Mathematical Logic

Lorna Gregory

August 14, 2025

Contents

1 Introduction 1

2 Propositional Logic 5
- 2.1 Propositional formulas . . . 5
- 2.2 Truth . . . 13
- 2.3 Proofs . . . 20
- 2.4 Completeness . . . 23

3 Predicate Logic 25
- 3.1 Languages and Structures . . . 25
- 3.2 Formulas . . . 26
- 3.3 Tarski's Truth Definition . . . 31
- 3.4 Proofs . . . 39
- 3.5 Completeness . . . 41

4 Proof of the Completeness Theorem for Predicate Logic (4th year material) 43

5 Application of the Compactness Theorem and Extension by Constants 51

6 Universal Algebra: Substructures and Homomorphisms 55

7 Further Model Theory (4th year material) 57
- 7.1 Definable Sets . . . 57
- 7.2 Elementary Substructures . . . 58
- 7.3 Downwards Löwenheim-Skolem Theorem . . . 59
- 7.4 Upwards Löwenheim-Skolem Theorem . . . 59
- 7.5 Categoricity . . . 59

1 Introduction

These are the notes from last time I taught the course. I will modify them during the course. I will mark in the margin where I plan to get to in each lecture and update this to where we have got to after each lecture (the margin comments in gray are from last time this course was taught). I recommend you keep an eye on the "Conventions" section. I've left blank pages so that the page numbers in the body of the notes don't need to change if the material at the start is modified.

What is Logic? What is Mathematical Logic? What is this course about?

Mathematical Logic is a branch of pure mathematics. The best description I have come across of Mathematical Logic is that Mathematical Logic takes seriously as mathematical objects things other mathematicians deal with informally. For instance, in this course, we will consider "statements about mathematical objects" as mathematical objects in their own right and we will prove theorems about them. As a discipline Mathematical Logic includes Computability Theory, Proof Theory, Model Theory, Set Theory and some branches of Category Theory (amongst other subjects). Logic also has a presence in and connections to Computer Science, Linguistics and Philosophy.

In this course we will be studying propositional logic, predicate logic and make some first steps in model theory.

Propositional Logic is the logic of combining statements. Propositional logic will also provide a warm up for the richer subject of predicate logic. In Predicate logic we are able to consider statements about real mathematical objects such as groups and rings.

Predicate Logic is the logic of quantifiers. This logic allows us to consider statements about real mathematical objects such as groups and rings.

Model Theory builds on Predicate Logic.

Prerequisites and warm-up questions

You won't need to know much of the *content* of your previous courses in order to understand this course. However, you will need a level of mathematical maturity which you won't have unless you have attended them! In particular, you will need to have experience of the role of definitions in pure mathematics and be comfortable with reading and producing proofs. I will also take some examples from your algebra courses.

I have produced some "warm-up" questions for the first week. There won't be a workshop on these questions but you are very welcome to talk to me about their solutions.

Mistakes

If you think you've found a mistake in the notes, please send me an email. I'm also happy to hear about typos. If you don't understand something in these notes (or in lectures) then please send me an email or come to one of my office hours. It is really useful for me to know what students find difficult to understand or what I might not have explained well.

Books and other resources

Although it is not necessary, I highly recommend taking a look at some books on mathematical logic. Many of them will give more details than I am able to give in lectures and/or different perspectives on the material that you might find helpful (or interesting). However, you need to approach them in a mature fashion; very few books, if any, will adopt exactly the conventions of this course.

Plan

Here is a plan of how the hours of the course will be used up to reading week.

	Monday		Thursday	
	$9-10$	$10-11$	$14-15$	$15-16$
Wk 1	L	L	L	L
Wk 2	L	L/W	L	L
Wk 3	L	W	L	L
Wk 4	L	W	L	L
Wk 5	L	W	L	L
Wk 6	Reading Week			

L = Lecture, W = Workshop

Coursework

Coursework 1: The first coursework is worth 10% of the marks for this module. It has 3 components:

- In-class exercise: label formal propositional logic proof (2%) [week 3, 4 or 5]
- In-class exercise: draw a construction tree of a term in predicate logic, compute complexity of a term and give examples of terms of specified complexity in a particular language (2%) [week 4 or 5 or if needed early week 7]
- Written submission to be uploaded to blackboard (6%)

Coursework 2: The first coursework is worth 10% of the marks for this module. It's exact form is to be determined but it will be similar to coursework 1.

This page is intentionally blank.

2 Propositional Logic

2.1 Propositional formulas

Definition 2.1. *An **alphabet** of a **propositional language** $\mathcal{L}$ consists of the following:*

- *A set of **propositional variables** $\mathrm{PROP}(\mathcal{L})$. The elements will usually be denoted p, r, q or $p_1, p_2, p_3, \ldots$.*
- *A set of **connectives** $\{\wedge, \vee, \neg, \rightarrow, \bot\}$.*
- *An open bracket " (" and a close bracket ") " which will collectively be called brackets.*

*We will always assume that $\mathrm{PROP}(\mathcal{L})$ does not contain any connectives or brackets. Each of the connectives has an element of the set $\{0, 1, 2\}$ assigned to it which we will call its **arity**. The connective $\bot$ has arity 0, the connective $\neg$ has arity 1 and all other connectives listed above have arity 2. Connectives with arity 0 are called **logical constants**.*

The table below shows useful information about each of the connectives.

Connective	Arity	Name	How to say it.	How to LaTeX it.
$\wedge$	2	conjunction	and	`\wedge`
$\vee$	2	disjunction	or	`\vee`
$\neg$	1	negation	not	`\neg`
$\rightarrow$	2	implication	implies	`\rightarrow`
$\bot$	0	falsity	false	`\bot`

Definition 2.2. *A **word** in (an alphabet of) a propositional language $\mathcal{L}$ is an n-tuple of elements of $\mathcal{L}$. Unusually, we will write the n-tuple with entries $a_1, a_2, \ldots, a_n$ as $a_1 a_2 \cdots a_n$ rather than $(a_1, \ldots, a_n)$. The **length** of a word is simply the length of tuple i.e. the length of $a_1 a_2 \ldots a_n$ is n. We will call the entries of a word $a = a_1 a_2 \ldots a_n$ the **letters** of a.*

We define the set of propositional formulas inductively.

Definition 2.3. *Let $\mathcal{L}$ be a propositional language. Define $\mathrm{S}_0\mathcal{L}$ to be the set of propositional variables of $\mathcal{L}$ together with the logical constants of $\mathcal{L}$. For each $i \in \mathbb{N}$, we define $\mathrm{S}_i\mathcal{L}$ by induction as follows:*

$$\mathrm{S}_{i+1}\mathcal{L} := \mathrm{S}_i\mathcal{L} \cup \{(\neg t) \mid t \in \mathrm{S}_i\mathcal{L}\} \cup \{(s_1 \,\square\, s_2) \mid s_1, s_2 \in \mathrm{S}_i\mathcal{L} \text{ and } \square \text{ is a connective of arity } 2\}.$$

*The set of **propositional formulas**, denoted $\mathrm{S}\mathcal{L}$, is the union of the sets $\mathrm{S}_i\mathcal{L}$ for $i \in \mathbb{N}_0$. The **complexity** of a propositional formula t is the least $i \in \mathbb{N}_0$ such that $t \in \mathrm{S}_i\mathcal{L}$.*

Note that the arity of a connective determines how it is used to construct propositional formulas.

Example 2.4. *Let $\mathcal{L}$ be the propositional language with set of propositional variables $\mathrm{PROP}(\mathcal{L}) = \{p\}$. The sets of propositional formulas of complexity 0 is*

$$\mathrm{S}_0\mathcal{L} := \{p, \bot\}.$$

There are 16 *propositional formulas of complexity* 0 *and* 1*.*

$$\mathrm{S}_1\mathcal{L} := \{p, \bot, (\neg p), (\neg\bot), (p \wedge p), (p \wedge \bot), (\bot \wedge p), (\bot \wedge \bot), \\ (p \vee p), (p \vee \bot), (\bot \vee p), (\bot \vee \bot), (p \to p), (p \to \bot), (\bot \to p), (\bot \to \bot)\}.$$

Here are a few propositional formulas of complexity 2*:*

$$((p \wedge p) \to \bot), \quad (\neg(\bot \to \bot)), \quad ((p \wedge p) \to (p \vee \bot)).$$

Lemma 2.5. *The set of propositional formulas* $\mathrm{S}\mathcal{L}$ *of a propositional language* $\mathcal{L}$ *is the smallest set* X *of words in the alphabet of* $\mathcal{L}$ *satisfying the following properties:*

(i) All propositional variables and logical constants of $\mathcal{L}$ *are members of* X*.*

(ii) If $s \in X$ *then* $(\neg s) \in X$*.*

(iii) For all connectives $\Box$ *of arity* 2*, if* $s_1, s_2 \in X$ *then* $(s_1 \Box s_2) \in X$*.*

Proof. See Exercise Sheet 1. $\Box$

For simple expressions like $(p \wedge q)$ and $(\neg(\bot \to p))$ it is easy argue that they are propositional formulas.

Example 2.6. *Let* p, q *be propositional variables of a propositional language* $\mathcal{L}$*.*

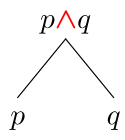
(i) The expression $(p \wedge q)$ *is a propositional formula because* $p, q \in \mathrm{S}_0\mathcal{L}$ *(since they are propositional variables) and so* $(p \wedge q) \in \mathrm{S}_1\mathcal{L}$*.*

(ii) We argue that $(\neg(\bot \to p))$ *is a propositional formula. Since* p *is a propositional variable and* $\bot$ *is a logical constant,* $p, \bot \in \mathrm{S}_0\mathcal{L}$*. Therefore* $(\bot \to p) \in \mathrm{S}_1\mathcal{L}$*. Hence* $(\neg(\bot \to p)) \in \mathrm{S}_2\mathcal{L}$*.*

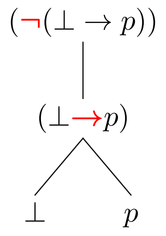
Explaining why a complicated expression is a propositional formula as in the previous example can become tedious and difficult to follow. For this reason we introduce construction trees of propositional formulas.

Construction Trees

Before explaining what a construction tree is, it is helpful to see some examples. The following is the construction tree of the propositional formula $(p \wedge q)$:



Here is the construction tree of $(\neg(\bot \to p))$:



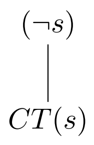
Definition 2.7. *The* ***construction tree*** $CT(t)$ *of a propositional formula* t *is defined recursively based on the definition of a propositional formula. The construction tree of a propositional variable* p *is:*

$$p$$

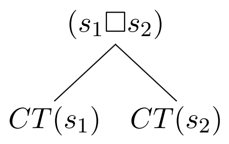
The construction tree of the logical constant $\perp$ *is:*

$$\perp$$

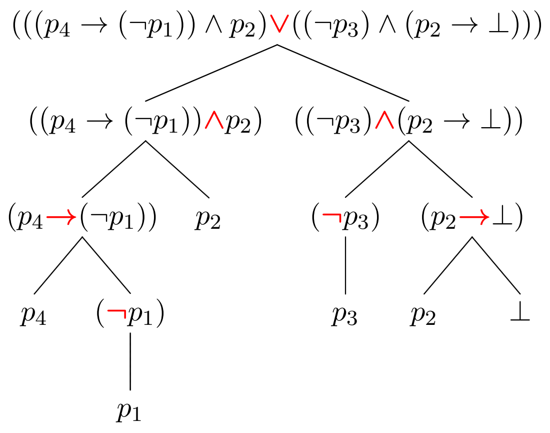
If s *is a propositional formula then* $CT((\neg s))$ *is:*



If s_1 *and* s_2 *are propositional formulas and* $\Box$ *is a connective of arity* 2 *then* $CT((s_1 \Box s_2))$ *is:*



Example 2.8. *The following is the construction tree of the propositional formula* $(((p_4 \to (\neg p_1)) \wedge p_2) \vee ((\neg p_3) \wedge (p_2 \to \perp)))$.



$(((p_4 \to (\neg p_1)) \wedge p_2) \vee ((\neg p_3) \wedge (p_2 \to \perp)))$
$((p_4 \to (\neg p_1)) \wedge p_2)$
$((\neg p_3) \wedge (p_2 \to \perp))$
$(p_4 \to (\neg p_1))$
p_2
$(\neg p_3)$
$(p_2 \to \perp)$
p_4
$(\neg p_1)$
p_3
p_2
$\perp$
p_1

To help you more easily see what is going on, the connective which is introduced at each node is in boldface red.

We will later see an algorithm which given a propositional formula allows you to compute a construction tree for it.

End of L1&L2

Induction on the complexity of propositional formulas

We will frequently prove properties of propositional formulas by "induction on complexity". To prove a statement about all propositional formulas, using this method of proof, we first prove the statement for $\mathrm{S}_0\mathcal{L}$ (this is our base case) and then show that, for all $n \in \mathbb{N}_0$, if the statement holds of $\mathrm{S}_n\mathcal{L}$ then it also holds for $\mathrm{S}_{n+1}\mathcal{L}$.

Here is a simple first example. Before reading the proof of this lemma, look at the definition of a propositional formula and think about why it is true.

Lemma 2.9. *Let $\mathcal{L}$ be a propositional language. Suppose $s \in \mathrm{S}\mathcal{L}$. The number of open brackets (occuring in s is equal to the number of close brackets) occuring in s.*

Proof. For any propositional formula s, define $l[s]$ (respectively $r[s]$) to be the number of open brackets " (" (respectively close brackets ") ") occurring in s. If $s \in \mathrm{S}_0\mathcal{L}$ then s is either a propositional variable or $\perp$. So $l[s] = 0 = r[s]$. Hence the statement of the lemma is true for all $s \in \mathrm{S}_0\mathcal{L}$. For the induction step, suppose that for all $s \in \mathrm{S}_n\mathcal{L}$, $l[s] = r[s]$. Let t be a propositional formula in $\mathrm{S}_{n+1}\mathcal{L}$. Then, examining the definition of $\mathrm{S}_{n+1}\mathcal{L}$, either $t \in \mathrm{S}_n\mathcal{L}$, t is $(\neg s)$ for some $s \in \mathrm{S}_n\mathcal{L}$ or t is $(s_1 \Box s_2)$ for some propositional formulas $s_1, s_2 \in \mathrm{S}_n\mathcal{L}$ and some connective $\Box$ of arity 2.

If $t \in \mathrm{S}_n\mathcal{L}$ then $l[t] = r[t]$ by our induction hypothesis.

If t is $(\neg s)$ then $l[t] = l[(\neg s)] = l[s] + 1$ and $r[t] = r[(\neg s)] = r[s] + 1$. By the induction hypothesis, $l[s] = r[s]$. Therefore $l[t] = r[t]$.

If t is $(s_1 \Box s_2)$ for some propositional formulas $s_1, s_2 \in \mathrm{S}_n\mathcal{L}$ and some connective $\Box$ of arity 2 then $l[(s_1 \Box s_2)] = l(s_1) + l(s_2) + 1$ and $r[(s_1 \Box s_2)] = r[s_1] + r[s_2] + 1$. By the induction hypothesis $l[s_1] = r[s_1]$ and $l[s_2] = r[s_2]$. Therefore

$$l[t] = l[(s_1 \Box s_2)] = l[s_1] + l[s_2] + 1 = r[s_1] + r[s_2] + 1 = r[(s_1 \Box s_2)] = r[t].$$

Thus we have proved the inductive step. Hence the lemma holds by induction on the complexity of propositional formulas. $\Box$

We now state a "meta" theorem. The reason it is "meta" is that we deliberately don't define what we mean by "property". It is intended to give a template for what needs to be shown in order to prove something by induction on complexity of formulas.

For a property X of propositional formulas and, in this instance, a propositional formula t, we will write $X(t)$ to mean that property X holds for t. For example, X could be the property that t has the same number of open brackets as close brackets as in the previous lemma.

Meta Theorem 2.10. *Let X be a property of propositional formulas and let $\mathcal{L}$ be a propositional language. Suppose that the following hold:*

1. *For all propositional variables p, $X(p)$ and $X(\perp)$.*
2. *For all $s \in \mathrm{S}\mathcal{L}$, if $X(s)$ then $X((\neg s))$.*
3. *For all $s_1, s_2 \in \mathrm{S}\mathcal{L}$, $X(s_1)$ and $X(s_2)$ implies $X(s_1 \wedge s_2)$.*
4. *For all $s_1, s_2 \in \mathrm{S}\mathcal{L}$, $X(s_1)$ and $X(s_2)$ implies $X(s_1 \vee s_2)$.*
5. *For all $s_1, s_2 \in \mathrm{S}\mathcal{L}$, $X(s_1)$ and $X(s_2)$ implies $X(s_1 \to s_2)$.*

Then $X(t)$ for all $t \in \mathrm{S}\mathcal{L}$.

Unique construction

The next theorem essentially says that every propositional formula has a unique construction tree.

Theorem 2.11 (Unique Construction Theorem). *For any propositional formula t, exactly one of the following is true:*

1. *t is a propositional variable.*
2. *t is $\bot$.*
3. *t is of the form $(\neg s)$ for a unique propositional formula s.*
4. *t is of the form $(s_1 \Box s_2)$ for a unique connective $\Box$ of arity 2 and a unique pair of propositional formulas (s_1, s_2).*

Without the "exactly one" and "unique", the theorem easily follows from the definition of a propositional formula. To deal with the uniqueness part, we need some definitions and a lemma.

Definition 2.12. *Let x be a word in a propositional language $\mathcal{L}$. If x is of the form yz for words y and z in $\mathcal{L}$ then we call y a* ***proper left subword*** *of x. A word is a* ***left subword*** *of x if it is either equal to x or is a proper left subword of x.*

Lemma 2.13. *Let $\mathcal{L}$ be a propositional language and let $t \in \mathrm{S}\mathcal{L}$. If w is a proper left subword of t then $l(w) < r(w)$ where $l(w)$ denotes the number of open brackets occurring in w and $r(w)$ denotes the number of close brackets occurring in w.*

Proof. We prove the statement by induction on the complexity of t. If $t \in \mathrm{S}_0\mathcal{L}$ then t has no proper left subwords. So the statement is vacuously true. We now prove the induction step. Suppose that the statement is true for propositional formulas in $\mathrm{S}_i\mathcal{L}$. Take $t \in \mathrm{S}_{i+1}\mathcal{L} \backslash \mathrm{S}_i\mathcal{L}$. We now have 2 cases to consider. If t is $(\neg s)$ for some $s \in \mathrm{S}_i\mathcal{L}$ and w is a proper left subword of t then either w is $($, $(\neg$ or $(\neg y$ where y is a left subword of s. It's clear that $l(w) > r(w)$ in the first 2 cases. So suppose that t is $(\neg y$ for y a left subword of s. By the induction hypothesis and 2.9, $l(y) \geq r(y)$. Therefore $l((\neg y) = 1 + l(y) \geq 1 + r(y) > r(y) = r((\neg y)$ as required.

If t is of the form $(s_1 \Box s_2)$ for $s_1, s_2 \in \mathrm{S}_i\mathcal{L}$ and w is a proper left subword of t then either w is $($, $(y$ where y is a left subword of s_1, $(s_1\Box$ or $(s_1\Box z$ where z is a left subword of s_2. If w is $($ then $l(w) = 1 > 0 = r(w)$. If y is a left subword of s_1 then by the induction hypothesis and 2.9, we know that $l(y) \geq r(y)$. Therefore $l((y) = l(y) + 1 > r(y) = r((y)$ and $l((y\Box) = l(y) + 1 > r(y) = r((y\Box)$ as required. We now suppose that z is a subword of s_2. Then $l(z) \leq r(z)$ by 2.9 and the induction hypothesis. So

$$\begin{aligned}
l((s_1\Box z) &= 1 + l(s_1) + l(z) \\
&= 1 + r(s_1) + l(z) \qquad \text{(by 2.9)} \\
&\geq 1 + r(s_1) + r(z) \\
&= 1 + r((s_1\Box z) \\
&> r((s_1\Box z)
\end{aligned}$$

as required. □

For future use we record the following remark.

Remark 2.14. *(a) The first letter of any propositional formula is either "(", a propositional variable or $\bot$.*

(b) All propositional formulas of length ≥ 2, start as $(\neg$, $(($, $(p$ or $(\bot$ where p is a propositional variable.

Proof. (a) Let $t \in \mathrm{S}\mathcal{L}$. If $t \in \mathrm{S}_0\mathcal{L}$ then the first and only letter of t is a propositional variable or $\bot$. Otherwise, it follows directly from the definition of a propositional formula that the first letter of t is $($.

