\sqrt{2}}$ (which we now believe to be irrational—had it been rational we would have taken the first horn) and take β to be $\sqrt{2}$.

α^β is now

$$(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2}\cdot\sqrt{2}} = \sqrt{2}^2 = 2$$

which is rational, as desired. However, we haven't met the challenge. We were asked to *find* a pair $\langle\alpha, \beta\rangle$ of irrationals such that α^β is rational, and we haven't found such a pair. We've proved that there *is* such a pair, and we have even narrowed the candidates down to a short list of two, but we haven't completed the job.[1]

What does this prove? It certainly doesn't straightforwardly show that the law of excluded middle is *false*; what it shows is that there are situations where you don't want to reason with it. There is a difference between proving that there is a widget, and actually getting your hands on the widget. Sometimes it matters, and if you happen to be in the kind of pickle where it matters, then you want to be careful about reasoning with excluded middle.

To get a semantics for constructive logic we need something a great deal more complicated than truth-tables! A first step is to increase the number of truth-values, and we have seen some uses of three-valued truth-tables (see exercise 6 p. 12). However, many-valued truth-tables are not the way to go. For one thing, it can be shown that three-valued truth-tables do not draw the distinctions we want, and no finite number of truth-values will ever do the trick. This is very hard to prove and we won't attempt it here. Another reason is that if we try to capture constructive validity by means of truth-tables we are still thinking of it as as an extensional logic not an intensional logic, and—since that does not correctly engage with the spirit that moves constructive logic—it is unlikely to give us decent semantics for it.

[1] We can actually exhibit such a pair, and using only elementary methods, at the cost of a little bit more work. $log_2(3)$ is obviously irrational: $2^p \neq 3^q$ for any naturals p, q. $log_{\sqrt{2}}(3)$ is also irrational, being $2 \cdot log_2(3)$. Clearly $(\sqrt{2})^{log_{\sqrt{2}}(3)} = 3$.

The final reason for not using many-valued truth-tables is that there is something to hand that is not only more faithful to the constructive conception but is much more fun and susceptible of wider application: that is **Possible World Semantics**, which is the subject of this chapter.

DEFINITION 15 *A* **possible world model** $\mathcal{M}$ *has several components:*

1. *There is a collection of* **worlds** *with a binary relation* $\leq$ *between them; If* $W_1 \leq W_2$ *we say* W_1 *can* **see** W_2*.*
2. *There is also a binary relation between worlds and formulæ, written '*$W \models \phi$*';*
3. *Finally there is a* **designated** *(or 'actual' or 'root') world* W_0^M*.*

We stipulate the following connections between the ingredients:

1. $W \models \bot$ *never holds. We write this as* $W \not\models \bot$*.*
2. $W \models A \wedge B$ *iff* $W \models A$ *and* $W \models B$*;*
3. $W \models A \vee B$ *iff* $W \models A$ *or* $W \models B$*;*
4. $W \models A \rightarrow B$ *iff every* $W' \geq W$ *that* $\models A$ *also* $\models B$*;*
5. $W \models \neg A$ *iff there is no* $W' \geq W$ *such that* $W' \models A$*;*
6. $W \models (\exists x)A(x)$ *iff there is an* x *in* W *such that* $W \models A(x)$*;*
7. $W \models (\forall x)A(x)$ *iff for all* $W' \geq W$ *and all* x *in* W'*,* $W' \models A(x)$*.*

We stipulate further that **for atomic formulæ** ϕ*, if* $W \models \phi$ *and* $W \leq W'$*, then* $W' \models \phi$*. (The idea is that if* $W \leq W'$*, then* W' *in some sense contains more information than* W*.)*

Then we say

$$\mathcal{M} \models A \quad \textit{if} \quad W_0^M \models A$$

4 is a special case of 3: $\neg A$ is just $A \rightarrow \bot$, and no world believes $\bot$!

The relation which we here write with a '$\leq$' is the **accessibility** relation between worlds. We assume for the moment that it is **transitive** and **reflexive**.

Chat about quantifier alternation. There is a case for writing out the definitions in a formal language, on the grounds that the quantifier alternation (which bothers a lot of people) can be made clearer by use of a formal language. The advantage of not using a formal language is that it makes the language-metalanguage distinction clearer.

The $\models$ relation between worlds and propositions is certainly epistemically problematic. For example W believes $\neg p$ iff no world beyond W believes p. This being so, how can anyone in W come to know $\neg p$? They would have to visit all worlds $\geq W$! So this possible worlds talk is not part of an *epistemic* story! This being the case, one should perhaps beware of the danger of taking the "world W believes ϕ" slang too literally. Even if W believes $\neg\phi$ then in some sense it doesn't know that it believes $\neg\phi$... unless of course W includes among its inhabitants all the worlds $\geq W$. But that makes for a scenario far too complicated for us to entertain in a book like this. And it is arguable that it is a scenario of which no coherent account can be given. See [**?**].

The possible worlds semantics is almost certainly not part of a constructivist account of truth or meaning at all. (Remember: we encountered it as the classical logicians' way of making sense of constructive logic!) If it were, the fact that it is epistemically problematic would start to matter.

The relation $\leq$ between worlds is transitive. A model $\mathcal{M}$ believes ϕ (or not, as the case may be) iff the designated world W_0 of $\mathcal{M}$ believes ϕ (or not). When cooking up W_0 to believe ϕ (or not) the recursions require us only to look at worlds $\geq W_0$. This has the effect that the designated world of $\mathcal{M}$ is $\leq$ all other worlds in $\mathcal{M}$. This is why we sometimes call it the 'root' world. This use of the word 'root' suggests that the worlds beyond W_0 are organised into a tree: so if W_1 and W_2 are two worlds that cannot see each other then there is no world they can both see. However we are emphatically *not* making this assumption.

Quantifiers

The rules for the quantifiers assume that worlds don't just believe primitive propositions but also that they have inhabitants. I think we generally take it that our worlds are never empty: every world has at least one inhabitant. However there is no global assumption that all worlds have the same inhabitants. Objects may pop in and out of existence. However we do take the identity relation between inhabitants across possible worlds as a given.

5.1 Language and Metalanguage again

It is very important to distinguish between the stuff that appears to the left of a '$\models$' sign and that which appears to the right of it. The stuff to the right of the '$\models$' sign belongs to the *object language* and the stuff to the left of the '$\models$' sign belongs to the *metalanguage*. So that we do not lose track of where we are I am going to write '$\rightarrow$' for *if–then* in the metalanguage and '`&`' for *and* in the metalanguage instead of '$\wedge$'. And I shall use square brackets instead of round brackets in the metalanguage.

If you do not keep this distinction clear in your mind you will end up making one of the two mistakes below (tho' you are unlikely to make both.)

Remember what the aim of the Possible World exercise was. It was to give people who believe in classical logic a way of making sense of the thinking of people who believe in constructive logic. That means that it's perfectly OK to use classical logic in reasoning with/manipulating stuff to the left of a '$\models$' sign.

For example here is a manœuvre that is perfectly legitimate: if

$$\neg[W \models A \rightarrow B]$$

then it is not the case that

$$(\forall W' \geq W)(W' \models A \ \rightarrow \ W' \models B)$$

So, in particular,

$$(\exists W' \geq W)(W' \models A \ \& \ \neg(W' \models B))$$

The inference drawn here from $\neg\forall$ to $\exists\neg$ is perfectly all right in the classical metalanguage, even though it's not allowed in the constructive object language.

In contrast it is *not* all right to think that—for example—$W \models \neg A \vee \neg B$ is the same as $W \models \neg(A \wedge B)$ (on the grounds that $\neg A \vee \neg B$ is the same as $\neg(A \wedge B)$). One way of warding off the temptation to do is is to remind ourselves—again—that the aim of the Possible World exercise was to give people who believe in classical logic a way of making sense of the thinking of people who believe in constructive logic. That means that it is not OK to use classical logic in reasoning with/manipulating stuff to the right of a '$\models$' sign.