(b) All propositional formulas not in $\mathrm{S}_0\mathcal{L}$ are of the form $(\neg s)$ or $(s_1 \,\square\, s_2)$ where s_1 and s_2 are propositional formulas and $\square$ is a connective of arity 2. Formulas of the form $(\neg s)$ are as described by the statement of the lemma, so we just need to consider those of the form $(s_1 \,\square\, s_2)$. The second letter of $(s_1 \,\square\, s_2)$ is the first letter of s_1. The statement now follows from (a). $\square$

We are now ready to prove the Unique Construction Theorem.

Proof of the Unique Construction Theorem. It follows from the definition of a propositional formula that each propositional formula is of one of the required forms.

Clearly a propositional formula which satisfies 1 or 2 does not satisfy any of the other cases. If a propositional formula satisfies 3 then its first 2 letters are $(\neg$. If a propositional formula satisfies 4 then its first 2 letters are either $(($, $(p$ or $(\bot$ for $p \in \mathrm{PROP}(\mathcal{L})$. Thus no propositional formula can satisfy 3 and 4.

We now just need to show the uniqueness in 3 and 4. If t is $(\neg s)$ and $(\neg s')$ for propositional formulas s and s'. Then clearly $s = s'$. So we consider the case where t is $(s_1 \square s_2)$ and $(s_1' \square' s_2')$. Then either s_1 is a left subword of s_1' or s_1' is a left subword of s_1. Without loss of generality, suppose s_1 is a left subword of s_1'. If s_1 is a proper left subword then the number of open brackets of s_1 is strictly greater than the number of close brackets of s_1. But since s_1 is a propositional formula, its number of open brackets is equal to its number of close brackets. Therefore $s_1 = s_1'$. It now follows that $\square = \square'$ and then that $s_2 = s_2'$ as required. $\square$

An algorithm to compute the construction tree of a propositional formula

When we introduced construction trees we saw the construction tree of the propositional formula $(((p_4 \to (\neg p_1)) \wedge p_2) \vee ((\neg p_3) \wedge (p_2 \to \bot)))$. Given the construction tree it was reasonably easy to check it was correct and hence confirm that $(((p_4 \to (\neg p_1)) \wedge p_2) \vee ((\neg p_3) \wedge (p_2 \to \bot)))$ is indeed a propositional formula. We will now see an algorithm to compute the construction tree of a propositional formula based on the proof of the Unique Construction Theorem.

Case 0: If a propositional formula is in $\mathrm{S}_0\mathcal{L}$ then we just write down the formula.

For propositional formulas not in $\mathrm{S}_0\mathcal{L}$, we split into 3 cases suggested by 2.14.

Case 1: The first 2 letters of t are $(\neg$.

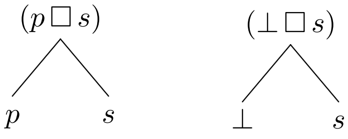
In this case t is of the form $(\neg s)$ and the construction tree starts:

$$\begin{tikzcd}(\neg s) \arrow[d, dash] \\ s\end{tikzcd}$$

To continue, we now just need to find the construction tree of s.

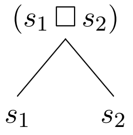
Case 2: The first 2 letters of t are "$(p$" or "$(\perp$" for some propositional variable p.

In this case t is of the form $(p \,\square\, s)$ or $(\perp \square\, s)$ for some propositional formula s and connective $\square$ of arity 2. So the construction tree starts respectively:



Case 3: The first 2 letters of t are "$(($".

In this case t is of the form $(s_1 \,\square\, s_2)$ and the first letter of s_1 is (. So the construction tree of t starts:



But how do we find s_1, $\square$ and s_2?

Since the first letter of s_1 is (, s_1 is not in $\mathrm{S}_0\mathcal{L}$. By 2.13, for all proper left subwords y of s, the number of open brackets in y is strictly less than the number of close brackets in y. Thus s_1 is the shortest left subword of $s_1 \square s_2)$ with equal number of open and close brackets.

Hence, in order to find s_1, we *just* need to count open and close brackets in $(s_1 \,\square\, s_2)$ to compute the last close bracket in s_1.

We number the first (open) bracket in $(s_1 \,\square\, s_2)$ with 1. We continue from left to right labelling each open bracket with $i+1$ if the previous bracket was labelled i and each close bracket $i-1$ if the previous bracket was labelled $i-1$. We continue until we label a close bracket with a 1. Then s_1 is the formula starting with the open bracket labelled 2 and ending with the close bracket labelled 1.

This process is easier to understand once you have seen an example. Consider the propositional formula

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

Assuming this is indeed a propositional formula, we know that it is the form $(s_1 \,\square\, s_2)$ and we want to find s_1, $\square$ and s_2. Here is the propositional formula with the brackets numbered:

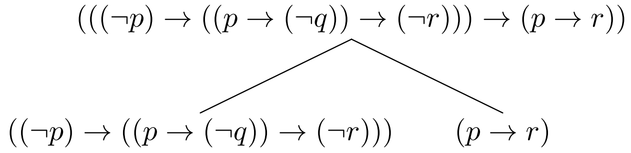
$$\underset{1}{(}\underset{2}{(}\underset{3}{(}\neg p\underset{2}{)} \to \underset{3}{(}\underset{4}{(}p \to \underset{5}{(}\neg q\underset{4}{)}\underset{3}{)} \to \underset{4}{(}\neg r\underset{3}{)}\underset{2}{)}\underset{1}{)} \to (p \to r))$$

Here it is with s_1 and s_2 labelled:

$$\underset{1}{(}\overbrace{\underset{2}{(}\underset{3}{(}\neg p\underset{2}{)} \to \underset{3}{(}\underset{4}{(}p \to \underset{5}{(}\neg q\underset{4}{)}\underset{3}{)} \to \underset{4}{(}\neg r\underset{3}{)}\underset{2}{)}\underset{1}{)}}^{s_1} \to \overbrace{(p \to r)}^{s_2})$$

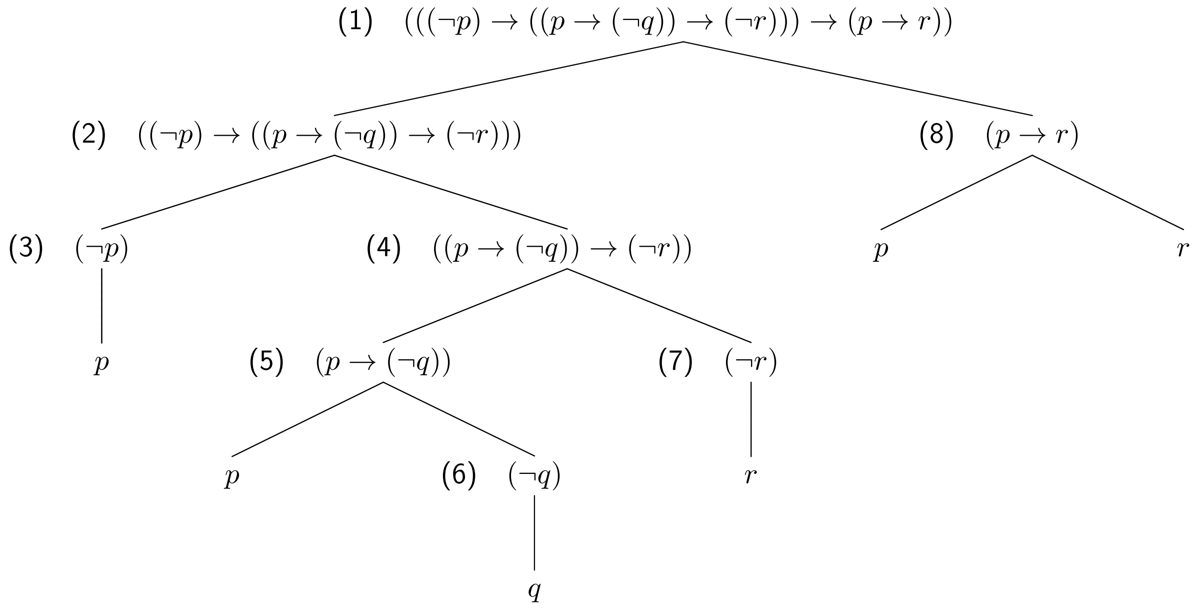
The last connective introduced in the construction of $(s_1 \square s_2)$ is $\to$ i.e. $\square$ is $\to$ and s_2 is $(p \to r)$.

So the start of the construction tree is


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

We will now see a(n almost) full worked example of applying the construction tree algorithm to the propositional formula $(((\neg p) \to ((p \to (\neg q)) \to (\neg r))) \to (p \to r))$.

End of L3&L4



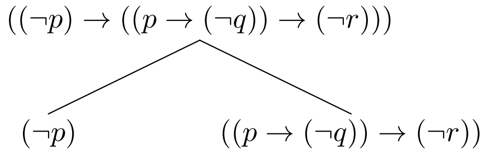
(1) $(((\neg p) \to ((p \to (\neg q)) \to (\neg r))) \to (p \to r))$
(2) $((\neg p) \to ((p \to (\neg q)) \to (\neg r)))$
(8) $(p \to r)$
(3) $(\neg p)$
(4) $((p \to (\neg q)) \to (\neg r))$
p
r
p
(5) $(p \to (\neg q))$
(7) $(\neg r)$
p
(6) $(\neg q)$
r
q

(1) The propositional formula $(((\neg p) \to ((p \to (\neg q)) \to (\neg r))) \to (p \to r))$ starts with ((so we apply case 3 and count the brackets. We've already done this step when describing the algorithm.

(2) The propositional formula $((\neg p) \to ((p \to (\neg q)) \to (\neg r)))$ starts with ((so we apply case 3 and count the brackets as follows:

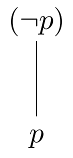
$$\underset{1}{(}\underset{2}{(}\neg p\underset{1}{)} \to ((p \to (\neg q)) \to (\neg r)))$$

So we get the construction tree



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

(3) The propositional formula $(\neg p)$ starts with $(\neg$ so we apply case 1 and get the construction tree



(4) The propositional formula $((p \to (\neg q)) \to (\neg r))$ starts with ((so we apply case 3 and count brackets as follows:

$$\underset{1}{(}\underset{2}{(}p \to \underset{3}{(}\neg q)\underset{2}{)}\underset{1}{)} \to (\neg r))$$

So we get the construction tree

$$\begin{tikzcd}[column sep=small]
 & ((p \to (\neg q)) \to (\neg r)) \arrow[dl, dash] \arrow[dr, dash] & \\
(p \to (\neg q)) & & (\neg r)
\end{tikzcd}$$

(5) The propositional formula $(p \to (\neg q))$ starts with $(p$ and p is a propositional variable so we apply case 2 and get the construction tree

$$\begin{tikzcd}[column sep=small]
 & (p \to (\neg q)) \arrow[dl, dash] \arrow[dr, dash] & \\
p & & (\neg q)
\end{tikzcd}$$

(6) and **(7)** are like **(3)**.

(8) The propositional formula $(p \to r)$ starts with $(p$ and p is a propositional variable so we apply case 2 and get the construction tree

$$\begin{tikzcd}[column sep=small]
 & (p \to r) \arrow[dl, dash] \arrow[dr, dash] & \\
p & & r
\end{tikzcd}$$

Remark 2.15. *Although I haven't justified this, if the construction tree algorithm goes wrong at any point then the word we started with is not a propositional formula. This roughly follows from the proof of the unique readability theorem. This is important because it means a computer can not only compute construction trees but determine whether a word is a propositional formula or not.*

2.2 Truth

The truth of the statement "5 is an odd number and 7 is prime" does not really depend on exactly the statements "5 is an odd number" and "7 is prime". What is important is that both "5 is an odd number" and is true "7 is prime" are true statements.

The truth of a propositional formula should only depend on the truth of the propositional variables it is made up from. Although we have not formally assigned any meaning to connectives, the names we have given indicate what meaning we might want them to have.

In the following definition the symbol $\mathbb{T}$ represents "true" and $\mathbb{F}$ represents "false". We think of the valuation v as telling us which propositional formulas are true and which are false according to v.

Definition 2.16. *A **valuation** is a function $v : \mathrm{S}\mathcal{L} \to \{\mathbb{T}, \mathbb{F}\}$ which satisfies the following properties for all $s, t \in \mathrm{S}\mathcal{L}$:*

1. $v(\perp) = \mathbb{F}$
2. $v((\neg s)) = \begin{cases} \mathbb{F}, & \textit{if } v(s) = \mathbb{T}; \\ \mathbb{T}, & \textit{otherwise.} \end{cases}$
3. $v((t \wedge s)) = \begin{cases} \mathbb{T}, & \textit{if } v(t) = \mathbb{T} \textit{ and } v(s) = \mathbb{T}; \\ \mathbb{F}, & \textit{otherwise.} \end{cases}$

4. $v((t \vee s)) = \begin{cases} \mathbb{F}, & \textit{if } v(s) = \mathbb{F} \textit{ and } v(t) = \mathbb{F}; \\ \mathbb{T}, & \textit{otherwise.} \end{cases}$

5. $v((s \to t)) = \begin{cases} \mathbb{F}, & v(s) = \mathbb{T} \textit{ and } v(t) = \mathbb{F}; \\ \mathbb{T}, & \textit{otherwise.} \end{cases}$

We say that a propositional formula $s \in \mathrm{S}\mathcal{L}$ is ***true (respectively false) under the valuation*** *v if $v(s) = \mathbb{T}$ (respectively if $v(s) = \mathbb{F}$).*

The least controversial parts of this definition are for the connectives are $\wedge$ (and) and $\neg$ (not). We have already discussed $\wedge$ (and). I hope you agree that if a statement is true then its negation should be false and conversely if a statement is false then its negation should be true. For example the statement "All mathematicians are female" is false and so its negation "Not all mathematicians are female" is true.

We now discuss "$\vee$" (or). In everyday life "or" is often treated a exclusive. However, in mathematics, "or" is usually meant non-exclusively. This give rise to the following (not funny) mathematician's/logician's joke:

UNSUSPECTING MEMBER OF THE NON-MATHEMATICAL POPULATION: Would you like a cup of coffee or a cup of tea?
MATHEMATICIAN/LOGICIAN: Yes, please.

The most controversial part of this definition is the condition for $\to$ (implies). As with "or" we could just say this is our convention, get used to it. Since our intention is to give a formal definition of truth, this would be a bit unsatisfactory. First note that this definition does not match up with informal speech even amongst mathematicians. For instance, if I claimed that

"The first isomorphism theorem for groups implies the intermediate value theorem (for continuous functions)."

many of my colleagues and your lecturers would reasonably disagree. On the other hand, most of us would accept that

"If $n = 1$ then $(n-1)(n-2) = 0$"

is true for all $n \in \mathbb{N}$. Although we are not yet studying a logic with quantifiers, it becomes clear that the rule for implication must be as we have defined it when we consider statements involving universal quantification (i.e. "for all" statements) like the following.

$$\forall\, x((x > 3) \to (x^2 > 3))$$

This statement is true if for each $r \in \mathbb{R}$, the statement "$((r > 3) \to (r^2 > 3))$" is true. If we set $r = 1$ then $(r > 3)$ is false and $r^2 > 3$ is false. If we set $r = 2$ then $r > 3$ is false and $r^2 > 3$ is true. If we set $r = 4$ then $r > 3$ is true and $r^2 > 3$ is also true.

Proposition 2.17. *Let $\mathcal{L}$ be a propositional language.*

(i) If $w : \mathrm{PROP}(\mathcal{L}) \to \{\mathbb{T}, \mathbb{F}\}$ is any function then there is a unique valuation $\mathrm{S}\mathcal{L} \to \{\mathbb{T}, \mathbb{F}\}$ such that $v(p) = w(p)$ for all $p \in \mathrm{PROP}(\mathcal{L})$.

(ii) If $t \in \mathrm{S}\mathcal{L}$ and v, w are valuations which agree on all of the propositional variables occurring in t then $v(t) = w(t)$.

Sketch proof. (i) Let $w : \mathrm{PROP}(\mathcal{L}) \to \{\mathbb{T}, \mathbb{F}\}$. For each $i \in \mathbb{N}_0$, we define $v_i : \mathrm{S}_i\mathcal{L} \to \{\mathbb{T}, \mathbb{F}\}$, by induction on i, which satisfies properties 1-5 of the definition of a valuation and such that $v_i(p) = w(p)$ for all $p \in \mathrm{PROP}(\mathcal{L})$. For each $i \in \mathbb{N}_0$, v_{i+1} will be the unique such function extending v_i.

We define v_0 by setting $v_0(p) = w(p)$ for all $p \in \mathrm{PROP}(\mathcal{L})$. In order that v_0 satisfies 1-5, we must define $v_0(\bot) = \mathbb{F}$.

Suppose we have already defined $v_i : \mathrm{S}_i\mathcal{L} \to \{\mathbb{T}, \mathbb{F}\}$. Each element of $\mathrm{S}_{i+1}\mathcal{L}$ is either in $\mathrm{S}_i\mathcal{L}$ or is of the form $(\neg s)$ or $(s_1 \square s_2)$ for some $s, s_1, s_2 \in \mathrm{S}_i\mathcal{L}$ and connective $\square$ of arity 2. For each connective extend v_i according to the rules for that connective given in the definition of a valuation. For instance, if $t = (\neg s)$ then define $v_{i+1}(t) = \mathbb{F}$ if $v_i(s) = \mathbb{T}$ and define $v_{i+1}(t) = \mathbb{T}$ if $v_i(s) = \mathbb{F}$. The function v_{i+1} is well-defined thanks to the Unique Construction Theorem. Moreover v_{i+1} is the unique function defined on $\mathrm{S}_{i+1}\mathcal{L}$ satisfying 1-5 of the definition of a valuation. Let $v : \mathrm{S}\mathcal{L} \to \{\mathbb{T}, \mathbb{F}\}$ be defined by $v(s) = v_i(s)$ for each $s \in \mathrm{S}_i\mathcal{L}$. Then v satisfies 1-5 of the definition of a valuation because each v_i does. Moreover it is the unique valuation extending w because each v_i is the unique function defined on $\mathrm{S}_i\mathcal{L}$ which satisfies 1-5 and extends w.

(ii) This is left as an exercise. □

The rules for valuations are displayed in the following two (truth) tables:

p	$(\neg p)$
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$

p	q	$p \wedge q$	$p \vee q$	$p \to q$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$

We have seen that if $s \in \mathrm{S}\mathcal{L}$ and v is a valuation on $\mathrm{S}\mathcal{L}$ then the value of $v(s)$ only depends on the value of $v(p)$ for propositional variables which occur in s (i.e. are letters of s).

Suppose that the propositional variables which occur in $s \in \mathrm{S}\mathcal{L}$ are $p_1, \ldots, p_n$. A **truth table** for s is a table with columns headed $p_1, p_2, \ldots, p_n, s$. If we ignore the column headed s then the rows of the table display all possible configurations of $\mathbb{T}$ and $\mathbb{F}$. The entry of each row of the column labelled s displays the value of $v(s)$ for a valuation with $v(p_i)$ equal to the ith entry of that row.

Here is an example of a truth table.

Example 2.18.

p_1	p_2	p_3	$(p_1 \to (p_2 \wedge p_3))$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$

Definition 2.19. *A propositional formula t is called a **tautology** if $v(t) = \mathbb{T}$ for all valuations v and **unsatisfiable** if $v(t) = \mathbb{F}$ for all valuations v.*

Remark 2.20. *To check if a propositional formula t is a tautology we just need to check that $v(t) = \mathbb{T}$ for valuations on the language with propositional variables occurring in t.*

For examples, to check if $((p_1 \vee (\neg p_1)) \vee p_2)$ is a tautology, we only need to consider valuations on $\mathrm{S}\mathcal{L}'$ where $\mathcal{L}'$ is the propositional language with $\mathrm{PROP}(\mathcal{L}') = \{p_1, p_2\}$.

A propositional formula s is a tautology if the column in its truth table headed s has all entries $\mathbb{T}$. We give a few tautologies which will be important for the proof system we will introduce in the next section.

End of L5&L6

Proposition 2.21. *For all propositional formulas r, s and t, the following propositional formulas are tautologies.*

(i) $(s \to (t \to s))$

(ii) $((s \to (t \to r)) \to ((s \to t) \to (s \to r)))$

(iii) $((\neg(\neg s)) \to s)$

Proof. (i) The following truth table shows that the propositional formula $(s \to (t \to s))$ is a tautology for any propositional formulas s and t.

s	t	$(t \to s)$	$(s \to (t \to s))$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$

(ii) This is on your exercise sheet.
(iii) The following truth table shows that $((\neg(\neg s)) \to s)$ is a tautology.

s	$(\neg s)$	$(\neg(\neg s))$	$((\neg(\neg s)) \to s)$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$

□

We introduce a useful abbreviation.

Definition 2.22. *For all propositional formulas t and s, define $(t \leftrightarrow s)$ to be the propositional formula $((t \to s) \wedge (s \to t))$. We call the symbol $\leftrightarrow$ biimplication.*

Remark 2.23.

- *For all propositional formulas t and s,*
$$v((t \leftrightarrow s)) = \begin{cases} \mathbb{T}, & \text{if } v(t) = v(s); \\ \mathbb{F}, & \text{otherwise.} \end{cases}$$
- *If s, t are propositional formulas of complexity l and m respectively then the complexity of $(t \leftrightarrow s)$ is $\max\{l, m\} + 2$.*

Definition 2.24. *Let $\mathcal{L}$ be a propositional language, let $t \in \mathrm{S}\mathcal{L}$ and let $S \subseteq \mathrm{S}\mathcal{L}$. We write $S \models t$ and say S **logically implies** t (or t **is a logical consequence of** S) if $v(t) = \mathbb{T}$ for all valuations v such that $v(s) = \mathbb{T}$ for all $s \in S$.*

If $S \subseteq \mathrm{S}\mathcal{L}$ and $t_1, \ldots, t_n, u \in \mathrm{S}\mathcal{L}$ then we, abuse notation and, write $S, t_1, \ldots, t_n \models u$ instead of $S \cup \{t_1, \ldots, t_n\} \models u$ and $t_1, \ldots, t_n \models u$ instead of $\{t_1, \ldots, t_n\} \models u$. Under this convention, $\models u$ means u is a tautology.

Example 2.25. *For a set of propositional formulas S, $S \models \bot$ means that there are no valuations for which $v(s) = \mathbb{T}$ for all $s \in S$.*

Remark 2.26. *If $S \models t$ and $S \models (t \to u)$ then $S \models u$.*

Proof. Suppose $S \models t$ and $S \models (t \to u)$. Let v be a valuation such that $v(s) = \mathbb{T}$ for all $s \in S$. So $v(t) = \mathbb{T}$ and $v(t \to u) = \mathbb{T}$. If $v(u) = \mathbb{F}$ then, since $v(t) = \mathbb{T}$, $v(t \to u) = \mathbb{F}$ which contradicts our assumptions. Therefore $v(u) = \mathbb{T}$. □

Definition 2.27. *We say propositional formulas s and t are **logically equivalent**, and write $s \equiv t$, if $v(s) = v(t)$ for all valuations v.*

Remark 2.28. *For a propositional language $\mathcal{L}$, the relation $\equiv$ is an equivalence relation on $\mathrm{S}\mathcal{L}$.*

Lemma 2.29. *For propositional formulas r, s, t, the following logical equivalences hold.*

1. $(\neg s) \equiv (s \to \bot)$
2. $(s \vee t) \equiv ((\neg s) \to t) \equiv ((s \to \bot) \to t)$
3. $(s \wedge t) \equiv (\neg(s \to (\neg t)))$
4. $(s \wedge t) \equiv (t \wedge s)$
5. $(s \vee t) \equiv (t \vee s)$
6. $(\neg(s \wedge t)) \equiv ((\neg s) \vee (\neg t))$
7. $(s \wedge t) \equiv (\neg((\neg s) \vee (\neg t)))$
8. $((r \wedge s) \wedge t) \equiv (r \wedge (s \wedge t))$
9. $((r \vee s) \vee t) \equiv (r \vee (s \vee t))$

Proof. These equivalences can be checked using truth tables. Here is an example.

s	t	$(s \vee t)$	$(t \vee s)$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

We see that $(s \vee t) \equiv (t \vee s)$ because the column headed $(s \vee t)$ and the column headed $(t \vee s)$ are equal.

□

Lemma 2.30. *Suppose that $s, t \in \mathrm{S}\mathcal{L}$. The following are equivalent.*

1. $s \equiv t$;

2. *$s \models t$ and $t \models s$;*

3. *$\models s \leftrightarrow t$; and*

4. *$s \leftrightarrow t$ is a tautology.*

Proof. Exercise (mainly in remembering definitions). $\square$

Normal Forms

We introduce some useful notation defined by induction. Define $\bigwedge_{i=1}^{1} s_1$ to be s_1 and $\bigvee_{i=1}^{1} s_1$ to be s_1. For $k \in \mathbb{N}$, we define

$$\bigwedge_{i=1}^{k+1} s_i \quad \text{to be} \quad (\left(\bigwedge_{i=1}^{k} s_i\right) \wedge s_{k+1}) \qquad \text{and} \qquad \bigvee_{i=1}^{k+1} s_i \quad \text{to be} \quad (\left(\bigvee_{i=1}^{k} s_i\right) \vee s_{k+1}).$$

We record how these new symbols behave with respect to valuations and logical equivalence.

Lemma 2.31. *Let $s_1, \ldots, s_n \in \mathrm{S}\mathcal{L}$ and let $v : \mathrm{S}\mathcal{L} \to \{\mathbb{T}, \mathbb{F}\}$ be a valuation. Then*

1. *$v(\bigwedge_{i=1}^{n} s_i) = \mathbb{T}$ if and only if $v(s_i) = \mathbb{T}$ for all $i = 1, \ldots, n$;*

2. *$v(\bigvee_{i=1}^{n} s_i) = \mathbb{T}$ if and only if $v(s_i) = \mathbb{T}$ for some $i = 1, \ldots, n$;*

3. *$\bigwedge_{i=1}^{n} s_i \equiv (\neg \bigvee_{i=1}^{n} (\neg s_i))$; and*

4. *$\bigvee_{i=1}^{n} s_i \equiv (\neg \bigwedge_{i=1}^{n} (\neg s_i))$.*

Proof. Each statement is proved by induction on n. I will prove the first statement. For $n = 1$ the statement is trivial since $\bigwedge_{i=1}^{1} s_i$ is s_1. Suppose the statement is true for n. Then $\bigwedge_{i=1}^{n+1} s_i$ is $((\bigwedge_{i=1}^{n} s_i) \wedge s_{n+1})$. So $v(\bigwedge_{i=1}^{n+1} s_i) = \mathbb{T}$ if and only if $v(\bigwedge_{i=1}^{n} s_i) = \mathbb{T}$ and $v(s_{i+1}) = \mathbb{T}$. By the induction hypothesis, this holds if and only if $v(s_i) = \mathbb{T}$ for all $1 \le i \le n$ and $v(s_{n+1}) = \mathbb{T}$ as required. $\square$

This gives a corollary.

Corollary 2.32. *If $s_1, \ldots, s_n \in \mathrm{S}\mathcal{L}$ and $t_1, \ldots, t_m \in \mathrm{S}\mathcal{L}$ are such that $\{s_1, \ldots, s_n\} = \{t_1, \ldots, t_m\}$ then*

$$\bigwedge_{i=1}^{n} s_i \equiv \bigwedge_{i=1}^{m} t_i$$

and

$$\bigvee_{i=1}^{n} s_i \equiv \bigvee_{i=1}^{m} t_i$$

Definition 2.33. *A propositional formula is in **disjunctive normal form (DNF)** if it is of the form $\bot$, $(\neg\bot)$ or*

$$\bigvee_{i=1}^{l} \bigwedge_{j=1}^{m_i} g_{ij}$$

where each g_{ij} is either a propositional variable or the negation of a propositional variable.

Theorem 2.34 (The disjunctive normal form theorem). *If t is a propositional formula then there exists a propositional formula s in DNF such that $t \equiv s$. Moreover, assuming some propositional variables occur in t, if $\{p_1, \dots, p_n\}$ are the propositional variables occurring in t then we can pick s of the form*

$$\bigvee_{i=1}^{l} \bigwedge_{j=1}^{m_i} g_{ij}$$

where $l \leq 2^n$, $m_i \leq n+1$ and each g_{ij} is of the form p_k or $(\neg p_k)$ for $1 \leq k \leq n$.

Proof. If there are no propositional variables occurring in t then either $t \equiv \bot$ or $t \equiv (\neg\bot)$.

Suppose the propositional variables occurring in t are $\{p_1, \dots, p_n\}$. If t is unsatisfiable then $t \equiv (p_1 \wedge (\neg p_1))$. Suppose t is not unsatisfiable. Let $v_1, \dots, v_l$ be the valuations on $\mathrm{S}\mathcal{L}$ where $\mathrm{PROP}(\mathcal{L}) = \{p_1, \dots p_n\}$ such that $v_i(t) = \mathbb{T}$. For each $i = 1, \dots, l$ and $j = 1, \dots, n$, define

$$g_{ij} = \begin{cases} p_j, & \text{if } v_i(p_j) = \mathbb{T}; \\ (\neg p_j), & \text{if } v_i(p_j) = \mathbb{F}. \end{cases}$$

Let v be any valuation on $\mathrm{S}\mathcal{L}$. Then $v(\bigwedge_{j=1}^{n} g_{ij}) = \mathbb{T}$ if and only if $v = v_i$. Therefore $v(\bigvee_{i=1}^{l} \bigwedge_{j=1}^{n} g_{ij}) = \mathbb{T}$ if and only if $v = v_i$ for some $1 \leq i \leq l$. Thus $\bigvee_{i=1}^{l} \bigwedge_{j=1}^{n} g_{ij}$ is of the required form and logically equivalent to t. We see that $l \leq 2^n$ because the number of valuations on $\mathrm{S}\mathcal{L}$ is 2^n. □

End of L7 & L8

This proof gives a method to find a propositional formula in DNF which is equivalent to a given formula. There is an exercise on your exercise sheet which asks you to do this in a few examples.

Adequate sets of connectives

In this subsection we need to expand our definition of a connective. Initially, in 2.1, they were a set of 5 symbols $\{\bot, \neg, \wedge, \vee, \to\}$ each of which has an associated number in $\{0, 1, 2\}$ called the arity of the connective which determined how they could be used to form propositional formulas. At this point, apart from what they look like as symbols and what we called them, there was no real difference between $\wedge, \vee$ and $\to$. In 2.2, we differentiated our 5 connectives by defining how they behave with valuations i.e. how they behaved with respect to truth. Later in 2.2, we introduced a new connective $\leftrightarrow$ as an abbreviation. In this section a connective will be a symbol with an arity in the set $\{0, 1, 2\}$ together with rules for how it interacts with valuations. We will usually present this final data as a truth table. For example, let $\star$ be a connective of arity 2 defined by the truth table

s	t	$(s \star t)$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$

Most of the time we will only be interested in the connectives defined in the previous subsections.

Definition 2.35. *Let Val_n be the set of valuations on the propositional language $\mathcal{L}$ with $\mathrm{PROP}(\mathcal{L}) := \{p_1, \dots, p_n\}$. Each propositional formula t in $\mathrm{S}\mathcal{L}$ defines a* ***truth function*** *$f_t : Val_n \to \{\mathbb{T}, \mathbb{F}\}$ defined by $f_t(v) := v(t)$.*

Definition 2.36. *A set of connectives Y is **adequate** if for each $n \in \mathbb{N}$ and every function $f : Val_n \to \{\mathbb{T}, \mathbb{F}\}$ there exists a propositional formula t with propositional variables in $\{p_1, \ldots, p_n\}$ in which only the connectives from Y occur such that $f = f_t$.*

Proposition 2.37. *The set $\{\wedge, \vee, \neg\}$ is an adequate set of connectives.*

Proof. This follows from the proof of the Disjunctive Normal Form Theorem. □

Corollary 2.38. *A set of connectives Y is adequate if and only if for each $n \in \mathbb{N}$ and propositional formula s with propositional variables in $\{p_1, \ldots, p_n\}$, there exists a propositional formula t with propositional variables in $\{p_1, \ldots, p_n\}$ in which only the connectives from Y occur such that $s \equiv t$.*

Lemma 2.39. *The set of connectives $\{\neg, \vee\}$ is adequate.*

Proof. For any propositional formulas s, t, $(s \wedge t) \equiv (\neg((\neg s) \vee (\neg t)))$. It now follows from 2.37 that $\{\neg, \vee\}$ is adequate. □

Lemma 2.40. *The set of connectives $\{\to, \bot\}$ is adequate.*

Proof. For any propositional formula s, $(s \to \bot) \equiv (\neg s)$. For any propositional formulas s, t, $(s \vee t) \equiv ((s \to \bot) \to t)$. It now follows from 2.39 that $\{\to, \bot\}$ is adequate. □

Example 2.41. *The Sheffer stroke $|$ is an arity 2 connective with truth table:*

p	q	$(p\|q)$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$

You will show on an exercise sheet that $\{|\}$ is an adequate set of connectives.

2.3 Proofs

In the previous section we introduced the notion of logical implication. We wrote $S \models t$ to mean if all $s \in S$ are true then t is true. This is a semantic notion (i.e. relates to meaning). In this section we introduce formal proofs which we will call deductions. We will use the notation $S \vdash t$ to mean t can be deduced from S. After introducing our formal proof system, the main aim of the section will be to prove that if $S \vdash t$ then $S \models t$ and that if $S \models t$ then $S \vdash t$. Informally, anything we can prove is true and anything that is true is provable. The first of these statements is a pretty minimal requirement for a proof system and is quite easy to prove. The second is much harder (and more impressive).

There are many deduction systems. I have chosen Hilbert style deductions because they are the quickest to define and also allow shorter proofs of the most important theorems. However, it's not very easy to come up with deductions in Hilbert style systems.

Before defining what a deduction is we make a modification to our propositional language. We saw in the previous section that the set $\{\to, \bot\}$ is an adequate set of connectives. This means that a propositional language just using those connectives is as expressive as any other propositional language. From this section onwards we will view

- $(\neg s)$ as an abbreviation for $(s \to \bot)$;

- $(s \vee t)$ as an abbreviation for $((\neg s) \to t)$; and
- $(s \wedge t)$ as an abbreviation for $(\neg(s \to (\neg t)))$.

The first thing we need in order to define our proof system are some axioms.

Definition 2.42. *The axioms of propositional logic are*

1. $(s \to (t \to s))$;
2. $((s \to (t \to r)) \to ((s \to t) \to (s \to r)))$; *and*
3. $((\neg(\neg s)) \to s)$.

where s, t, r are any propositional formulas.

What I've called axioms here are in fact axiom schemas. Each of $1.$, $2.$ and $3.$ actually give infinitely many single axioms i.e. if p is a propositional variable then $((\neg(\neg p)) \to p)$ and $((\neg(\neg(\bot \to p))) \to (\bot \to p))$ are (different) instances of axiom $3.$. The most important thing about these axioms is that they are all tautologies. Any set of tautologies containing this set would work just as well (although if you take the set large enough then the results you prove become less impressive). Finally, without using abbreviations the final axiom can be written as $((s \to (s \to \bot)) \to s)$.

Definition 2.43. *Let S be a set of propositional formulas. A* ***deduction*** *from S is a finite tuple $(t_1, t_2, \dots, t_n)$ of propositional formulas such that for each $1 \leq k \leq n$ one of the following is true:*

1. *t_k is an axiom;*
2. *$t_k \in S$; or*
3. *there exist $i, j < k$ such that t_i is $(t_j \to t_k)$.*

We will call a deduction $(t_1, \dots, t_n)$ from S with $t_n = t$ ***a deduction of*** *t* ***from*** *S and write $S \vdash t$ to mean there is a deduction of t from S.*

We call S the set of **premisses** of the deduction. We call the final rule "**modus ponens**" which I will sometimes abbreviate MP. End of L9 & L10

Presenting formal deductions as tuples of propositional formulas is mathematically concise but difficult for humans to read. Moreover, it doesn't tell us why a tuple is a deduction. To display a deduction in a way that is readable for humans we write the entries of a deduction $(t_1, \dots, t_n)$ as a numbered list and next to each line of the list we write the rule we are using to deduce that line i.e. is it a premise, an axiom or are we using modus ponens. Here is an example.

Let p, q, r be propositional formulas. The following tuple of propositional formulas is a deduction of $(p \to r)$ from $\{(p \to q), (q \to r)\}$.

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

In human readable form this is written as follows. We will refer to the comments at the side like "premiss", "instance of axiom 1" and "modus ponens using 5. and 6. as the justification of the step.

1.	$(q \to r)$	premiss
2.	$((q \to r) \to (p \to (q \to r)))$	instance of axiom 1.
3.	$(p \to (q \to r))$	modus ponens using 1. and 2.
4.	$((p \to (q \to r)) \to ((p \to q) \to (p \to r)))$	instance of axiom 2.
5.	$((p \to q) \to (p \to r))$	modus ponens using 3. and 4.
6.	$(p \to q)$	premiss
7.	$(p \to r)$	modus ponens using 5. and 6.

Lemma 2.44. *For any propositional formula s, $\vdash (s \to s)$.*

Proof. We need to write down a formal deduction of $(s \to s)$.

1.	$(s \to ((s \to s) \to s))$	instance of axiom 1.
2.	$((\underline{\underline{s}} \to (\underline{(s \to s)} \to \utilde{s})) \to ((\underline{\underline{s}} \to \underline{(s \to s)}) \to (\underline{\underline{s}} \to \utilde{s})))$	instance of axiom 2.
3.	$((s \to (s \to s)) \to (s \to s))$	modus ponens using 1. and 2.
4.	$(s \to (s \to s))$	instance of axiom 1.
5.	$(s \to s)$	modus ponens using 3. and 4.

The colours and underlining in the deduction are there purely to highlight the propositional formulas occurring in the instance of axiom 2. □

We will freely use the contents of the following remark.

Remark 2.45. *The following are immediate consequences of the definition of a deduction.*

(i) If $(t_1, \dots, t_n)$ is a deduction from S then so is $(t_1, \dots, t_m)$ for all $m \leq n$. In particular, $S \vdash t_m$.

(ii) If $(t_1, \dots, t_n)$ and $(s_1, \dots, s_m)$ are deductions from S then so is their concatenation, that is $(t_1, \dots, t_n, s_1, \dots, s_m)$.

(iii) If $t \in S$ then $S \vdash t$.

(iv) If t is an instance of an axiom of propositional logic then $\vdash t$.

(v) If $S' \vdash t$ and $S' \subseteq S$ then $S \vdash t$.

Theorem 2.46 ("Proofs are finite")**.** *Let $\mathcal{L}$ be a propositional language. For all $S \subseteq \mathrm{S}\mathcal{L}$ and $t \in \mathrm{S}\mathcal{L}$, the following are equivalent:*

(i) $S \vdash t$.

(ii) There is a finite subset $S' \subseteq S$ with $S' \vdash t$.

Proof. Exercise - check all the details. □

Theorem 2.47 (Deduction Theorem)**.** *Let s, t be propositional formulas and let S be a set of propositional formulas. Then*

$$S \vdash (s \to t) \iff S \cup \{s\} \vdash t$$

Proof. ($\Rightarrow$): If $(t_1, \ldots, t_n)$ is a deduction of $(s \to t)$ from S then $(t_1, \ldots, t_n, s, t)$ is a deduction of t from $S \cup \{s\}$. This is because if $(t_1, \ldots, t_n)$ is a deduction from S then it is also a deduction from $S \cup \{s\}$. Therefore $(t_1, \ldots, t_n, s)$ is a deduction from $S \cup \{s\}$ because s is now a premiss. Since t_n is $(s \to t)$, using modus ponens applied to $t_n = (s \to t)$ and s, we see that $(t_1, \ldots, t_n, s, t)$ is a deduction of t from $S \cup \{s\}$.
($\Leftarrow$): Let $(t_1, t_2, \ldots, t_n)$ be a deduction from $S \cup \{s\}$ where $t_n = t$. We will show by induction on i that $S \vdash (s \to t_i)$.

Suppose $i = 1$. Then t_1 is either a premiss, that is, $t_1 \in S \cup \{s\}$ or t_1 is an axiom. If t_1 is s then 2.44 shows that $S \vdash (s \to t_1)$. If $t_1 \in S$ or t_1 is an axiom then the following is a deduction of $(s \to t_1)$ from S.

1.	t_1	premiss/axiom
2.	$(t_1 \to (s \to t_1))$	instance of axiom 1.
3.	$(s \to t_1)$	modus ponens using 1. and 2.