Another way of warding off the same temptation is to think of the stuff after the '$\models$' sign as stuff that goes on in a fiction. You, the reader of a fiction, know things about the characters in the fiction that they do not know about each

other. Just because something is true doesn't mean they know it!! (This is what the literary people call **Dramatic Irony**.)[2]

(This reflection brings with it the thought that reading "$W \models \neg\neg A$" as "W believes not not A" is perhaps not the happiest piece of slang. After all, in circumstances where $W \models \neg\neg A$ there is no suggestion that the fact-that-no-world-$\geq$-W-believes-A is encoded in W in any way at all.)

Could say more about this

Another mistake is to think that we are obliged to use constructive logic in the metalanguage which we are using to discuss constructive logic—to the left of the '$\models$' sign.

I suspect it's a widespread error. It may be the same mistake as the mistake of supposing that you have to convert to Christianity to understand what is going on in the heads of Christians. Christians of some stripes would no doubt agree with the assertion that there are bits of it you can't understand until you convert, but I think that is just a mind-game.

Doesn't this duplicate earlier stuff?

We could make it easier for the nervous to discern the difference between the places where it's all right to use classical reasoning (the metalanguage) and the object language (where it isn't) by using different fonts or different alphabets. One could write "For all W" instead of $(\forall W)\ldots$". That would certainly be a useful way of making the point, but once the point has been made, persisting with it looks a bit obsessional: in general people seem to prefer overloading to disambiguation.

5.1.1 A possibly helpful illustration

Let us illustrate with the following variants on the theme of "there is a Magic Sword." All these variants are classically equivalent. The subtle distinctions that the possible worlds semantics enable us to make are very pleasing.

1. $\neg\forall x\neg MS(x)$
2. $\neg\neg\exists x MS(x)$
3. $\exists x\neg\neg MS(x)$
4. $\exists x MS(x)$

The first two are constructively equivalent as well.

To explain the differences we need the difference between **histories** and **futures.**

- A *future* (from the point of view of a world W) is any world $W' \geq W$.
- A *history* is a string of worlds—an unbounded trajectory through the available futures.

No gaps between worlds...?

$\neg\forall x\neg MS(x)$ and $\neg\neg\exists x MS(x)$ say that every future can see a future in which there is a Magic Sword, even though there might be histories that avoid Magic Swords altogether: *Magic Swords are a permanent possibility: you should never give up hope of finding one.*

How can this be, that every future can see a future in which there is a magic sword but there is a history that contains no magic sword–ever? It could happen like this: each world has precisely two immediate children. If it is a world with a magic sword then those two worlds also have magic swords in them. If it is a world without a magic sword then one of its two children continues swordless, and the other one acquires a sword. We stipulate that the root world contains no magic sword. That way every world can see a world that has a magic sword, and yet there is a history that has no magic swords.

[2]Appreciation of the difference between something being true and your interlocutor knowing it is something that autists can have trouble with. Some animals that have "a theory of other minds" (in that they know that their conspecifics might know something) too can have difficulty with this distinction. Humans seem to be able to cope with it from the age of about three.

$\exists x \neg\neg MS(x)$ says that every history contains a Magic Sword and moreover the thing which is destined to be a Magic Sword is already here. Perhaps it's still a lump of silver at the moment but it will be a Magic Sword one day.

5.2 Some Useful Short Cuts

5.2.1 Double negation

The first one that comes to mind is $W \models \neg\neg\phi$. This is the same as $(\forall W' \geq W)(\exists W'' \geq W')(W'' \models \phi)$. "Every world that W can see can see a world that believes ϕ". Let's thrash this out by hand.