Suppose that $S \vdash (s \to t_l)$ for all $l < i$. We want to show that $S \vdash (s \to t_i)$. If t_i is an axiom or a member of $S \cup \{s\}$ then we can prove $S \vdash (s \to t_i)$ as in the $i = 1$ case. So suppose that there are $j, k < i$ such that t_k is $(t_j \to t_i)$. By the induction hypothesis $S \vdash (s \to t_j)$ and $S \vdash (s \to (t_j \to t_i))$. By 2.45, the concatenation of a deduction of $(s \to t_j)$ from S and a deduction of $(s \to (t_j \to t_i))$ from S is a deduction from S. Moreover this deduction has $(s \to t_j)$ (say entry n) and $(s \to (t_j \to t_i))$ (say entry m) as entries. Hence, we can then extend this deduction by adding the entries

$m+1$.	$((s \to (t_j \to t_i)) \to ((s \to t_j) \to (s \to t_i)))$	instance of axiom 2.
$m+2$.	$((s \to t_j) \to (s \to t_i))$	modus ponens with m. and $m+1$.
$m+3$.	$(s \to t_i)$	modus ponens with n. and $m+2$.

Therefore we have a deduction of $(s \to t_i)$ from S. □

2.4 Completeness

It seems reasonable to hope that all propositional formulas that we can prove with our formal proof system are true according to our definition of truth. This property is known as soundness. Soundness together with the property that all true statements are provable in our formal proof system is known as completeness.

Proposition 2.48 (Soundness Theorem). *Let $\mathcal{L}$ be a propositional language, $S \subseteq \mathrm{S}\mathcal{L}$ and $t \in \mathrm{S}\mathcal{L}$. If $S \vdash t$ then $S \models t$.*

Proof. Suppose that $S \vdash t$. Let $(t_1, \ldots, t_n)$ be a deduction from S with $t_n = t$. We will prove by induction on i that $S \models t_i$.

First the base case i.e. $i = 1$. Either $t_1 \in S$ or t_1 is an instance of one of the axioms of propositional logic. It follows directly from the definition of "$\models$" that if $t_1 \in S$ then $S \models t_1$. We have already shown in 2.21 that each instance of our axioms is a tautology i.e. for each instance r of the axioms of propositional logic $\models r$. Therefore if t_1 is an axiom then $\models t_1$ and hence $S \models t_1$.

Now suppose that $i > 1$ and that for all $l < i$, $S \models t_l$. If $t_i \in S$ or t_i is an axiom then proceed as in the base case. The remaining case to deal with is when there exists $j, k < i$ such that t_k is $(t_j \to t_i)$. By our induction hypothesis $S \models t_j$ and $S \models (t_j \to t_i)$. Now 2.26 states that $S \models t_i$.

Therefore, by induction on i, $S \models t_i$ for all i and hence $S \models t_n$. □

End of L11 & L12

Definition 2.49. *We say a set of propositional formulas S is* ***inconsistent*** *if $S \vdash \bot$ (and* ***consistent*** *if $S \nvdash \bot$).*

Theorem 2.50 (Completeness Theorem). *Let $\mathcal{L}$ be a propositional language, $S \subseteq \mathrm{S}\mathcal{L}$ and $t \in \mathrm{S}\mathcal{L}$. Then $S \models t$ if and only if $S \vdash t$.*

The reverse direction is the Soundness Theorem. At first glance I have no idea how to prove the forward direction of the completeness theorem. So we consider a special case, namely "$S \models \bot$ implies $S \vdash \bot$". Now the contrapositive of this statement is "$S \nvdash \bot$ implies $S \not\models \bot$". This says that if S is consistent then there is a valuation such that $v(s) = \mathbb{T}$ for all $s \in S$.

Before we get on with the proof of the Completeness Theorem, we record a couple of statements we will need in the proof.

Remark 2.51. *For all propositional formulas s, t, we have that $\bot \vdash t$ and $s, (\neg s) \vdash t$.*

Proof. See Exercise Sheet 4. □

It follows from 2.51 that a set of propositional formulas $S \subseteq \mathrm{S}\mathcal{L}$ is inconsistent if and only if $S \vdash t$ for all $t \in \mathrm{S}\mathcal{L}$.

Proposition 2.52. *Let $\mathcal{L}$ be a propositional language and let $S \subseteq \mathrm{S}\mathcal{L}$. If S is consistent (i.e. $S \nvdash \bot$) then there is a valuation such that $v(s) = \mathbb{T}$ for all $s \in S$.*

Proof. We will prove the proposition under the assumption that the set of propositional variables and hence the set of propositional formulas is countable. This is not necessary for the statement to be true but it is necessary for this proof to work.

Claim: For any propositional formula t, either $S \cup \{t\}$ or $S \cup \{(\neg t)\}$ is consistent.

Suppose that $S \cup \{t\}$ is inconsistent, that is $S \cup \{t\} \vdash \bot$. By the deduction theorem, $S \vdash (t \to \bot)$. So $S \vdash (\neg t)$. Therefore the consistency of S implies the consistency of $S \cup \{(\neg t)\}$. So we have proved the claim.

Enumerate the propositional formulas as t_i for $i \in \mathbb{N}$. We will define consistent sets S_i for $i \in \mathbb{N}_0$ such that $S_0 := S$ and $S_{i+1} \supseteq S_i$ for all $i \in \mathbb{N}_0$. Supposing we have already defined S_i then let $S_{i+1} := S_i \cup \{t_{i+1}\}$ if this set is consistent and let $S_{i+1} := S_i \cup \{(\neg t_{i+1})\}$ otherwise. By the claim, S_{i+1} is consistent.

Now let $S_* := \bigcup_{i \in \mathbb{N}_0} S_i$. Suppose, for a contradiction, that S_* is not consistent. Then $S_* \vdash \bot$. But then, since proofs are finite, there is some finite subset $S' \subseteq S$ such that $S' \vdash \bot$. Since S' is finite, it is contained in S_i for some i. Therefore $S_i \vdash \bot$. This contradicts the fact that each S_i is consistent. Therefore S_* is consistent.

We now show that S_* is deductively closed i.e. if $S_* \vdash t$ then $t \in S_*$. Suppose $S_* \vdash t$ and, for a contradiction, that $t \notin S_*$. Then $(\neg t) \in S_*$. So $S_* \vdash (t \to \bot)$ and $S_* \vdash t$. Therefore $S_* \vdash \bot$ which contradicts the fact that S_* is consistent.

Now define $v : \mathrm{S}\mathcal{L} \to \{\mathbb{T}, \mathbb{F}\}$ by setting $v(t) = \mathbb{T}$ if $t \in S_*$ and $v(t) = \mathbb{F}$ otherwise. We will now show that v is a valuation.

Since we now view $\vee, \wedge$ and $\neg$ as abbreviations, in order to check that v is a valuation, we just need to check the parts of the definition of a valuation for $\bot$ and $\to$. Since S_* is consistent, $\bot \notin S_*$. Therefore $v(\bot) = \mathbb{F}$. To deal with $(s \to t)$, we consider 3 cases:

Case $v(s) = \mathbb{F}$: In this case $s \notin S_*$ and we want to show that $(s \to t) \in S_*$ By definition of S_*, this means $(\neg s) \in S_*$. So, it is enough to show that $(s \to \perp) \vdash (s \to t)$. By the deduction theorem this is equivalent to $(s \to \perp), s \vdash t$. That this is true is part of 2.51.

Case $v(s) = \mathbb{T}$ **and** $v(t) = \mathbb{F}$: In this case $s \in S_*$, $t \notin S_*$ and we want to show $(s \to t) \notin S_*$. Suppose for a contradiction that $(s \to t) \in S_*$. Then, since $s \in S_*$, the following is a deduction of t from S_*:

1.	$(s \to t)$	premiss
2.	s	premiss
3.	t	modus ponens using 1. and 2.

But this would mean $t \in S_*$ because S_* is deductively closed. So we have a contradiction and hence $(s \to t) \notin S_*$ as required.

Case $v(s) = \mathbb{T}$ **and** $v(t) = \mathbb{T}$: In this case $s, t \in S_*$ and we want to show $(s \to t) \in S_*$. Since $(t \to (s \to t))$ is an instance of axiom 1. of propositional logic, $S_* \vdash (t \to (s \to t))$. By the deduction theorem, since $t \in S_*$, $S_* \vdash (s \to t)$. Since S_* is deductively closed, $(s \to t) \in S_*$ as required.

Hence we have shown that v is a valuation and $v(s) = \mathbb{T}$ for all $s \in S$. □

Proof of the Completeness Theorem. The reverse direction is the Soundness Theorem, so we prove the forward direction. Suppose that $S \models t$. So if v is a valuation with $v(s) = \mathbb{T}$ for all $s \in S$ then $v(t) = \mathbb{T}$ and hence $v((\neg t)) = \mathbb{F}$. Hence $S \cup \{(\neg t)\}$ is unsatisfiable. Therefore $S \cup \{(\neg t)\} \models \perp$. So, by 2.52, $S \cup \{(\neg t)\} \vdash \perp$. Hence, by the deduction theorem, $S \vdash (\neg(\neg t))$. Since $((\neg(\neg t)) \to t)$ is an instance of axiom 1. of propositional logic $S \vdash ((\neg(\neg t)) \to t)$. Therefore $S \vdash t$ as required. □

3 Predicate Logic

3.1 Languages and Structures

We want to define a language which is rich enough to talk about most mathematical objects (or at least most of those which are sets with extra structure).

Definition 3.1. *A (first order)* ***language*** *consists of*

- 3 *mutually disjoint sets:* $\mathcal{R}$ *the set of* ***relation symbols****,* $\mathcal{F}$ *the set of* ***function symbols*** *and* $\mathcal{C}$ *the set of* ***constant symbols****; and*
- *an* ***arity function*** $\lambda : \mathcal{R} \cup \mathcal{F} \to \mathbb{N}$*.*

For any relation symbol $R \in \mathcal{R}$ *we will refer to* $\lambda(R)$ *as the* ***arity*** *of* R *and for any function symbol* F *we will refer to* $\lambda(F)$ *as the* ***arity*** *of* F*.*

To specify a language $\mathcal{L}$*, we will sometimes write* $\mathcal{L} := \langle \mathcal{R}; \mathcal{F}; \mathcal{C} \rangle$, $\mathcal{L} := \langle R_1, \dots, R_n, F_1, \dots F_m, c_1, \dots, c_n \rangle$ *or* $\mathcal{L} := \langle R_1, \dots, R_n; F_1, \dots F_m; c_1, \dots, c_n \rangle$ *where* $R_1, \dots, R_n$ *is the list of relations symbols,* $F_1, \dots F_m$ *is the list of function symbols and* $c_1, \dots, c_n$ *is the list of constant symbols of* $\mathcal{L}$*. If we do this then in order to fully specify the language, we still need to give the arity of each of the relation and function symbols. If we specify a language with the second notation, then we also need to say which symbols are relation symbols, which are function symbols and which are constant symbols.*

Definition 3.2. *Let* $\mathcal{L}$ *be a language. An* $\mathcal{L}$***-structure*** $\mathcal{A}$ *is a non-empty set* A *called the* ***domain*** *(or universe or underlying set) of* $\mathcal{A}$ *together with* *(i) a subset $R^{\mathcal{A}}$ of $A^{\lambda(R)}$ for each $R \in \mathcal{R}$;*

(ii) a function $F^{\mathcal{A}}$ from $A^{\lambda(F)} \to A$ for each $F \in \mathcal{F}$; and

(iii) an element $c^{\mathcal{A}} \in A$ for each $c \in \mathcal{C}$.

*We call $R^{\mathcal{A}}$, respectively $F^{\mathcal{A}}$, respectively $c^{\mathcal{A}}$ the **interpretation** of R, respectively F, respectively c in $\mathcal{A}$.*

We give some examples of languages and structures.

Examples 3.3. *1. The language of groups is $\mathcal{L}_{gp} := \langle \cdot, (-)^{-1}, e\rangle$ where $\cdot$ is a binary function symbol, $(-)^{-1}$ is a unary function symbol and e is a constant symbol. Any group G gives rise to an $\mathcal{L}$-structure $\mathcal{A}$ by interpreting $\cdot$ as the group multiplication, $(-)^{-1}$ as the function which takes an element of the group to its inverse and e as the identity element. However, not all $\mathcal{L}_{gp}$-structures are groups. For instance, let $\mathcal{A}$ be the $\mathcal{L}$-structure with domain $\mathbb{N}$, $\cdot^{\mathcal{A}}$ defined by $a \cdot^{\mathcal{A}} b := a+b+4$, $((-)^{-1})^{\mathcal{A}}$ interpreted as the identity function and $e^{\mathcal{A}} := 7$.*

2. *The language of rings is $\mathcal{L}_{ring} := \langle +, \cdot, -, 0, 1\rangle$ where $+$ and $\cdot$ are binary function symbols, and 0 and 1 are constant symbols. Any ring gives rise to an $\mathcal{L}_{ring}$-structure. Again, not all $\mathcal{L}_{ring}$-structures are rings.*

3. *The language of graphs is $\mathcal{L}_{graph} := \langle E\rangle$ where E is a binary relation symbol. In this class when we say graph we mean a (possibly infinite) simple graph - this means I don't allow double edges or loops but I do allow infinitely many vertices (unlike in your Graph Theory and combinatorics course). A graph becomes an $\mathcal{L}_{graph}$-structure $\mathcal{A}$ by taking the domain to be the set of vertices V and letting $E^{\mathcal{A}}$ be the set*

$$\{(a,b) \in V^2 \mid \text{ there is an edge between } a \text{ and } b\}.$$

Of course there are $\mathcal{L}_{graph}$-structures which are not graphs. Exercise in class: write one down.

3.2 Formulas

Definition 3.4. *The **alphabet** of a language $\mathcal{L}$ is the relation, functions and constant symbols of $\mathcal{L}$ together with a set of logical symbols which are part of every language consisting of:*

1. *Connectives: $\{\to, \perp\}$*
2. *A quantifier: $\forall$*
3. *The equality symbols $=$*
4. *Brackets $)$ and $($*
5. *Comma: ,*
6. *A set of variables denoted $Vbl := \{v_i \mid i \in \mathbb{N}\}$*

As in propositional logic, words in an alphabet are just finite tuples from that alphabet and again, as in propositional logic, we write $a_1 a_2 \ldots a_n$ rather than $(a_1, \ldots, a_n)$.

Terms and the functions they define

We define the set of terms of $\mathcal{L}$ inductively.

Definition 3.5. *Let $\mathcal{L}$ be a language. Define $tm_0(\mathcal{L})$ to be the set $Vbl \cup \mathcal{C}$. For all $k \in \mathbb{N}$, let*

$$tm_{k+1}(\mathcal{L}) := tm_k(\mathcal{L}) \cup \{F(t_1, t_2, \ldots, t_n) \mid F \in \mathcal{F}, n \text{ is the arity of } F \text{ and } t_1, \ldots, t_n \in tm_k(\mathcal{L})\}.$$

We define the set of $\mathcal{L}$-terms to be

$$tm(\mathcal{L}) := \bigcup_{k \in \mathbb{N}_0} tm_k(\mathcal{L}).$$

The complexity $cmpx(t)$ of an $\mathcal{L}$-term t is the least k such that $t \in tm_k(\mathcal{L})$.

When a language $\mathcal{L}$ has no function symbols the set of $\mathcal{L}$-terms is just $\mathrm{Vbl} \cup \mathcal{C}$.

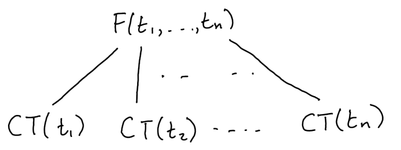
Examples 3.6.
1. *Let $\mathcal{L}$ be the empty language i.e. the language with no relation, function or constant symbols. Then the $\mathcal{L}$-terms are exactly the set of variables.*
2. *The set of $\mathcal{L}_{graph}$-terms is exactly the set of variables.*

Every variable which occurs in a term t is called a **free variable** of t and the set of free variables of t is denoted $\mathrm{Fr}(t)$. We call those terms without free variables **constant terms**.

As we did for propositional formulas, we can draw construction trees for terms. We define them inductively.

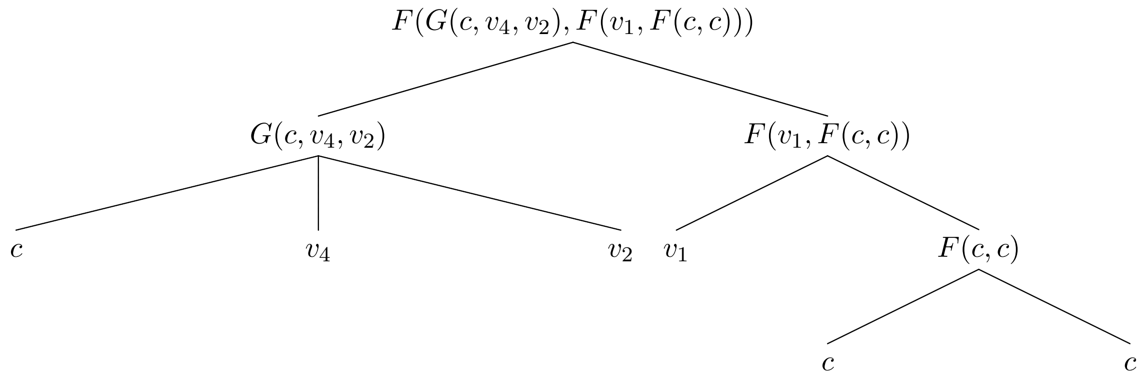
(I) If t is a constant symbol or a variable then the construction tree $\mathsf{CT}(t)$ of t is just t.

(II) If t is of the form $F(t_1, t_2, \ldots, t_n)$ for some function symbol of arity n and terms $t_1, \ldots t_n$ then the construction tree of t is

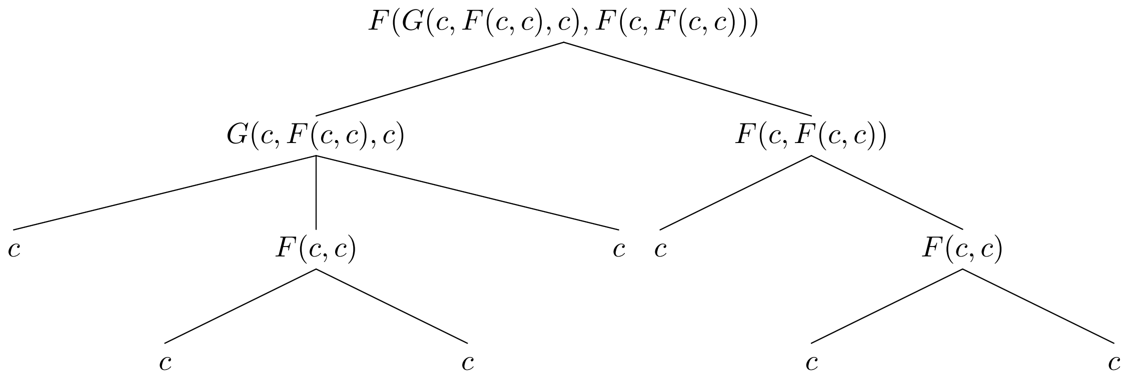


Exactly as for propositional formulas, we can draw a construction tree for a term to show that it is indeed a term.

Example 3.7. *Let $\mathcal{L} = \langle F, G, c\rangle$ where F is a function symbol of arity 2, G is a function symbol of arity 3 and c is a constant symbol. Then $F(G(c, v_4, v_2), F(v_1, F(c, c)))$ is and $\mathcal{L}$-term. Here is its construction tree:*



Its complexity is 3.
The $\mathcal{L}$*-term* $F(G(c,F(c,c),c),F(c,F(c,c)))$ *is a constant term. Here is its construction tree:*



In many algebraic subjects, for instance Group Theory, we use "infix" notation to denote the group operation applied to elements of the group. For instance, in $\mathbb{Z}$ as a group under $+$ we write $5+4$ rather than $+(5,4)$. We will do this for function symbols in some situations (mainly when our language is meant to be the language for an algebraic structure where this is the common notation) to make our terms more easily readable.

Example 3.8. *In this language of rings* $\mathcal{L}_{ring}$*,* $+(v_1,\cdot(v_1,v_2))$ *is an* $\mathcal{L}_{ring}$*-term of complexity* 2*. In infix notation this is written* $v_1+(v_1\cdot v_2)$*. The expression* $\cdot(v_2,v_2)$ *is an* $\mathcal{L}_{ring}$*-term. In infix notation it's written* $v_2\cdot v_2$*. We might even write it* v_2^2 *sometimes.*

Informally, this theorem says that every term has a unique construction tree. It's proof is similar in flavour to the Unique Construction Theorem for propositional formulas. We won't cover it.

Theorem 3.9 (Unique Construction Theorem for Terms)**.** *If* t *is an* $\mathcal{L}$*-term then exactly one of the following is true:*

1. t *is a variable.*

2. t *is a constant symbol.*

3. t *is of the form* $F(t_1,\dots,t_n)$ *for a unique function symbol* F *of arity* n *and a unique* n*-tuple of terms* $(t_1,\dots,t_n)$*.*

Corollary 3.10. *For* $n\in\mathbb{N}$*,* F *a function symbol of* $\mathcal{L}$ *of arity* n *and* $t_1,\dots,t_n\in tm(\mathcal{L})$*,*

$$\textit{cmpx}(F(t_1,\dots,t_n)) = 1+\max\{\textit{cmpx}(t_1),\dots,\textit{cmpx}(t_n)\}.$$

We are now ready to discuss the functions terms define. We start with some informal examples.

1. Let $\mathcal{L}$ be a language with a binary function symbol F and a unary function symbol H. Let $\mathcal{A}$ be the $\mathcal{L}$-structure with underlying set $\mathbb{R}$, $F^{\mathcal{A}}(r,s)=r+7$ and $H^{\mathcal{A}}(r)=r^2+1$. Let t be the $\mathcal{L}$-term $F(v_1,H(v_1))$. Then t defines the function from $\mathbb{R}^2$ to $\mathbb{R}$ which maps (a,b) to $a+7$.

2. Let t be the $\mathcal{L}_{ring}$-term $+(v_1,\cdot(v_2,+(v_1,v_2)))$, written in infix notation as $v_1+(v_2\cdot(v_1+v_2))$. If we view $\mathbb{Z}$ as an $\mathcal{L}_{ring}$-structure in the usual way then this term defines the function from $\mathbb{Z}^2$ to $\mathbb{Z}$ which maps $(a,b)\in\mathbb{Z}^2$ to $a+ab+b^2$.

In some books you will see $\mathcal{L}$-terms t having interpretations in $\mathcal{L}$-structures $\mathcal{A}$ as functions $t^{\mathcal{A}} : A^n \to A$ where n is the size of a set of free variables containing the free variables of t. We choose to do something slightly different which, although slightly contrary to our natural intuition about how $\mathcal{L}$-terms should be interpreted, will result in each $\mathcal{L}$-term having just one interpretation rather than technically having many.

Before making our definition we compare this with the situation of polynomials in many variables. The polynomial $X_1^2 + 5X_3$ defines a function $\mathbb{R}^3 \to \mathbb{R}$. However, perhaps I had in mind that $X_1^2 + 5X_3$ was actually an element of the polynomial ring $\mathbb{R}[X_1, \ldots, X_4]$. In this case it defines a function $\mathbb{R}^4 \to \mathbb{R}$. So, one polynomial defines many functions. The next definition avoids this phenomena.

Definition 3.11. *Let $\mathcal{L}$ be a language and let $\mathcal{A}$ be an $\mathcal{L}$-structure with domain A.*

*(i) An **assignment** of an $\mathcal{L}$-structure $\mathcal{A}$ is a map*

$$h : Vbl \to A.$$

Given an assignment h of $\mathcal{A}$, a variable x and an element $a \in A$ we denote by $h\binom{x}{a}$ the assignment of $\mathcal{A}$ which differs from h only at the variable x, with value a at x:

$$h\binom{x}{a}(y) = \begin{cases} h(y) & \text{if } y \neq x \\ a & \text{if } y = x. \end{cases}$$

(ii) We define by induction on the complexity of an $\mathcal{L}$-term t an element $t^{\mathcal{A}}[h] \in A$ for each assignment h of $\mathcal{A}$ as follows:

(a) If $\mathit{cmpx}(t) = 0$, then

$$t^{\mathcal{A}}[h] = \begin{cases} t^{\mathcal{A}} & \text{if } t \in \mathcal{C} \\ h(t) & \text{if } t \in Vbl. \end{cases}$$

(b) If $t_1, \ldots, t_n$ are $\mathcal{L}$-terms and $F \in \mathcal{F}$ is of arity n, then we define

$$F(t_1, \ldots, t_n)^{\mathcal{A}}[h] := F^{\mathcal{A}}(t_1^{\mathcal{A}}[h], \ldots, t_n^{\mathcal{A}}[h]).$$

For each $\mathcal{L}$-term t, this definition gives a well-defined element $t^{\mathcal{A}}$ of A by 3.9.

For each $\mathcal{L}$-term t, we define a function

$$t^{\mathcal{A}} : A^{\infty} \to A,$$

where

$$A^{\infty} = \{\bar{a} \mid \bar{a} = (a_1, a_2, a_3, \ldots, a_l, \ldots) \text{ with } a_l \in A \text{ for all } l \in \mathbb{N}\},$$

that is A^{∞} is the set of all infinite sequences of elements of A. So here an assignment $h : Vbl \to A$ is denoted as a sequence $\bar{a} = (h(v_1), h(v_2), h(v_3), \ldots)$.

Remark 3.12. *You will show on your exercise sheet that for t an $\mathcal{L}$-term and $\mathcal{A}$ an $\mathcal{L}$-structure, $t^{\mathcal{A}}[h]$ only depends on the values of $h(v_i)$ for $v_i \in Fr(t)$.*

Formulas

Definition 3.13. *An **atomic $\mathcal{L}$-formula** is a word in the alphabet of $\mathcal{L}$ of the form*

$$t_1 = t_2$$

where t_1 and t_2 are $\mathcal{L}$-terms or of the form

$$R(t_1, \dots, t_n)$$

where $n \in \mathbb{N}$, $t_1, \dots, t_n$ are $\mathcal{L}$-terms and $R \in \mathcal{R}$ has arity n.

Definition 3.14. *We define the set of formulas $Fml(\mathcal{L})$ of a language by induction. Let $Fml_0(\mathcal{L})$ be the set of all atomic $\mathcal{L}$-formulas together with $\perp$. For each $k \in \mathbb{N}_0$, let*

$$Fml_{k+1}(\mathcal{L}) := Fml_k(\mathcal{L}) \cup \{(\phi \to \psi), \mid \phi, \psi \in Fml_k\} \cup \{(\forall x \phi) \mid \phi \in Fml_k\}.$$

*We define the set of $\mathcal{L}$-**formulas** to be*

$$Fml(\mathcal{L}) := \bigcup_{i \in \mathbb{N}_0} Fml_i(\mathcal{L}).$$

*The **complexity** of an $\mathcal{L}$-formula ϕ is the least $k \in \mathbb{N}_0$ such that $\phi \in Fml_k(\mathcal{L})$.*

Theorem 3.15 (Unique Construction for Formulas in Predicate Logic).
Let $\mathcal{L}$ be a language and let ϕ be an $\mathcal{L}$-formula. Then exactly one of the following is true:

(i) ϕ is $\perp$;

(ii) ϕ is atomic and there are uniquely determined $t_1, t_2 \in tm(\mathcal{L})$ such that ϕ is $t_1 = t_2$;

(iii) ϕ is atomic and there are a unique $n \in \mathbb{N}$, a unique relation symbol $R \in \mathcal{R}$ and uniquely determined $\mathcal{L}$-terms $t_1, \dots, t_n$ such that ϕ is $R(t_1, \dots, t_n)$;

(iv) ϕ is equal to $(\phi_1 \to \phi_2)$ for uniquely determined $\phi_1, \phi_2 \in Fml(\mathcal{L})$; or

(v) ϕ is equal to $(\forall x \psi)$ for uniquely determined $\psi \in Fml(\mathcal{L})$ and $x \in Vbl$.

Exercise. *Come up with a definition of the construction tree $CT(\phi)$ of a formula ϕ in a (first order) language $\mathcal{L}$ based on the inductive definition of an $\mathcal{L}$-formula. Use the following as your base case:*

(I) If ϕ is either an atomic formula or $\perp$ then the construction tree of ϕ is just ϕ.

This exercise will be on Exercise Sheet 6. For inspiration, you should look at the definition of the construction tree of a propositional formula and the definition of the construction tree of a term.

Definition 3.16. *We define the set of free-variables $Fr(\phi)$ of an $\mathcal{L}$-formula ϕ by induction.*

(0) $Fr(\perp) = \emptyset$.

(1) For all terms t_1, t_2, $\mathit{Fr}(t_1 = t_2) = \mathit{Fr}(t_1) \cup \mathit{Fr}(t_2)$ and for all terms $t_1, \ldots, t_n$ and relation symbols R, $\mathit{Fr}(R(t_1, \ldots, t_n)) = \bigcup_{i=1}^n \mathit{Fr}(t_i)$.

(2) For the inductive step, for ϕ, ψ $\mathcal{L}$-formulas and $x \in \mathit{Vbl}$, we define

$$\mathit{Fr}((\phi \to \psi)) := \mathit{Fr}(\phi) \cup \mathit{Fr}(\psi)$$

and

$$\mathit{Fr}((\forall x \phi)) := \mathit{Fr}(\phi) \backslash \{x\}.$$

Warning: There will be a more complicated definition relating to free variables when we study formal proofs.

End of L19 & L20

Abbreviations and making the language readable:

(i) In some parts of the notes, we will drop some brackets when it does not lead to ambiguity. In some places I will add brackets to increase readability.

(ii) As we did in the later part of Propositional Logic, we will view

- $(\neg \phi)$ for ϕ an $\mathcal{L}$-formula as an abbreviation for $(\phi \to \bot)$;
- $(\phi \vee \psi)$ for ϕ, ψ $\mathcal{L}$-formulas as an abbreviation for $((\neg \phi) \to \psi)$;
- $(\phi \wedge \psi)$ for ϕ, ψ $\mathcal{L}$-formulas as an abbreviation for $(\neg(\phi \to (\neg \psi)))$; and
- $(\phi \leftrightarrow \psi)$ for ϕ, ψ $\mathcal{L}$-formulas as an abbreviation for $((\phi \to \psi) \wedge (\psi \to \phi))$.

(iii) You may have expected there to be a second quantifier. We will write $(\exists x \phi)$ as an abbreviation for $(\neg(\forall x (\neg \phi)))$ when $x \in \mathrm{Vbl}$ and ϕ is an $\mathcal{L}$-formula.

(iv) If $t, s \in \mathrm{tm}(\mathcal{L})$ then we will write $t \neq s$ to mean $(\neg t = s)$.

(v) We will write

$$\forall x_1, \ldots, x_n\ \phi \text{ instead of } \forall x_1 \ldots \forall x_n \phi \ldots \text{ and } \exists x_1, \ldots, x_n\ \phi \text{ instead of } \exists x_1 \ldots \exists x_n \phi$$

where each x_i is a variable and ϕ is a formula. Words in $\mathcal{L}$ of the form $\forall x$ and $\exists x$ where $x \in \mathrm{Vbl}$ are called **quantifiers**.

3.3 Tarski's Truth Definition

Definition 3.17 (Tarski's Truth Definition). *Let $\mathcal{L}$ be a (first order) language and $\mathcal{M}$ an $\mathcal{L}$-structure with domain M. We define by induction on the complexity of an $\mathcal{L}$-formula, the expression ϕ **is true in** $\mathcal{M}$ **at** h, denoted*

$$\mathcal{M} \models \phi[h],$$

for each assignment $h : \mathit{Vbl} \to M$ as follows:

(0) $\mathcal{M} \not\models \bot[h]$

(1) If ϕ is of the form $t_1 = t_2$ with $\mathcal{L}$-terms t_1, t_2 then

$$\mathcal{M} \models t_1 = t_2\ [h] \iff t_1^{\mathcal{M}}[h] = t_2^{\mathcal{M}}[h].$$

(2) If ϕ is of the form $R(t_1, \dots, t_n)$ with $R \in \mathcal{R}$ of arity n and $t_1, \dots, t_n \in tm(\mathcal{L})$ then

$$\mathcal{M} \models R(t_1, \dots, t_n)\ [h] \iff (t_1^{\mathcal{M}}[h], \dots, t_n^{\mathcal{M}}[h]) \in R^{\mathcal{M}}.$$

(3) For the induction step we take $\phi, \psi \in Fml(\mathcal{L})$, $x \in Vbl$ and define

- *$\mathcal{M} \models (\phi \to \psi)[h]$ if $\mathcal{M} \models \phi[h]$ implies $\mathcal{M} \models \psi[h]$,*

and

- *$\mathcal{M} \models (\forall x \phi)[h]$ if for all $a \in M$ we have $\mathcal{M} \models \phi[h\binom{x}{a}]$.*

You will sometimes, perhaps even in these notes, see ϕ **holds in** $\mathcal{M}$ **at** h or $\mathcal{M}$ **satisfies** ϕ **at** h. Both these phrases mean ϕ is true in $\mathcal{M}$ at h.

Proposition 3.18 (Derived rules for truth). *Let $\mathcal{L}$ be a (first order) language, let $\mathcal{M}$ be an $\mathcal{L}$-structure with domain M and let $h : Vbl \to M$ be an assignment.*

1. *For all $\mathcal{L}$-formulas ϕ,*
$$M \models (\neg\phi)[h] \text{ if and only if } \mathcal{M} \not\models \phi[h].$$
2. *For all $\mathcal{L}$-formulas ϕ, ψ,*
$$\mathcal{M} \models (\phi \wedge \psi)[h] \text{ if and only if } M \models \phi[h] \text{ and } M \models \psi[h]$$
and
$$\mathcal{M} \models (\phi \vee \psi)[h] \text{ if and only if } M \models \phi[h] \text{ or } M \models \psi[h]$$
3. *For all $\mathcal{L}$-formulas ϕ and $x \in Vbl$, $\mathcal{M} \models (\exists x \phi)[h]$ if and only if there exists $a \in M$ such that $\mathcal{M} \models \phi[h\binom{x}{a}]$.*

Proof. 1. and 2. will be on the next exercise sheet. We will prove 3. assuming 1. Recall that $(\exists x \phi)$ is an abbreviation for $(\neg(\forall x(\neg\phi)))$. Therefore, by 1., $\mathcal{M} \models (\exists x\phi)[h]$ if and only if $\mathcal{M} \models (\forall x(\neg\phi))[h]$ does not hold. Now $\mathcal{M} \models (\forall x(\neg\phi))[h]$ if and only if for all $a \in M$, $\mathcal{M} \models (\neg\phi)[h\binom{x}{a}]$. So $\mathcal{M} \models (\forall x(\neg\phi))[h]$ if and only if for all $a \in M$, $\mathcal{M} \models \phi[h\binom{x}{a}]$ does not hold. So $\mathcal{M} \models (\forall x(\neg\phi))[h]$ if and only if there exists an $a \in M$, such that $\mathcal{M} \models \phi[h\binom{x}{a}]$ as required. □

We work through an example of how Tarski's Truth Definition works.

Examples 3.19. *(i) Let $\mathcal{L} := \langle F, c, d\rangle$ where F is a unary function symbol and both c and d are constant symbols. Let χ be the $\mathcal{L}$-formula*

$$(\forall v_1(F(v_1) = c \to F(v_2) = d)).$$

Let $\mathcal{M}$ be the $\mathcal{L}$-structure with domain $\mathbb{R}$, $F^{\mathcal{M}} : \mathbb{R} \to \mathbb{R}$ defined by $F^{\mathcal{M}}(r) = r^2 + 1$, $c^{\mathcal{M}} := 1$ and $d^{\mathcal{M}} := 8$. We will check for which assignments $h : Vbl \to \mathbb{R}$, we have that ϕ is true in $\mathcal{M}$ at h.

Step 1: *By Tarski's Truth Definition, $\mathcal{M} \models \chi[h]$ if and only if for all $a \in \mathbb{R}$, $\mathcal{M} \models (F(v_1) = c \to F(v_2) = d)[h\binom{v_1}{a}]$.*

Step 2: *By Tarski's Truth Definition, $\mathcal{M} \models (F(v_1) = c \to F(v_2) = d)[h\binom{v_1}{a}]$ if and only if $\mathcal{M} \models (F(v_1) = c)[h\binom{v_1}{a}]$ implies $\mathcal{M} \models (F(v_2) = d)[h\binom{v_1}{a}]$.*

Step 3a: *$\mathcal{M} \models (F(v_1) = c)[h\binom{v_1}{a}]$ if and only if $F^{\mathcal{M}}(h\binom{v_1}{a}(v_1)) = c^{\mathcal{M}}$.*
Now $h\binom{v_1}{a}(v_1) = a$ by definition and $c^{\mathcal{M}} = 1$. So $\mathcal{M} \models (F(v_1) = c)[h\binom{v_1}{a}]$ if and only if $a^2 + 1 = 1$.

Step 3b: *$\mathcal{M} \models (F(v_2) = d)[h\binom{v_1}{a}]$ if and only if $F^{\mathcal{M}}(h\binom{v_1}{a}(v_2)) = d^{\mathcal{M}}$.*
Now $h\binom{v_1}{a}(v_2) = h(v_2)$ and $d^{\mathcal{M}} = 8$. Therefore $F^{\mathcal{M}}(h\binom{v_1}{a}(v_2)) = d^{\mathcal{M}}$ if and only if $h(v_2)^2 + 1 = 8$.

Step 4: *$\mathcal{M} \models \chi[h]$ if and only if for all $a \in \mathbb{R}$, $a^2 = 0$ implies $h(v_2)^2 = 7$.*

Conclusion: *$\mathcal{M} \models \chi[h]$ if and only if $h(v_2) = \pm\sqrt{7}$.*

(ii) Let $\mathcal{L} := \langle R \rangle$ where R is a binary relation symbol. Let χ be the formula

$$(\forall v_1(\forall v_2(R(v_1, v_2) \to R(v_2, v_1)))).$$

Let $\mathcal{M}$ be an $\mathcal{L}$-structure with domain M and let $h : Vbl \to M$ be an assignment. We consider when $\mathcal{M} \models \chi[h]$.

Step 1: *$\mathcal{M} \models \chi[h]$ if and only if for all $a \in M$,*

$$\mathcal{M} \models (\forall v_2(R(v_1, v_2) \to R(v_2, v_1)))[h\binom{v_1}{a}].$$

Step 2: *$\mathcal{M} \models \chi[h]$ if and only if for all $a \in M$ and $b \in M$,*

$$\mathcal{M} \models (R(v_1, v_2) \to R(v_2, v_1))[h\binom{v_1}{a}\binom{v_2}{b}].$$

Step 3: *$\mathcal{M} \models \chi[h]$ if and only if for all $a \in M$ and $b \in M$,*

$$(a, b) \in R^{\mathcal{M}} \text{ implies } (b, a) \in R^{\mathcal{M}}.$$

Conclusion: *$\mathcal{M} \models \chi[h]$ if and only if $R^{\mathcal{M}}$ is a symmetric relation.*

We now state some results which link truth in propositional logic with truth in predicate logic. End of L21 & L22

Definition 3.20. *Let $\mathcal{L}$ be a first order language. Let t be a propositional formula in a propositional language with propositional variables $p_1, \ldots, p_n$. Let $\phi_1, \ldots, \phi_n$ be $\mathcal{L}$-formulas. We define $t(p_1/\phi_1, \ldots, p_n/\phi_n)$ to be the $\mathcal{L}$-formula obtained by replacing each instance of p_i by ϕ_i for $1 \leq i \leq n$.*

Remark 3.21. *We should really prove that $t(p_1/\phi_1, \ldots, p_n/\phi_n)$ in the above definition actually is an $\mathcal{L}$-formula. If you want to carry this out then prove it by induction on the complexity of t.*

Proposition 3.22. *[Propositional tautologies are true in all structures at all assignments] Let $\mathcal{L}$ be a (first order) language. If t is a propositional tautology in a propositional language with propositional variables $p_1, \dots, p_n$ and $\phi_1, \dots, \phi_n$ are $\mathcal{L}$-formulas then $\mathcal{M} \models t(p_1/\phi_1, \dots, p_n/\phi_n)[h]$ for all $\mathcal{L}$-structures $\mathcal{M}$ and assignments h.*

This lemma is non-examinable. It implies the previous proposition and its proof if fun.