By clause 5 of definition 15

$$W \models \neg(\neg\phi)$$

iff

$$(\forall W' \geq W)\neg[W' \models \neg\phi] \tag{6.1}$$

Now

$W' \models \neg\phi$ iff $(\forall W'' \geq W')\neg[W'' \models \phi]$ by clause 5 of definition 15 so

$\neg[W' \models \neg\phi]$ is the same as $\neg(\forall W'' \geq W')\neg[W'' \models \phi]$ which is

$$(\exists W'' \geq W')(W'' \models \phi).$$

Substituting this last formula for for '$W' \models \neg\phi$' in (6.1) we obtain

$$(\forall W' \geq W)(\exists W'' \geq W')(W'' \models \phi)$$

5.2.2 If there is only one world then the logic is classical

If $\mathcal{M}$ contains only one world—W, say—then $\mathcal{M}$ believes classical logic. Let me illustrate this in two ways:

1. Suppose $\mathcal{M} \models \neg\neg A$. Then $W \models \neg\neg A$, since W is the root world of $\mathcal{M}$. If $W \models \neg\neg A$, then for every world $W' \geq W$ there is $W'' \geq W$ that believes A. So in particular there is a world $\geq W$ that believes A. But the only world $\geq W$ is W itself. So $W \models A$. So every world $\geq W$ that believes $\neg\neg A$ also believes A. So $W \models \neg\neg A \rightarrow A$.

2. W either believes A or it doesn't. If it believes A then it certainly believes $A \vee \neg A$, so suppose W does not believe A. Then no world that W can see believes A. So $W \models \neg A$ and thus $W \models (A \vee \neg A)$. So W believes the law of excluded middle.

We must show that the logic of quantifiers is classical too

The same arguments can be used even in models with more than one world, if the worlds in question can see only themselves.

5.3 Persistence

For atomic formulæ ϕ we know that if $W \models \phi$ then $W' \models \phi$ for all $W' \geq W$. We achieved this by stipulation, and it echoes our original motivation. Even though $\neg\neg(\exists x)(x$ is a Magic Sword) is emphatically not to be the same as $(\exists x)(x$ is a Magic Sword), it certainly is inconsistent with $\neg(\exists x)(x$ is a Magic Sword) and so it can be taken as *prophecy* that a Magic Sword will turn up one day. The idea of worlds as states of knowledge where we learn more as time elapses sits very well with this. By interpreting $\neg\neg(\exists x)(x$ is a Magic Sword) as "Every

future can see a future that contains a Magic Sword" possible world semantics captures the a way in which $\neg\neg(\exists x)(x$ is a Magic Sword) can be incompatible with the nonexistence of Magic Swords while nevertheless not telling us how to find a Magic Sword.

We will say ϕ is **persistent** if whenever $W \models \phi$ then $(\forall W' \geq W)(W' \models \phi)$

We want to prove that all formulæ are persistent.

THEOREM 16 *All formulæ are persistent.*

Proof:

We have taken care of the atomic case. Now for the induction on quantifiers and connectives.

$\neg$ $W \models \neg\phi$ iff $(\forall W' \geq W)\neg(W' \models \phi)$. Therefore if $W \models \neg\phi$ then $(\forall W' \geq \phi)\neg[W' \models \phi]$, and, by transitivity of $\geq$, $(\forall W'' \geq W')\neg[W'' \models \phi]$. But then $\neg[W' \models \neg\phi]$.

$\bigvee$ Suppose ϕ and ψ are both persistent. If $W \models \psi \vee \phi$ then either $W \models \phi$ or $W \models \psi$. By persistence of ϕ and ψ, every world $\geq$ satisfies ϕ (or ψ, whichever it was) and will therefore satisfy $\psi \vee \phi$.

$\bigwedge$ Suppose ϕ and ψ are both persistent. If $W \models \psi \wedge \phi$ then $W \models \phi$ and $W \models \psi$. By persistence of ϕ and ψ, every world $\geq$ satisfies ϕ and every world $\geq$ satisfies ψ and will therefore satisfy $\psi \wedge \phi$.

$\exists$ Suppose $W \models (\exists x)\phi(x)$, and ϕ is persistent. Then there is an x in W which W believes to be ϕ. Suppose $W' \geq W$. As long as x is in W' then $W' \models \phi(x)$ by persistence of ϕ and so $W' \models (\exists x)(\phi(x))$.

$\forall$ Suppose $W \models (\forall x)\phi(x)$, and ϕ is persistent. That is to say, for all $W' \geq W$ and all x, $W' \models \phi(x)$. But if this holds for all $W' \geq W$, then it certainly holds for all $W' \geq$ any given $W'' \geq W$. So $W'' \models (\forall x)(\phi(x))$.