Lemma 3.23. *Let $\mathcal{L}$ be a (first order) language, $\phi_1, \dots, \phi_n$ $\mathcal{L}$-formulas, $\mathcal{M}$ an $\mathcal{L}$-structure and h an assignment of $\mathcal{M}$. Define a function v_h from the set of propositional formulas in the propositional language with propositional variables $p_1, \dots, p_n$ to the set $\{\mathbb{T}, \mathbb{F}\}$ by setting $v_h(t) = \mathbb{T}$ if and only if $\mathcal{M} \models t(p_1/\phi_1, \dots, p_n/\phi_n)[h]$. Then v_h is a valuation.*

Proof. The proof will be a non-compulsory exercise. □

Proof of Proposition 3.22. Let v_h be as in the Lemma 3.23. If t is a tautology then $v_h(t) = \mathbb{T}$. Thus $\mathcal{M} \models t(p_1/\phi_1, \dots, p_n/\phi_n)[h]$ as required. □

Free variables and the scope of a quantifier

We have a defined in 3.16 the set of free variables of a formula in a way that is best for using in most proofs. It is not the easiest definition to use in order to calculate the set of free variables or to understand what a free variable is.

Let $\phi := a_1 a_2 \dots a_n$ be a formula. A **subformula** of ϕ is a word $a_i a_{i+1} \dots a_k$ such that $i \leq k$ which is also a formula. One can prove that the subformulas of a formula are exactly those formulas which appear in its construction tree.

Definition 3.24.

*(i) An instance of a variable x in a formula ϕ is a **quantifier instance** if it appears directly after $\forall$ or $\exists$.*

*(ii) An instance of a variable x in a formula ϕ is **bound** if there is a subformula of ϕ of the form $(\forall x \psi)$ (or $(\exists x \psi)$) and this instance of x occurs in ψ. We call ψ the **scope** of the quantifier $\forall$ (respectively $\exists$).*

*(iii) Any instance of a variable x which is not a quantifier instance or a bound instance is a **free** instance.*

This exercise will be done during the class.

Exercise 3.25. *Let $\mathcal{L}_{ring} := \langle 0, 1, +, \cdot, - \rangle$ be the language of rings. Label all the quantifier, free and bound instances of the variable v_1 in the following $\mathcal{L}_{ring}$-formula*

$$(((\forall v_1 (\exists v_2 (v_1 \cdot v_2 = 1))) \vee v_1 \cdot v_2 = 0) \to (\forall v_2\, v_1 + v_2 = 0))$$

Proposition 3.26. *Let ϕ be a formula. The set of variables which occur as free instances in ϕ is equal to the set of free variables of ϕ.*

Proof. Non-examinable and not lectured. □

Notation 3.27. *For ϕ a formula, we will write $\phi(x_1, \dots, x_n)$ where $x_1, \dots, x_n$ are variables to indicate that the set of free variables of ϕ is contained in $\{x_1, \dots, x_n\}$.*

Definition 3.28. *An $\mathcal{L}$-**sentence** is a formula with no free variables i.e. ϕ is a sentence if $Fr(\phi) = \emptyset$.*

Back to Tarski's Truth Definition

Proposition 3.29. *Let $\mathcal{L}$ be a language, $\mathcal{M}$ be an $\mathcal{L}$-structure with domain M and ϕ an $\mathcal{L}$-formula. If $h, g : Vbl \to M$ are assignments such that $h(x) = g(x)$ for all $x \in Fr(\phi)$ then $\mathcal{M} \models \phi[h]$ if and only if $\mathcal{M} \models \phi[g]$.*

Proof. We won't include all details in some parts of the proof. On Exercise Sheet 6 you will show that for $t \in \mathrm{tm}(\mathcal{L})$, $t^{\mathcal{M}}[h] = t^{\mathcal{M}}[g]$ if g and h agree on the free variables of t. This implies that if ϕ is an atomic formula then $\mathcal{M} \models \phi[h]$ if and only if $\mathcal{M} \models \phi[g]$. Note that $\perp$ has no free variables and $\mathcal{M} \not\models \perp[f]$ for all assignments f. So the statement is true for all formulas of complexity 0. We now proceed by induction on complexity of formulas.

Suppose the statement is true for all formulas $\psi \in \mathrm{Fml}_k(\mathcal{L})$. Take $\phi \in \mathrm{Fml}_{k+1}(\mathcal{L}) \backslash \mathrm{Fml}_k(\mathcal{L})$. Then either ϕ is $(\psi_1 \to \psi_2)$ for some $\psi_1, \psi_2 \in \mathrm{Fml}_k(\mathcal{L})$ or ϕ is $(\forall y \psi)$ for some $\psi \in \mathrm{Fml}_k(\mathcal{L})$. I leave it to you to write a proof for the case ϕ is $(\psi_1 \to \psi_2)$.

Suppose ϕ is $(\forall y \psi)$. Then $\mathrm{Fr}(\phi) = \mathrm{Fr}(\psi) \backslash \{y\}$. Suppose $h(x) = g(x)$ for all $x \in \mathrm{Fr}(\phi)$. Then $\mathcal{M} \models \phi[h]$ if and only if for all $a \in M$, $\mathcal{M} \models \psi[h(\tbinom{y}{a})]$ and $\mathcal{M} \models \phi[g]$ if and only if for all $a \in M$, $\mathcal{M} \models \psi[g(\tbinom{y}{a})]$.

By assumption, $h(x) = g(x)$ for all $x \in \mathrm{Fr}(\phi)$. Therefore $h(\tbinom{y}{a})(x) = g(\tbinom{y}{a})(x)$ for all $x \in \mathrm{Fr}(\phi) \cup \{y\} = \mathrm{Fr}(\psi)$. Therefore, by our induction hypothesis, for all $a \in M$, $\mathcal{M} \models \psi[h(\tbinom{y}{a})]$ if and only if $\mathcal{M} \models \psi[g(\tbinom{y}{a})]$. Therefore $\mathcal{M} \models \phi[h]$ if and only if $\mathcal{M} \models \phi[g]$ as required. Hence, by induction on the complexity of formulas, the statement of the proposition is true. $\square$

Definition 3.30.

(i) *Let $\phi(x_1, \ldots, x_n)$ be an $\mathcal{L}$-formula and $\mathcal{M}$ an $\mathcal{L}$-structure with domain M. For $a_1, \ldots, a_n \in M$, we write*

$$\mathcal{M} \models \phi[a_1, \ldots, a_n]$$

to mean that

$$\mathcal{M} \models \phi[h]$$

for some/every assignment $h : Vbl \to M$ such that $h(x_i) = a_i$ for $1 \leq i \leq n$.

(ii) *Let ϕ be an $\mathcal{L}$-sentence and $\mathcal{M}$ an $\mathcal{L}$-structure. We write $\mathcal{M} \models \phi$ to mean that $\mathcal{M} \models \phi[h]$ for some/every valuation h. In this case we say $\mathcal{M}$ **satisfies** ϕ or ϕ **is true in** $\mathcal{M}$ or ϕ **holds in** $\mathcal{M}$. We will sometimes also say $\mathcal{M}$ **is a model of** ϕ or $\mathcal{M}$ **models** ϕ.*

(iii) *Let Σ be a set of $\mathcal{L}$-sentences and $\mathcal{M}$ is an $\mathcal{L}$-structure. Then we write $\mathcal{M} \models \Sigma$ to mean that for all $\phi \in \Sigma$, $\mathcal{M} \models \phi$. In this situation we say $\mathcal{M}$ **is a model of** Σ or $\mathcal{M}$ **satisfies** Σ.*

We are finally ready to define logical implication and logical equivalence for $\mathcal{L}$-formulas. For arbitrary formulas, the valuations of propositional logic are replaced with $\mathcal{L}$-structures together with assignments. For (sets of) sentences, valuations are replaced by $\mathcal{L}$-structures.

Definition 3.31.

(i) *Let $\Sigma \subseteq Fml(\mathcal{L})$ and $\phi \in Fml(\mathcal{L})$. We say Σ **logically implies** ϕ and write $\Sigma \models \phi$ if for all $\mathcal{L}$-structures $\mathcal{M}$ and all assignments h, if $\mathcal{M} \models \psi[h]$ for all $\psi \in \Sigma$ then $\mathcal{M} \models \phi[h]$.*

As in propositional logic, we will write $\models \phi$ to mean that $\emptyset \models \phi$ and $\psi_1, \ldots, \psi_n \models \phi$ to mean $\{\psi_1, \ldots, \psi_n\} \models \phi$.

(ii) *We say that $\mathcal{L}$-formulas ϕ and ψ are **logically equivalent** if $\phi \models \psi$ and $\psi \models \phi$. In this situation we write $\phi \equiv \psi$.*

Proposition 3.22 can now be rewritten as follows.

End of L23 & L24

Proposition 3.32. *Let $\mathcal{L}$ be a (first order) language. If t is a propositional tautology in a propositional language with propositional variables $p_1, \dots, p_n$ and $\phi_1, \dots, \phi_n$ are $\mathcal{L}$-formulas then $\models t(p_1/\phi_1, \dots, p_n/\phi_n)$. In particular, for all $\mathcal{L}$-formulas ϕ, ψ and σ,*

(i) $\models (\phi \to (\psi \to \phi))$

(ii) $\models ((\phi \to (\psi \to \sigma)) \to ((\phi \to \psi) \to (\phi \to \sigma)))$

(iii) $\models ((\neg(\neg\phi)) \to \phi)$

Example 3.33. *For all sets of $\mathcal{L}$-formulas Σ, $\Sigma \models \bot$ means there is no $\mathcal{L}$-structure $\mathcal{M}$ and assignment h such that $\mathcal{M} \models \phi[h]$ for all $\phi \in \Sigma$.*

Remark 3.34. *Let ϕ, ψ be $\mathcal{L}$-formulas and Σ a set of $\mathcal{L}$-formulas. If $\Sigma \models \phi$ and $\Sigma \models (\phi \to \psi)$ then $\Sigma \models \psi$.*

Proof. Easy exercise! $\square$

So far everything we have said about logical implication is the same as for Propositional Logic. Here are some examples and non-examples involving quantifiers.

Examples 3.35. *Let ϕ, ψ be $\mathcal{L}$-formulas and $x \in Vbl$.*

1. *If $x \notin Fr(\phi)$ then $\phi \equiv \forall x\, \phi \equiv \exists x\, \phi$.*
2. *$\forall x(\phi \wedge \psi) \equiv (\forall x\phi) \wedge (\forall x\psi)$.*
3. *$(\forall x\phi) \vee (\forall x\psi)$ logically implies $\forall x(\phi \vee \psi)$.*
4. *$\forall x(\phi \vee \psi)$ does not logically imply $(\forall x\phi) \vee (\forall x\psi)$.*

Proof. 1. Let $\mathcal{M}$ be an $\mathcal{L}$-structure with domain M and let $h : \mathrm{Vbl} \to M$ be an assignment. Then $\mathcal{M} \models \forall x\phi[h]$ if and only if for all $a \in \mathcal{M}$, $\mathcal{M} \models \phi[h\binom{x}{a}]$. Since $x \notin \mathrm{Fr}(\phi)$, for all $y \in \mathrm{Fr}(\phi)$, $h(y) = h\binom{x}{a}(y)$. Therefore, by 3.29, $\mathcal{M} \models \phi[h]$ if and only if $\mathcal{M} \models \phi[h\binom{x}{a}]$. Thus $\mathcal{M} \models \phi[h]$ if and only if $\mathcal{M} \models \forall x\phi$. The proof for $\exists$ is similar.

4. Let $\mathcal{L} := \langle c \rangle$ where c is a constant symbol. Let ϕ be $v_1 = c$ and let ψ be $v_1 \neq c$. Let $\mathcal{M}$ be an $\mathcal{L}$-structure with domain $\{0, 1\}$ and $c^{\mathcal{M}} := 0$. Let $h : \mathrm{Vbl} \to \{0, 1\}$ be the assignment defined by $h(v_i) = 0$ if $i = 1$ and $h(v_i) = 1$ otherwise. Then $\mathcal{M} \models \forall v_1(\phi \vee \psi)[h]$ but $\mathcal{M} \not\models (\forall v_1\phi) \vee (\forall v_1\psi)$.

I leave 2. and 3. to the reader. $\square$

Proposition 3.36. *Let ϕ, ψ be $\mathcal{L}$-formulas and let $x, x_1, \dots, x_n \in Vbl$.*

1. *For all $x \notin Fr(\phi)$, $\models \phi \to \forall x\phi$.*
2. *If $\models \phi$ then $\models \forall x_1 \dots \forall x_n\phi$.*
3. *$\models (\forall x(\phi \to \psi)) \to ((\forall x\phi) \to (\forall x\psi))$*

Proof. 1. follows from 3.35 and 3.34. I leave 3. as an exercise to the reader.

2. It's enough to show that if $\models \phi$ then $\models \forall x\phi$. Suppose $\models \phi$. Let $\mathcal{M}$ be an $\mathcal{L}$-structure with domain M and let h be an assignment of $\mathcal{M}$. Then $\mathcal{M} \models (\forall x\phi)[h]$ if and only if for all $a \in M$, $\mathcal{M} \models \phi[h(\binom{x}{a})]$. But this is true because $\models \phi$. $\square$

The proof of this remark will be on Exercise sheet 7.

Remark 3.37. *If $x \in Fr(\phi)$ then $\models \phi \to \forall x\phi$ is not always true.*

Substituting terms for variables

The words free and simultaneously are important in the next definition.

Definition 3.38. *Let $\phi \in Fml(\mathcal{L})$, let $t, t_1, \ldots, t_n \in tm(\mathcal{L})$ and let $x_1, \ldots, x_n$ be distinct variables.*

(i) The expression $t(x_1/t_1, \ldots, x_n/t_n)$ denotes the word in (the alphabet of) $\mathcal{L}$ obtained by replacing every instance of x_i in t by t_i for $1 \leq i \leq n$ simultaneously.

(ii) The expression $\phi(x_1/t_1, \ldots, x_n/t_n)$ denotes the word in (the alphabet of) $\mathcal{L}$ obtained by replacing every <u>free</u> instance of x_i in ϕ by t_i for $1 \leq i \leq n$ simultaneously.

Remark 3.39. *It can be shown by induction on the complexity of terms and induction on the complexity of formulas that $t(x_1/t_1, \ldots, x_n/t_n)$ is an $\mathcal{L}$-term and that $\phi(x_1/t_1, \ldots, x_n/t_n)$ is an $\mathcal{L}$-formula.*

Example 3.40. *Let $\mathcal{L} := \langle c \rangle$ where c is a constant symbol. Let ϕ be the $\mathcal{L}$-formula*

$$(((\forall v_1(v_1 = c \vee v_3 = c)) \wedge v_1 = v_1) \wedge v_2 = v_1).$$

Let t_1 be v_2 and let t_2 be c. Then $\phi(v_1/t_1, v_2/t_2)$ is

$$(((\forall v_1(v_1 = c \vee v_3 = c)) \wedge v_2 = v_2) \wedge c = v_2).$$

Definition 3.41.

*(1) A variable y is **substitutable for the variable x in a formula ϕ** if the variable x does not occur as a free instance in the scope of a quantifier $\forall y$ (or $\exists y$) in ϕ.*

*(2) A term t is **substitutable for the variable x in a formula ϕ** if all $y \in Fr(t)$ are substitutable for x in ϕ.*

Remark. *For any formula ϕ and variable x, x is substitutable for x in ϕ.*

Examples 3.42.

(i) The variable v_2 is substitutable for v_1 in the formula $(v_1 = v_2 \wedge (\forall v_1 \exists v_2\, v_1 = v_2))$.

(ii) The variable v_1 is not substitutable for v_2 in the formula $\exists v_1\, v_1 \neq v_2$.

(iii) The term $F(v_2, v_3)$ where F is a binary function symbol is substitutable for v_1 in the formula $(v_1 = v_2 \wedge (\forall v_1 \exists v_2\, v_1 = v_2))$.

(iv) The term $F(v_1, v_2)$ is not substitutable for v_2 in the formula $\exists v_1\, v_1 \neq v_2$.

Proposition 3.43. *Let ϕ be an $\mathcal{L}$-formula, x a variable and t an $\mathcal{L}$-term. If t is substitutable for x in ϕ then*

$$\models (\forall x\, \phi) \to \phi(x/t).$$

Warning: The condition "t is substitutable for x in ϕ" is necessary. For example, if ϕ is $(\exists v_1\, v_1 \neq v_2)$ and t is v_1 then $\not\models (\forall v_2\, \phi) \to \phi(v_2/v_1)$. Checking the details is on Exercise Sheet 7. End of L25 & L26

In order to prove this proposition we need to make some observations and prove a lemma.

Remark 3.44. *Let $\phi, \psi, \psi_1, \psi_2$ be formulas, t a term and x, y, z variables.*

(I) Suppose ϕ is $(\psi_1 \to \psi_2)$. Then t is substitutable for x in ϕ if and only if t is substitutable for x in ψ_1 and in ψ_2.

(II) Suppose ϕ is $(\forall z\, \psi)$.

(a) Then y is substitutable for x in ϕ if and only if
- *$x \notin Fr(\phi)$; or*
- *$z \neq y$ and y is substitutable for x in ψ.*

(b) Then t is substitutable for x in ϕ if and only if
- *$x \notin Fr(\phi)$; or*
- *$z \notin Fr(t)$ and t is substitutable for x in ψ.*

Proof. (II) is the tricky statement and (II)(b) is an easy consequence of (II)(a). So we prove (II)(a).

We first prove the forward direction. Suppose y is substitutable for x in ϕ and $x \in \mathrm{Fr}(\phi)$. Then $z \neq x$ and $x \in \mathrm{Fr}(\psi)$. Since y is substitutable for x in ϕ and $x \in \mathrm{Fr}(\psi)$, $y \neq z$. Suppose y is not substitutable for x in ψ. Then a free instance of x occurs in the scope of $\forall y$ in ψ. Since $z \neq x$, that instance stays free in ϕ.

We now prove the reverse direction. If $x \notin \mathrm{Fr}(\phi)$ then trivially y is substitutable for x. So suppose $z \neq y$ and y is substitutable for x in ψ. Then no free instance of x occurs in the scope of a quantifier $\forall y$ in ψ and since $z \neq y$ this is also true for ϕ. Therefore y is substitutable for x in ϕ as required. □

Lemma 3.45. *Let ϕ be a formula, x a variable and t a term. If t is substitutable for x then for all $\mathcal{L}$-structures $\mathcal{M}$ and assignments h of $\mathcal{M}$,*

$$\mathcal{M} \models \phi(x/t)[h] \text{ if and only if } \mathcal{M} \models \phi[h\begin{pmatrix} x \\ t^{\mathcal{M}}[h]\end{pmatrix}].$$

Proof. The case when ϕ is an atomic formula is on exercise sheet 7.

Suppose the statement is true for all $\psi \in \mathrm{Fml}_k(\mathcal{L})$. Take $\phi \in \mathrm{Fml}_{k+1}(\mathcal{L})$. Then ϕ is either of the form $(\psi_1 \to \psi_2)$ for $\psi_1, \psi_2 \in \mathrm{Fml}_k(\mathcal{L})$ or ϕ is $\forall z \psi$ for $\psi \in \mathrm{Fml}_k(\mathcal{L})$ and $z \in \mathrm{Vbl}$. I leave the case where ϕ is $(\psi_1 \to \psi_2)$ to the reader.

Suppose ϕ is $\forall z\, \psi$ for $\psi \in \mathrm{Fml}_k(\mathcal{L})$ and $z \in \mathrm{Vbl}$. By 3.44, since t is substitutable for x in ϕ, either $x \notin \mathrm{Fr}(\phi)$, or, $z \notin \mathrm{Fr}(t)$ and t is substitutable for x in ψ. In the first case, $\phi(x/t)$ is ϕ and $\mathcal{M} \models \phi[h]$ if and only if $\mathcal{M} \models \phi[h\binom{x}{a}]$ for any $a \in M$ because $x \notin \mathrm{Fr}(\phi)$. So we just need to consider the second case. Suppose $z \notin \mathrm{Fr}(t)$ and t is substitutable for x in ψ. We split further into 2 cases. First suppose that $x = z$. Then $(\forall z \psi)(x/t)$ is $\forall x \psi$. Thus the statement of the lemma holds because $x \notin \mathrm{Fr}(\forall x \psi)$. Now suppose that $x \neq z$. Then $\phi(x/t)$ is $\forall z \psi(x/t)$. So $\mathcal{M} \models \phi(x/t)[h]$ if and only if $\mathcal{M} \models \psi(x/t)[h\binom{z}{a}]$ for all $a \in M$. By the induction hypothesis, this is true if and only if $\mathcal{M} \models \psi[h\binom{z}{a}\binom{x}{t^{\mathcal{M}}}]$. Since $x \neq z$, this is true if and only if $\mathcal{M} \models \phi[h\binom{x}{a}]$ as required. The lemma now follows by induction on the complexity of formulas. $\square$

The work to prove 3.43 is mostly in the proof of the previous lemma.

Proof of Proposition 3.43. We suppose that ϕ, t and x are as in the statement of the proposition. Let $\mathcal{M}$ be an $\mathcal{L}$-structure with domain M and let h be an assignment of $\mathcal{M}$. We need to show $\mathcal{M} \models ((\forall x \phi) \to \phi(x/t))[h]$. According to Tarski's Truth Definition, we need to show that if $\mathcal{M} \models (\forall x\phi)[h]$ then $\mathcal{M} \models \phi(x/t)[h]$. Suppose $\mathcal{M} \models (\forall x\phi)[h]$. Then for all $a \in M$, $\mathcal{M} \models \phi[h\binom{x}{a}]$. So, in particular, $\mathcal{M} \models \phi[h\left(\begin{smallmatrix} x \\ t^{\mathcal{M}}[h] \end{smallmatrix}\right)]$. By 3.45 and since t is substitutable for x in ϕ, this is equivalent to $\mathcal{M} \models \phi(x/t)[h]$ as required. $\square$

3.4 Proofs

I have deliberately chosen a Hilbert style proof system which is as similar as possible to the one we used for Propositional Logic. Like in Propositional Logic we will have a number of axioms and one deduction rule (modus ponens). If you look in books, even amongst those that use a Hilbert style proof system, you will see a lot of variance particularly amongst what formulas are chosen to be axioms. One fairly unusual choice I have made is to have only one deduction rule whereas many other sources also have a "generalisation rule". I have removed the need for this by adding extra axioms.

Our first task for this section is to define what a deduction is for predicate logic and hence to define the symbol $\vdash$.

Definition 3.46. *Let $\mathcal{L}$ be a (first order) language. Each of the following $\mathcal{L}$-formulas are called* ***logical axioms*** *(of $\mathcal{L}$), where ϕ, ψ and γ are $\mathcal{L}$-formulas and x is a variable:*

(AxProp) *(a)* $\phi \to (\psi \to \phi)$
(b) $(\phi \to (\psi \to \gamma)) \to ((\phi \to \psi) \to (\phi \to \gamma))$
(c) $((\neg\psi) \to (\neg\phi)) \to (\phi \to \psi)$

(Ax$\forall\to$) $\forall x(\phi \to \psi) \to ((\forall x\phi) \to (\forall x\psi))$

(AxSub) $(\forall x\phi) \to \phi(x/t)$, *where $t \in tm(\mathcal{L})$ is substitutable in ϕ for x*

(AxGen) $\phi \to \forall x\phi$, *where x is not a free variable of ϕ*

(Ax x=x) $x = x$

(AxEq) $x = y \to (\phi \to \phi(x/y))$ *where y is substitutable for x in ϕ.*

(Ax$\forall$) *Any formula of the form*

$$\forall x_1 \ldots \forall x_n\ \alpha$$

where α is one of the formulas introduced by the other logical axioms above and $x_1, ..., x_n$ are variables.

We call ***(AxSub)*** *the substitution axiom,* ***(AxEq)*** *the axiom of equality and* ***(AxGen)*** *the generalisation axiom.*

I will also refer to these $\mathcal{L}$-formulas as the axioms for predicate logic. This is a tiny bit dodgy because the axioms are actually different depending on $\mathcal{L}$ because the formulas involved are $\mathcal{L}$-formulas.

Definition 3.47. *Let $\mathcal{L}$ be a (first order) language. Let Σ be a set of $\mathcal{L}$-formulas. A* ***deduction*** *from Σ is a finite tuple $(\phi_1, \phi_2, \ldots, \phi_n)$ of $\mathcal{L}$-formulas such that for each $1 \leq k \leq n$ one of the following is true:*

1. *ϕ_k is a logical axiom (of $\mathcal{L}$);*
2. *$\phi_k \in \Sigma$; or*
3. *there exist $i, j < k$ such that ϕ_i is $(\phi_j \to \phi_k)$.*

We will call a deduction $(\phi_1, \ldots, \phi_n)$ from Σ with $\phi_n = \phi$ ***a deduction of ϕ from Σ*** *and write $\Sigma \vdash \phi$ to mean there is a deduction of ϕ from Σ.*

Note that I copied and pasted this definition from the propositional logic section, swapped "axiom" for logical axiom (of $\mathcal{L}$), swapped t for ϕ and S for Σ. We will call the final rule "**modus ponens**" and sometimes abbreviate it MP. Exactly as in propositional logic, I will display deductions as a series of numbered lines with a justification of each line on the right hand side. Finally, I will some times say Σ proves ϕ to mean $\Sigma \vdash \phi$.

We get the same immediate consequences of the definition of a deduction as we did for propositional logic.

Remark 3.48. *The following are immediate consequences of the definition of a deduction.*

(i) If $(\phi_1, \ldots, \phi_n)$ is a deduction from Σ then so is $(\phi_1, \ldots, \phi_m)$ for all $m \leq n$. In particular, $\Sigma \vdash \phi_m$.

(ii) If $(\phi_1, \ldots, \phi_n)$ and $(\psi_1, \ldots, \psi_m)$ are deductions from Σ then so is their concatenation, that is $(\phi_1, \ldots, \phi_n, \psi_1, \ldots, \psi_m)$.

(iii) If $\phi \in \Sigma$ then $\Sigma \vdash \phi$.

(iv) If ϕ is an instance of a logic axiom (of $\mathcal{L}$) then $\vdash \phi$.

(v) If $\Sigma' \vdash \phi$ and $\Sigma' \subseteq \Sigma$ then $\Sigma \vdash \phi$.

I didn't include the next statement in the propositional logic section but the appropriate version does also hold for propositional logic.

Corollary 3.49 (Modus Ponens for proofs)**.** *If $\Sigma \vdash \phi \to \psi$ and $\Sigma \vdash \phi$ then $\Sigma \vdash \psi$.*

We also get a "Proofs are finite" Theorem for Predicate Logic.

Theorem 3.50 ("Proofs are finite")**.** *Let $\mathcal{L}$ be a (first order) language. For all $\Sigma \subseteq Fml(\mathcal{L})$ and $\phi \in Fml(\mathcal{L})$, the following are equivalent:*

(i) $\Sigma \vdash \phi$.

(ii) There is a finite subset $\Sigma' \subseteq \Sigma$ with $\Sigma' \vdash \phi$.

Proof. The audience should tell me how to prove this. It is the same for Propositional Logic where it was an exercise. □

Remark 3.51. *Replacing propositional variables by $\mathcal{L}$-formulas in a deduction in propositional logic gives us a deduction in predicate logic. For instance, we show that $\vdash (s \to s)$ for s a propositional formula. Replacing every instance of s in the deduction by a formula ϕ in a (first order) language $\mathcal{L}$ give us a deduction in predicate logic and this shows that $\vdash (\phi \to \phi)$*

Theorem 3.52 (Deduction Theorem). *Let ϕ, ψ be $\mathcal{L}$-formulas and let Σ be a set of $\mathcal{L}$-formulas. Then*

$$\Sigma \vdash (\phi \to \psi) \textit{ if and only if } \Sigma \cup \{\phi\} \vdash \psi.$$

Proof. Exercise: Check that the proof for propositional logic works for predicate logic (with cosmetic changes). □

The next theorem is usually included as an extra deduction rule but since we have done that, we need to prove it.

Theorem 3.53 (Generalisation Theorem). *Let Σ be a set of $\mathcal{L}$-formulas and let ϕ be an $\mathcal{L}$-formula. If $x \in Vbl$ does not occur as a free instance in any formula in Σ then*

$$\Sigma \vdash \phi \textit{ if and only if } \Sigma \vdash \forall x\, \phi.$$

Note: *There may well be a free instance of x occurring in ϕ.*

Proof. The reverse direction holds even without the assumption that x does not occur freely in any formula from Σ by **(AxSub)** applied with $x = t$ and Modus Ponens for proofs.

The more difficult and important direction is the forwards direction, i.e., $\Sigma \vdash \phi$ implies $\Sigma \vdash \forall x\, \phi$.

Suppose that the variable x does not occur as a free instance in any formula in Σ and that $(\phi_1, \dots, \phi_n)$ is a deduction. We will prove that $\Sigma \vdash \forall x\, \phi$ by induction on the length of the deduction.

If ϕ is a logical axiom then, by **(Ax$\forall$)**, $\forall x\phi$ is a logical axiom and thus $\Sigma \vdash \forall x\phi$. If $\phi \in \Sigma$, then by assumption, x does not occur freely in ϕ. Hence, by **(AxGen)**, $\phi \to \forall x\phi$ is a logical axiom. Applying Modus Ponens for proofs we get $\Sigma \vdash \forall x\phi$.

For the induction step, suppose $(\phi_1, ..., \phi_{n+1})$ is a proof from Σ with $\phi = \phi_{n+1}$. From what we have seen above we may assume that ϕ is neither a logical axiom nor an element of Σ. Hence there are $i, j \leq n$ such that ϕ_j is $\phi_i \to \phi_{n+1}$.

By **(Ax$\forall\to$)** we have that

$$\forall x(\phi_i \to \phi_{n+1}) \;\to\; (\forall x\phi_i \;\to\; \forall x\phi_{n+1})$$

is a logical axiom. By the induction hypothesis we know $\Sigma \vdash \forall x\phi_j$, in other words $\Sigma \vdash \forall x(\phi_i \to \phi_{n+1})$. So by Modus Ponens for proofs we get $\Sigma \vdash \forall x\phi_i \;\to\; \forall x\phi_{n+1}$. Since also $\Sigma \vdash \forall x\phi_i$ by induction we may apply Modus Ponens for proofs again to obtain $\Sigma \vdash \forall x\phi_{n+1}$. □

3.5 Completeness

Using propositional logic as a guide, we first prove the soundness theorem for predicate logic.

Proposition 3.54 (Soundness Theorem). *Let $\mathcal{L}$ be a (first order) language, $\Sigma \subseteq Fml(\mathcal{L})$ and $\phi \in Fml(\mathcal{L})$. If $\Sigma \vdash \phi$ then $\Sigma \models \phi$.*

The proof goes as in propositional logic (because our only deduction rule is modus ponens). We need to input the fact that for all logical axioms ϕ, we have $\models \phi$. This is more work than in propositional logic because our axioms are more complicated and more numerous.

We have already done some work towards the proof of the next lemma and some of the remaining parts will be on your exercise sheet.

Lemma 3.55. *Let $\mathcal{L}$ be a (first order) language. For all logical axioms ϕ (of $\mathcal{L}$), $\models \phi$.*

Proof. If ϕ is an instance of one of the axioms **(AxProp)** then $\models \phi$ by 3.22. If ϕ is an instance of **(AxGen)** then $\models \phi$ by 3.36. If ϕ is an instance of **(AxSub)** then $\models \phi$ by 3.43. The work towards proving $\models \phi$ where ϕ is an instance of **(AxEq)** is started on Exercise Sheet 7 in Exercises 3 and 4. There will be more on Exercise Sheet 8 (I will include the full details after Exercise Sheet 8 has been released). The proof that $\models \phi$ when ϕ is an instance of **(Ax$\forall \rightarrow$)** will be on Exercise Sheet 8 - this is a relatively straightforward exercise.

It remains to show that $\models \phi$ when ϕ is an instance of **(Ax$\forall$)**. If ϕ is an instance of **(Ax$\forall$)** then it is of the form $\forall x_1 \ldots \forall x_n \, \psi$ where ψ is an instance of one of the other logical axioms. We have shown (or at least, in some cases, you will show on your exercise sheets) that for all $\mathcal{L}$-structures $\mathcal{M}$ with domain M and assignments $h : \mathrm{Vbl} \to M$, $\mathcal{M} \models \psi[h]$. For any variable x, Tarski's Truth Definition says that $\mathcal{M} \models (\forall x \psi)$ if and only if for all $a \in M$, $\mathcal{M} \models \psi[h\binom{x}{a}]$. Since $h\binom{x}{a}$ is also an assignment, this is true. To prove $\mathcal{M} \models \phi[h]$ we just iterate this proof. ☐

We now prove the Soundness Theorem.

Proof of the Soundness Theorem. The proof is exactly as for the Soundness Theorem for Propositional Logic (2.48) except we replace 2.21 with 3.55 and 2.26 with 3.34. ☐

End of L27 & L28

Definition 3.56. *We say that a set of $\mathcal{L}$-sentences Σ is **consistent** if $\Sigma \nvdash \perp$ and **inconsistent** if $\Sigma \vdash \perp$.*

Definition 3.57. *We say that a set of $\mathcal{L}$-sentences Σ is **satisfiable** if $\Sigma \not\models \perp$ and **unsatisfiable** if $\Sigma \models \perp$.*

The Soundness Theorem has a useful corollary. We will essentially never prove that a set of $\mathcal{L}$-sentences is consistent syntactically. Instead, we will always use the Soundness Theorem.

Corollary 3.58. *If Σ a set of $\mathcal{L}$-sentences is satisfiable, that is, has a model, then Σ is consistent.*

The following theorem is called Gödel's Completeness Theorem. I will also refer to it as the Completeness Theorem for Predicate Logic.

Theorem 3.59. *Let $\mathcal{L}$ be a (first order) language. Let Σ be a set of $\mathcal{L}$-formulas and let ϕ be an $\mathcal{L}$-formula. Then $\Sigma \models \phi$ if and only if $\Sigma \vdash \phi$.*

Proof. The proof of this theorem is only examinable for 4th years. The proof is in section 4. ☐

Corollary 3.60. *If Σ a set of $\mathcal{L}$-sentences is consistent then Σ is satisfiable.*

Theorem 3.61 (The Compactness Theorem). *Let $\mathcal{L}$ be a (first order) language and let Σ be a set of $\mathcal{L}$-sentences. If every finite subset of Σ is satisfiable then Σ is satisfiable.*

Proof. We prove the contrapositive of the theorem. That is, we show that if a set of $\mathcal{L}$-sentences Σ does not have a model then there is some finite subset Σ_0 of Σ which does not have a model. The Completeness Theorem states that if Δ is a set of $\mathcal{L}$-formulas and ϕ is an $\mathcal{L}$-formula then $\Delta \vdash \phi$ if and only if $\Delta \vDash \phi$. Suppose Σ does not have a model. Then $\Sigma \vDash \perp$. Therefore $\Sigma \vdash \perp$. Since "Proofs are finite", there is a subset Σ_0 of Σ such that $\Sigma_0 \vdash \perp$. By the Completeness Theorem, $\Sigma_0 \models \perp$. □

The next 3rd year section is 5.

4 Proof of the Completeness Theorem for Predicate Logic (4th year material)

The content of this section is one of the advanced topics. We will refer forward to material in the next section at least once. The aim of this section is to prove the Model Existence Theorem and to deduce from it Gödel's Completeness Theorem.

Theorem 4.1 (Model Existence Theorem). *Let $\mathcal{L}$ be a (first order) language. If Σ is a consistent set of $\mathcal{L}$-sentences then Σ has a model.*

Although this theorem is true without any assumption on $\mathcal{L}$, to avoid using Zorn's Lemma, we will prove it only for the case when $\mathcal{L}$ and hence $\mathrm{Fml}(\mathcal{L})$ are countable. I will make it clear exactly when I am using countability of $\mathcal{L}$.

The first thing we need is a tool for constructing $\mathcal{L}$-structures out of sentences. What we will do is a bit more general than what we actually need for the proof of the Model Existence Theorem. The $\mathcal{L}$-structures we will construct are called term algebras. They are built out of the constant terms of $\mathcal{L}$. If our language has no constant symbols then it has no constant terms. So we work under the assumption that our language has at least one constant symbol. This might seem a bit strange at this point but by the time we actually get to using them we will have extended our original language to a language with lots of constants - so it won't cause any harm.

Term Algebras

Definition 4.2. *Let $\mathcal{L}$ be a language and $\Sigma \subseteq Fml(\mathcal{L})$.*

(i) Let $ctm(\mathcal{L})$ denote the set of constant terms of $\mathcal{L}$.

(ii) Define the relation $\sim_\Sigma$ on $ctm(\mathcal{L})$ by $t_1 \sim_\Sigma t_2$ for $t_1, t_2 \in ctm(\mathcal{L})$ if $\Sigma \vdash t_1 = t_2$.

Lemma 4.3. *Let $\Sigma \subseteq Fml(\mathcal{L})$ and let $t_1, s_1, \ldots, t_n, s_n \in ctm(\mathcal{L})$.*

(i) $\sim_\Sigma$ is an equivalence relation.

(ii) If F is a function symbol of $\mathcal{L}$ of arity n such that $t_i \sim_\Sigma s_i$ for each $1 \leq i \leq n$ then $F(t_1, \ldots, t_n) \sim_\Sigma F(s_1, \ldots, s_n)$.

(iii) If R is a relation symbol of $\mathcal{L}$ of arity n such that $t_i \sim_\Sigma s_i$ for each $1 \leq i \leq n$ then $\Sigma \vdash R(t_1, \ldots, t_n) \leftrightarrow R(s_1, \ldots, s_n)$.

Proof. (i) Let $t \in \mathrm{ctm}(\mathcal{L})$. By **(Ax x=x)** and **(Ax$\forall$)**,$\vdash v_1 = v_1$ and $\vdash \forall v_1\, v_1 = v_1$. Thus, by **(AxSub)** and modus ponens, $\vdash t = t$. So $\sim_\Sigma$ is reflexive.

From Exercise Sheet 8, we know that for all variables x, y, z

$$\vdash \forall x, y(x = y \to y = x)$$

and

$$\vdash \forall x, y, z(x = y \to (y = z \to x = z)).$$

Suppose $t_1 \sim_\Sigma t_2$. Then $\Sigma \vdash t_1 = t_2$. By **(AxSub)** and modus ponens,

$$\Sigma \vdash t_2 = t_1.$$

So $\sim_\Sigma$ is symmetric.

Suppose $t_1 \sim_\Sigma t_2$ and $t_2 \sim_\Sigma t_3$. Then $\Sigma \vdash t_1 = t_2$ and $\Sigma \vdash t_2 = t_3$. By **(AxSub)**,

$$\Sigma \vdash (t_1 = t_2 \to (t_2 = t_3 \to t_1 = t_3)).$$

By modus ponens applied twice,

$$\Sigma \vdash t_1 = t_3.$$

So $t_1 \sim_\Sigma t_3$. Therefore $\sim_\Sigma$ is transitive.

(ii) I will do the case where F is a unary function symbol. Suppose that $\Sigma \vdash t = s$. We need to show that $\Sigma \vdash F(t) = F(s)$. Let $x, y, z \in \mathrm{Vbl}$ which are distinct and don't occur in t or s.

By **(AxEq)**,

$$\vdash x = y \to (z = F(x) \to z = F(y)).$$

By **(AxSub)**,

$$\vdash t = s \to (z = F(t) \to z = F(s)).$$

Using modus ponens,

$$\Sigma \vdash z = F(t) \to z = F(s).$$

Using **(AxSub)**,

$$\Sigma \vdash F(t) = F(t) \to F(t) = F(s).$$

Since

$$\vdash \forall x\, x = x,$$

by **(AxSub)** and modus ponens,

$$\Sigma \vdash F(t) = F(t).$$

Therefore, by modus ponens,

$$\Sigma \vdash F(t) = F(s)$$

as required.

(iii) This is a similar nightmare to (ii).

□

Definition 4.4 (The term algebra). *Let $\mathcal{L}$ be a language with at least one constant symbol and let $\Sigma \subseteq Fml(\mathcal{L})$. We define an $\mathcal{L}$-structure $\mathrm{TmAlg}(\Sigma)$ called the **term algebra of** Σ. The domain of $\mathrm{TmAlg}(\Sigma)$ is*

$$ctm(\mathcal{L})/\sim_\Sigma .$$

We write $[t]_\Sigma$ for the $\sim_\Sigma$-equivalence class of t.

(i) For each relation symbol R of $\mathcal{L}$ of arity n, we define

$$R^{\mathrm{TmAlg}(\Sigma)} := \{([t_1]_\Sigma, \dots, [t_n]_\Sigma) \mid \Sigma \vdash R(t_1, \dots, t_n)\}.$$

(ii) For each function symbol F of $\mathcal{L}$ of arity n we define

$$F^{\mathrm{TmAlg}(\Sigma)}([t_1]_\Sigma, \dots, [t_n]_\Sigma) = [F(t_1, \dots, t_n)]_\Sigma.$$

(iii) For c a constant symbol of $\mathcal{L}$ we define

$$c^{\mathrm{TmAlg}(\Sigma)} := [c]_\Sigma.$$

For a single formula σ, we define the term algebra of σ to be $\mathrm{TmAlg}(\{\sigma\})$.

We do some examples.

Examples 4.5.