$\rightarrow$ Finally suppose $W \models (A \rightarrow B)$, and $W' \geq W$. We want $W' \models (A \rightarrow B)$. That is to say we want every world beyond W' that believes A to also believe B. We do know that every world beyond W that believes A also believes B, and every world beyond W' is a world beyond W, and therefore believes B if it believes A. So W' believes $A \rightarrow B$.

That takes care of all the cases in the induction. ■

It's worth noting that we have made heavy use of the fact that $\leq$ is transitive. Later we will consider other more general settings where this assumption is not made.

Now we can use persistence to show that this possible world semantics always makes $A \rightarrow \neg\neg A$ comes out true. Suppose $W \models A$. Then every world $\geq W$ also believes A. No world can believe A and $\neg A$ at the same time. ($W \models \neg A$ only if none of the worlds $\geq W$ believe A; one of the worlds $\geq W$ is W itself.) So none of them believe $\neg A$; so $W \models \neg\neg A$.

This is a small step in the direction of a completeness theorem for the possible world semantics.

5.4 Independence Proofs Using Possible world semantics

5.4.1 Some Worked Examples

Challenge 5.4.1.1: Find a countermodel for $A \vee \neg A$

The first thing to notice is that this formula is a classical (truth-table) tautology. Because of subsection 5.2.2 this means that any countermodel for it must contain

more than one world.

The root world W_0 must not believe A and it must not believe $\neg A$. If it cannot see a world that believes A then it will believe $\neg A$, so we will have to arrange for it to see a world that believes A. One will do, so let there be W_1 such that $W_1 \models A$. picture here

Challenge 5.4.1.2: Find a countermodel for $\neg\neg A \vee \neg A$

The root world W_0 must not believe $\neg\neg A$ and it must not believe $\neg A$. If it cannot see a world that believes A then it will believe $\neg A$, so we will have to arrange for it to see a world that believes A. One will do, so let there be W_1 such that $(W_1 \models A)$. It must also not believe $\neg\neg A$. It will believe $\neg\neg A$ as long as every world it can see can see a world that believes A. So there had better be a world it can see that cannot see any world that believes A. This cannot be W_1 because $W_1 \models A$, and it cannot be W_0 itself, since $W_0 \leq W_1$. So there must be a third world W_2 which does not believe A.

Challenge 5.4.1.3: Find a countermodel that satisfies $(A \to B) \to B)$ but does not satisfy $A \vee B$

insert details here

Challenge 5.4.1.4: Find a countermodel for $((A \to B) \to A) \to A$

You may recall from exercise 6 on page 12 that this formula is believed to be false on Planet Zarg. There we had a three-valued truth table. Here we are going to use possible worlds. As before, with $A \vee \neg A$, the formula is a truth-table tautology and so we will need more than one world

Recall that a model $\mathcal{M}$ satisfies a formula ψ iff the root world of $\mathcal{M}$ believes ψ: that is what it is for a model to satisfy ψ. Definition!

As usual I shall write 'W_0' for the root world; and will also write '$W \models \psi$' to mean that the world W believes ψ; and $\neg[W \models \psi]$ to mean that W does not believe ψ.

So we know that $\neg[W_0 \models ((A \to B) \to A) \to A]$.

Now the definition of $W \models X \to Y$ is (by definition 15)

$$(\forall W' \geq W)[W' \models X \ \to \ W' \models Y] \tag{5.1}$$

So since

$$\neg[W_0 \models ((A \to B) \to A) \to A]$$

we know that there must be a $W' \geq W_0$ which believes $((A \to B) \to A$ but does not believe A. (In symbols: $(\exists W' \geq W_0)[W' \models ((A \to B) \to A)$ & $\neg(W' \models A)]$.) Remember too that in the metalanguage we are allowed to exploit the equivalence of $\neg\forall$ with $\exists\neg$. Now every world can see itself, so might this W' happen to be W_0 itself? No harm in trying...

So, on the assumption that this W' that we need is W_0 itself, we have:

1. $W_0 \models (A \to B) \to A$; and
2. $\neg[W_0 \models A]$.

This is quite informative. Fact (1) tells us that every $W' \geq W_0$ that believes $A \to B$ also believes A. Now one of those W' is W_0 itself (Every world can see itself: remember that $\geq$ is reflexive). Put this together with fact (2) which says that W_0 does not believe A, and we know at once that W_0 cannot believe $A \to B$. How can we arrange for W_0 not to believe $A \to B$? Recall the definition 15 above of $W \models A \to B$. We have to ensure that there is a $W' \geq W_0$ that believes A but does not believe B. This W' cannot be W_0 because W_0 does not believe A. So there must be a *new* world (we always knew there would be!)

visible from W_0 that believes A but does not believe B. (In symbols this is $(\exists W' \geq W_0)[W' \models A \quad \& \quad \neg(W' \models B)]$.)

So our countermodel contains two worlds W_0 and W', with $W_0 \leq W'$. $W' \models A$ but $\neg[W_0 \models A]$, and $\neg[W' \models B]$.

Let's check that this really works. We want

$$\neg[W_0 \models ((A \rightarrow B) \rightarrow A) \rightarrow A]$$

We have to ensure that at least one of the worlds beyond W_0 satisfies $(A \rightarrow B) \rightarrow A$ but does not satisfy A. W_0 doesn't satisfy A so it will suffice to check that it does satisfy $(A \rightarrow B) \rightarrow A$. So we have to check (i) that if W_0 satisfies $(A \rightarrow B)$ then it also satisfies A and we have to check (ii) that if W' satisfies $(A \rightarrow B)$ then it also satisfies A. W' satisfies A so (ii) is taken care of. For (i) we have to check that W_0 does not satisfy $A \rightarrow B$. For this we need a world $\geq W_0$ that believes A but does not believe B and W' is such a world.

Challenge 5.4.1.5: Find a model that satisfies $(A \rightarrow B) \rightarrow B$ but does not satisfy $(B \rightarrow A) \rightarrow A$

We must have

$$W_0 \models (A \rightarrow B) \rightarrow B \tag{1}$$

and

$$\neg[W_0 \models (B \rightarrow A) \rightarrow A] \tag{2}$$

By (2) we must have $W_1 \geq W_0$ such that

$$W_1 \models B \rightarrow A \tag{3}$$

but

$$\neg[W_1 \models A] \tag{4}$$

We can now show

$$\neg[W_1 \models A \rightarrow B] \tag{5}$$

If (5) were false then $W_1 \models B$ would follow from (1) and then $W_1 \models A$ would follow from (3). (5) now tells us that there is $W_2 \geq W_1$ such that

$$W_2 \models A \tag{6}$$

and

$$\neg[W_2 \models B] \tag{7}$$

From (7) and persistence we infer

$$\neg[W_1 \models B] \tag{8}$$

and

$$\neg[W_0 \models B] \tag{9}$$

Also, (4) tells us

$$\neg[W_0 \models A]. \tag{10}$$

So far we have nothing to tell us that $W_0 \neq W_1$. So perhaps we can get away with having only two worlds W_0 and W_1 with $W_1 \models A$ and W_0 believing nothing.

W_0 believes $(A \rightarrow B) \rightarrow B$ vacuously: it cannot see a world that believes $A \rightarrow B$ so—vacuously—every world that it can see that believes $A \rightarrow B$ also believes B. However, every world that it can see believes $(B \rightarrow A)$ but it does not believe A itself. That is to say, it can see a world that does not believe A so it can see a world that believes $B \rightarrow A$ but does not believe A so it does not believe $(B \rightarrow A) \rightarrow A$.

5.4.2 Exercises

Exercise 36 *Return to Planet Zarg![3]*
The truth-tables for Zarg-style connectives are on p 12.

1. *Write out a truth-table for* $((p \to q) \to q) \to (p \vee q)$.
   (Before you start, ask yourself how many rows this truth-table will have).
2. *Identify a row in which the formula does not take truth-value* 1.
3. *Find a sequent proof for* $((p \to q) \to q) \to (p \vee q)$.

Exercise 37 *Find a model that satisfies* $(p \to q) \to q$ *but not* $p \vee q$.

It turns out that Zarg-truth-value 1 means "true in W_0 and in W'"; Zarg-truth-value 2 means "true in W'", and Zarg-truth-value 3 means "true in neither"—where W_0 and W_1 are the two worlds in the countermodel we found for Peirce's law. (Challenge 5.4.1.5)

Exercise 38 *Find a model that satisfies* $p \to q$ *but not* $\neg p \vee q$.

Exercise 39 *Find a model that doesn't satisfy* $p \vee \neg p$. *How many worlds has it got? Does it satisfy* $\neg p \vee \neg\neg p$? *If it does, find one that doesn't satisfy* $\neg p \vee \neg\neg p$.

Exercise 40
1. *Find a model that satisfies* $A \to (B \vee C)$ *but doesn't satisfy* $(A \to B) \vee (A \to C))$
2. *Find a model that satisfies* $(A \to B) \wedge (C \to D)$ *but doesn't satisfy* $(A \to D) \vee (B \to C))$
3. *Find a model that satisfies* $\neg(A \wedge B)$ *but does not satisfy* $\neg A \vee \neg B$
4. *Find a model that satisfies* $(A \to B) \to B)$ *and* $(B \to A) \to A$ *but does not satisfy* $A \vee B$. *(Check that in the three-valued Zarg world* $((A \to B) \to B) \wedge ((B \to A) \to A)$ *always has the same truth-table as* $A \vee B$*).*

Exercise 41 *Find countermodels for:*

1. $(A \to B) \vee (B \to A)$;
2. $(\exists x)(\forall y)(F(y) \to F(x))$ *(which is the formula in exercise 33 part 1 on page 45).*

Exercise 42 *Consider the model in which there are two worlds,* W_0 *and* W_1, *with* $W_0 \leq W_1$. W_0 *contains various things, all of which it believes to be frogs;* W_1 *contains everything in* W_1 *plus various additional things, none of which it believes to be frogs. Which of the following assertions does this model believe?*

1. $(\forall x)(F(x))$;
2. $(\exists x)(\neg F(x))$;
3. $\neg\exists x\neg F(x)$;
4. $\neg\neg(\exists x)(\neg F(x))$.

[3] Beware: Zarg is a planet, not a possible world!

Bibliography

[1] H.B. Enderton *A Mathematical Introduction to Logic*.

[2] T.E.Forster *Logic, Induction and Sets*, London Mathematical Society Undergraduate Texts in Mathematics **56** Cambridge University Press.

[3] Quine, W.V. (1962) Mathematical Logic (revised edition) Harper torchbooks 1962

Exam questions.

In the following table

$$\begin{array}{cccccc} & & & p_2 & p_1 & p_0 \\ + & & & p_5 & p_4 & p_3 \\ \hline = & p_9 & p_8 & p_7 & p_6 & \end{array}$$

The propositional letters represent bits of two three-bit words and a four-bit word. The table says that the four-bit word is the sum of the two three-bit words. Write out a propositional theory that captures this.

Devise an exercise to represent the pigeonhole principle in propositional logic.

Propositional letters $p_{i,j}$ for $i = 1, 2, 3$ and $j = 1, 2$. The intended meaning is that $p_{i,j}$ says that pigeon i is in pigeonhole j. Every pigeon is in a pigeonhole

$$(p_{1,1} \vee p_{1,2}) \wedge (p_{2,1} \vee p_{2,2}) \wedge (p_{3,1} \vee p_{3,2})$$

One pigeonhole has at least two pigeons.

$$(p_{1,1} \wedge p_{2,1}) \vee (p_{1,2} \wedge p_{2,2}) \vee (p_{1,1} \wedge p_{3,1}) \vee (p_{1,2} \wedge p_{3,2}) \vee (p_{2,1} \wedge p_{3,1}) \vee (p_{2,2} \wedge p_{3,2})$$