(i) Let $\mathcal{L} := \langle c \rangle$ where c is a constant symbol. Then $ctm(\mathcal{L})$ has just one element c. This means that no matter what set of $\mathcal{L}$-sentences with pick, $\mathrm{TmAlg}(\Sigma)$ has just one element. Let $\Sigma = \{\exists v_1\, v_1 \neq c\}$. Then $\mathrm{TmAlg}(\Sigma)$ is the $\mathcal{L}$-structure with domain $\{[c]_\Sigma\}$ c interpreted as $[c]_\Sigma$. Note, $\mathrm{TmAlg}(\Sigma)$ is not a model of Σ in this case.

(ii) Let $\mathcal{L} := \langle R, c, d, e \rangle$ where R is a unary relation symbol and c, d, e are constant symbols. Then $ctm(\mathcal{L})$ has 3 elements c, d, e. Let σ be the formula $c = d \vee c = e$. The term algebra of σ has at most 3 elements $[c]_\sigma, [d]_\sigma$ and $[e]_\sigma$.

We can check using the soundness theorem that $\sigma \nvdash c = d$, $\sigma \nvdash c = e$ and $\sigma \nvdash d = e$. I will show that $\sigma \nvdash c = d$. By the Soundness Theorem, $\sigma \nvDash c = d$ implies $\sigma \nvdash c = d$. Let $\mathcal{M}$ be the $\mathcal{L}$-structure with domain $\{1, 2\}$, $R^{\mathcal{M}} := \{1, 2\}$ and

$$c^{\mathcal{M}} := 1, d^{\mathcal{M}} := 2 \text{ and } e^{\mathcal{M}} = 1.$$

Then $\mathcal{M} \models c = d \vee c = e$ because $c^{\mathcal{M}} = e^{\mathcal{M}}$. But $\mathcal{M} \nvDash c = d$. Therefore $\sigma \nvDash c = d$. So $\sigma \nvdash c = d$ as required.

This shows that $\mathrm{TmAlg}(\sigma)$ has 3 elements $[c]_\sigma, [d]_\sigma$ and $[e]_\sigma$. I claim $R^{\mathrm{TmAlg}(\sigma)} = \emptyset$. To show this, we should show that $\sigma \nvdash R(t)$ for all $t \in ctm(\mathcal{L})$. Let $\mathcal{M}$ be the $\mathcal{L}$-structure with domain $\{1\}$ and $R^{\mathcal{M}} = \emptyset$. Then $\mathcal{M} \models \sigma$ but $\mathcal{M} \nvDash R(t)$ for all $t \in ctm(\mathcal{L})$ because $R^{\mathcal{M}} = \emptyset$.

Proposition 4.6. *Let $\mathcal{L}$ be a language with at least one constant symbol and let $\Sigma \subseteq Fml(\mathcal{L})$. Let $n \in \mathbb{N}$, $t_1, \dots, t_n \in ctm(\mathcal{L})$, $x_1, \dots, x_n \in Vbl$ pairwise distinct and let h be an assignment of $\mathrm{TmAlg}(\Sigma)$ with $h(x_i) = [t_i]_\Sigma$ $(1 \leq i \leq n)$.*

(i) If $t(x_1, ..., x_n) \in tm(\mathcal{L})$ then $t^{\mathrm{TmAlg}(\Sigma)}[h] = [t(x_1/t_1, ..., x_n/t_n)]_\Sigma$.

(ii) If $\phi(x_1, ..., x_n) \in Fml(\mathcal{L})$ is atomic then

$$\mathrm{TmAlg}(\Sigma) \models \phi[h] \textit{ if and only if } \Sigma \vdash \phi(x_1/t_1, ..., x_n/t_n).$$

Proof. (i) The first statement is proved by induction on the complexity of the term t. If t has complexity 0 then t is either a variable or a constant symbol.

Suppose t is the variable x_i for some $1 \leq i \leq n$. Then $t(x_1/t_1, \ldots, x_n/t_n)$ is t_i and $t^{\mathrm{TmAlg}}(\Sigma)[h] = h(x_i)$. By assumption, $h(x_i) = [t_i]_\Sigma$. Therefore $t^{\mathrm{TmAlg}(\Sigma)}[h] = [t(x_1/t_1, ..., x_n/t_n)]_\Sigma$ as required.

Suppose t is a constant symbol c. Then $t(x_1/t_1, \ldots, x_n/t_n)$ is c and $t^{\mathrm{TmAlg}}(\Sigma)[h] = c^{\mathrm{TmAlg}(\Sigma)}$. By definition of $\mathrm{TmAlg}(\Sigma)$, $c^{\mathrm{TmAlg}(\Sigma)} = [c]_\Sigma$. So we get the required conclusion.

Now suppose that for all terms t of complexity less than or equal to n, (i) holds. Suppose $s \in \mathrm{tm}_{n+1}(\mathcal{L})$. Then s is of the form $F(s_1, \ldots, s_m)$ for F a function symbol of arity m and terms $s_i \in \mathrm{tm}_n(\mathcal{L})$. By definition, $F(s_1, \ldots, s_m)[h] = F^{\mathrm{TmAlg}(\Sigma)}(s_1[h], \ldots, s_m[h])$. By the induction hypothesis,

$$\begin{aligned} F^{\mathrm{TmAlg}(\Sigma)}(s_1[h], \ldots, s_m[h]) &= F^{\mathrm{TmAlg}(\Sigma)}([s_1(x_1/t_1, ..., x_n/t_n)]_\Sigma, \ldots, [s_m(x_1/t_1, ..., x_n/t_n)]_\Sigma) \\ &= [F(s_1(x_1/t_1, ..., x_n/t_n), \ldots, s_m(x_1/t_1, ..., x_n/t_n))]_\Sigma \\ &= [s(x_1/t_1, \ldots, x_n/t_n)]_\Sigma. \end{aligned}$$

(ii) This is an application of (i) and the definition of the term algebra. □

We've seen that the term algebra of a set of sentences Σ is not always a model of Σ. In the 2 cases we have seen there are ways to fix this.

We first consider 4.5(i). Let $\mathcal{L} := \langle c \rangle$ where c is a constant symbol and let σ be the formula $\exists v_1 v_1 \neq c$. In order to make the term algebra of $\Sigma := \{\sigma\}$ into a model of σ we can extend the language $\mathcal{L}$ by a constant c_σ to get $\mathcal{L}(c_\sigma)$ and replace Σ with

$$\{\sigma, (\exists v_1\, v_1 \neq c) \to c_\sigma \neq c\}.$$

Then $\mathrm{TmAlg}(\Sigma) \models \sigma$.

Now consider 4.5(ii). Let $\mathcal{L} := \langle R, c, d, e \rangle$ and let σ be $c = d \vee c = e$. If we replace $\Sigma := \{\sigma\}$ with a consistent set of $\mathcal{L}$-sentences Σ^+ containing Σ such that for all $\mathcal{L}$-sentences ϕ, either $\phi \in \Sigma^+$ or $\neg\phi \in \Sigma^+$ then a little work will allow you to conclude that either $c = d \in \Sigma^+$ or $c = e \in \Sigma^+$. This will force $\mathrm{TmAlg}(\Sigma^+)$ to be a model of σ.

Definition 4.7. *Let $\mathcal{L}$ be a language with set of constant symbols $\mathcal{C}$. We write $Fml(\mathcal{L})(1)$ for the set of all $\mathcal{L}$-formulas with at most one free variable.*

*Let Σ be a set of $\mathcal{L}$-sentences. A **system of witnesses** for Σ is a map*

$$\gamma : Fml(\mathcal{L})(1) \to \mathcal{C}$$

such that for all $x \in Vbl$ and all $\phi(x) \in Fml(\mathcal{L})(1)$ we have

$$\Sigma \vdash (\exists x\, \phi) \to \phi(x/\gamma(\phi)).$$

Definition 4.8. *An $\mathcal{L}$-**theory** is a consistent set of $\mathcal{L}$-sentences. An $\mathcal{L}$-theory T is **complete** if for all $\mathcal{L}$-sentences ϕ, either $\phi \in T$ or $\neg\phi \in T$.*

The set of sentences satisfied by an $\mathcal{L}$-structure is an example of a complete theory (in fact, all complete theories are of this form but we need the Completeness Theorem to prove this).

Proposition 4.9. *If Σ is a complete $\mathcal{L}$-theory which has a system of witnesses then the term algebra of Σ is a model of Σ.*

We can use the Completeness Theorem for Propositional Logic to prove 1. and 2. of the following lemma.

Lemma 4.10. *Let Σ be a complete $\mathcal{L}$-theory. Then*

1. *$\Sigma \vdash \phi$ if and only if $\phi \in \Sigma$;*
2. *$\phi_1 \to \phi_2 \in \Sigma$ if and only if $\phi_1, \phi_2 \in \Sigma$ or $\phi_1 \notin \Sigma$; and*
3. *$\forall y\, \psi \notin \Sigma$ implies $\exists y\, \neg\psi \in \Sigma$.*

This proof is why we are first proving the completeness theorem for sets of sentences. Essentially it would be much more confusing if we tried to prove a version for formulas.

Proof of Proposition 4.9. We prove by induction on the complexity of $\phi(x_1, \dots, x_n) \in \mathrm{Fml}(\mathcal{L})$ that: For all constant terms $t_1, \dots, t_n$,

$$\phi(x_1/t_1, \dots, x_n/t_n) \in \Sigma \text{ if and only if } \mathrm{TmAlg}(\Sigma) \models \phi(x_1/t_1, \dots, x_n/t_n). \qquad (\dagger)$$

If ϕ is atomic then this is 4.6. The case when ϕ is $\perp$ follows from the fact that Σ is consistent.

Suppose $(\dagger)$ is true for ϕ_1 and ϕ_2. We show that $(\dagger)$ is true for $\phi_1 \to \phi_2$.

Since Σ is complete, $\phi_1 \to \phi_2 \in \Sigma$ if and only if $\phi_1, \phi_2 \in \Sigma$ or $\phi_1 \notin \Sigma$. Since $(\dagger)$ holds for ϕ_1 and ϕ_2,

$$\mathrm{TmAlg}(\Sigma) \models \phi_1(x_1/t_1, \dots, x_n/t_n) \text{ and } \mathrm{TmAlg}(\Sigma) \models \phi_2(x_1/t_1, \dots, x_n/t_n),$$

or

$$\mathrm{TmAlg}(\Sigma) \not\models \phi_1(x_1/t_1, \dots, x_n/t_n).$$

Considering Tarski's Truth definition, this is true if and only if

$$\mathrm{TmAlg}(\Sigma) \models \phi_1(x_1/t_1, \dots, x_n/t_n) \to \phi_2(x_1/t_1, \dots, x_n/t_n).$$

We now need to consider the case where ϕ is $\forall y\, \psi$ for some $y \in \mathrm{Vbl}$. Suppose $(\dagger)$ is true for ψ. By definition, y is not a free variable of ϕ. Since $\mathrm{Fr}(\phi) \subseteq \{x_1, \dots, x_n\}$, no harm is done by assuming that $y \neq x_i$ for $1 \leq i \leq n$.

We first prove the forward direction of $(\dagger)$ for ϕ. Suppose $\phi(x_1/t_1, \dots, x_n/t_n) \in \Sigma$. Then $\forall y\, \psi(x_1/t_1, \dots, x_n/t_n, y) \in \Sigma$. By the substitution axiom $\Sigma \vdash \psi(x_1/t_1, \dots, x_n/t_n, y/t)$ for all constant terms t. Thus, by the induction hypothesis,

$$\mathrm{TmAlg}(\Sigma) \models \psi(x_1/t_1, \dots, x_n/t_n, y/t).$$

Therefore

$$\mathrm{TmAlg}(\Sigma) \models \psi(x_1/t_1, \dots, x_n/t_n)[h]$$

for all h with $h(y) = [t]_\Sigma$. Since t was an arbitrary constant term, this means

$$\mathrm{TmAlg}(\Sigma) \models \phi(x_1/t_1, \dots, x_n/t_n)$$

as we wanted.

We now prove the reverse direction of (†) for ϕ. This is where we need that Σ has a system of witnesses. Let

$$\gamma : \mathrm{Fml}(\mathcal{L})(1) \to \mathcal{C}$$

be a system of witnesses for Σ.

Suppose $\mathrm{TmAlg}(\Sigma) \models \forall y\, \psi(x_1/t_1, \ldots, x_n/t_n, y)$ but $\forall y\, \psi(x_1/t_1, \ldots, x_n/t_n, y) \notin \Sigma$ with the intention of deriving a contradition. Since Σ is complete, by 4.10, $\exists y\, \neg\psi(x_1/t_1, \ldots, x_n/t_n, y) \in \Sigma$. Let $c := \gamma(\neg\psi(x_1/t_1, \ldots, x_n/t_n, y))$. Since γ is a system of witnesses for Σ we have

$$\Sigma \vdash \exists y\, \neg\psi(x_1/t_1, \ldots, x_n/t_n, y) \to \neg\psi(x_1/t_1, \ldots, x_n/t_n, y/c).$$

Then modus ponens gives

$$\Sigma \vdash \neg\psi(x_1/t_1, \ldots, x_n/t_n, y/c)$$

Hence $\psi(x_1/t_1, \ldots, x_n/t_n, y/c) \notin \Sigma$. Since (†) holds for ψ, this implies

$$\mathrm{TmAlg}(\Sigma) \not\models \psi(x_1/t_1, \ldots, x_n/t_n, y/c).$$

But then by 3.45, this contradicts our assumption that

$$\mathrm{TmAlg}(\Sigma) \models \forall y\, \psi(x_1/t_1, \ldots, x_n/t_n).$$

□

Adding witnesses

Proposition 4.11. *Let $\mathcal{L}$ be a language and let Σ be a set of $\mathcal{L}$-formulas. Let $\gamma : Fml(\mathcal{L})(1) \to \mathcal{D}$ be a bijection onto a some set of constant symbols disjoint from those of $\mathcal{L}$. If Σ is consistent then so is the set of $\mathcal{L}(\mathcal{D})$-sentences*

$$\Sigma^W := \Sigma \cup \{\exists x\, \phi(x) \to \phi(x/\gamma(\phi)) \mid \phi \in Fml(\mathcal{L})(1)\}.$$

In order to prove this we need some not totally trivial facts.

Proposition 4.12. *Let $\mathcal{L}$ be a language, let Σ a set of $\mathcal{L}$-formulas and let c be a constant symbol which is not a constant symbol of $\mathcal{L}$. Suppose that y is a variable that does not occur in ϕ. Let ϕ^c_y be the result of replacing every instance of c in ϕ by y. Then $\Sigma \vdash_{\mathcal{L}(c)} \phi$ implies $\Sigma \vdash_{\mathcal{L}} \phi^c_y$.*

Here $\vdash_{\mathcal{L}}$ indicates that there is an "$\mathcal{L}$-deduction" of ϕ from Σ.

Proof. Omitted for now. □

Lemma 4.13. *Let $\mathcal{L}$ be a language, $\mathcal{D}$ a set of constant symbols disjoint from those of $\mathcal{L}$. Let Σ be a set of $\mathcal{L}$-formulas and ϕ an $\mathcal{L}$-formula. Then*

$$\Sigma \vdash_{\mathcal{L}} \phi \text{ if and only if } \Sigma \vdash_{\mathcal{L}(\mathcal{D})} \phi.$$

Proof. Omitted for now. □

Lemma 4.14. *Let ϕ, ψ be $\mathcal{L}$-formulas and Σ a set of $\mathcal{L}$-formulas.*

1. *If y does not occur free in ϕ then $\vdash \forall y\, \phi(x/y) \to \forall x\, \phi$.*

2. *If $\Sigma \vdash \phi \to \psi$ then $\Sigma \vdash \neg\psi \to \neg\phi$.*

3. $\vdash (\phi \to \exists x\, \psi) \to \exists x\, (\phi \to \psi)$

Proof. Omitted for now. □

Remark 4.15. *If Σ has a system of witnesses and Σ^* is a complete theory containing Σ then Σ^* has a system of witnesses (the same system works with no changes).*

Proof of 4.11. **Claim:** If Σ is a finite consistent set of sentences and $\phi \in \mathrm{Fml}(\mathcal{L})(1)$ is such that $\gamma(\phi)$ does not occur in any $\sigma \in \Sigma$ then

$$\Sigma' := \Sigma \cup \{\exists x\, \phi(x) \to \phi(x/\gamma(\phi))\}$$

is consistent.

Suppose for a contradiction that $\Sigma' \vdash \bot$. By the deduction theorem $\Sigma \vdash \neg(\exists x\, \phi(x) \to \phi(x/\gamma(\phi)))$. Let y be a variable that does not occur in any formula in Σ' and let $\psi := \exists x\, \phi(x) \to \phi(x/y)$. By 4.12, $\Sigma \vdash \neg\psi$. The Generalisation Theorem now implies $\Sigma \vdash \forall y\, \neg\psi$. Since $\exists y\, \psi$ is an abbreviation for $\neg\forall y\, \neg\psi$, it is enough to show $\Sigma \vdash \exists y\, \psi$ because this implies $\Sigma \vdash \bot$. By 4.14, $\Sigma \vdash \forall y\, (\neg\phi(x/y)) \to \forall x\, \neg\phi$. By 4.14, $\Sigma \vdash \neg\forall x\, \neg\phi \to \neg\forall y(\neg\phi(x/y))$, i.e., $\Sigma \vdash \exists x\phi \to \exists y\, \phi(x/y)$. By 4.14, $\Sigma \vdash \exists y\, (\exists x\, \phi \to \phi(x/y))$, i.e., $\Sigma \vdash \exists y\, \psi$. So $\Sigma \vdash \bot$. This contradicts our assumption that Σ is consistent. Therefore Σ' is consistent.

We now prove the proposition based on the claim. Suppose $\Sigma^W \vdash \bot$. Since proofs are finite, $\Sigma_0 \vdash \bot$ for some finite subset Σ_0 of Σ^W. We may assume that Σ_0 is of minimal size with this property. Since Σ is consistent, this means there is some consistent set Σ_0' and some $\phi \in \mathrm{Fml}(\mathcal{L})(1)$ such that $\Sigma_0 = \Sigma_0' \cup \{\exists x\, \phi(x) \to \phi(x/\gamma(\phi))\}$. But Σ_0 being inconsistent contradicts the claim. Thus Σ^W is consistent. □

We are not quite there because there is no reason that all $\mathcal{L}(\mathcal{D})$-formulas in one free variable should have witnesses.

Proposition 4.16. *Let Σ be a consistent set of $\mathcal{L}$-sentences. There is an extension $\mathcal{L}^+$ of $\mathcal{L}$ by constants and a consistent set of $\mathcal{L}^+$-sentences $\Sigma^+ \supseteq \Sigma$ such that Σ^+ has a system of witnesses.*

Proof. We iterate the construction from 4.11. Define $\mathcal{L}_0 := \mathcal{L}$ and $\Sigma_0 := \Sigma$. For $i \in \mathbb{N}_0$, let $\mathcal{L}_{i+1} := (\mathcal{L}_i)^W$ where $\mathcal{L}_i^W$ denotes the language $\mathcal{L}(\mathcal{D})$ from 4.11 and $\Sigma_{i+1} := (\Sigma_i)^W$ is as in the statement of 4.11. Then $\mathcal{L}_\infty := \bigcup_{i\in\mathbb{N}_0} \mathcal{L}_i$ is an extension of $\mathcal{L}$ by constants and $\Sigma_\infty := \bigcup_{i\in\mathbb{N}_0} \Sigma_i$ is a consistent set of $\mathcal{L}_\infty$-sentences containing Σ. To see that Σ_∞ is consistent note that, since "Proofs are finite", if $\Sigma_\infty \vdash \bot$ then $\Sigma_i \vdash \bot$ for some $i \in \mathbb{N}_0$. By 4.11, each Σ_i is consistent. Moreover Σ_∞ has a set of witnesses because every $\mathcal{L}_\infty$-formula ϕ is an $\mathcal{L}_i$-formula for some $i \in \mathbb{N}_0$. By construction, if $\phi \in \mathrm{Fml}(\mathcal{L}_i)(1)$ then ϕ "has a witness" gains a witness in $\mathcal{L}_{i+1}$. □

Every consistent set of sentences is contained in a complete theory.

This is the only point in the proof where use the fact that $\mathcal{L}$ is countable. Everything here works as it does for Propositional Logic. For this reason, I may not lecture this bit but I do expect you (4th years and PhD students) to be able to produce the proofs in this subsection in an exam.

Lemma 4.17. *Let Σ be a consistent set of $\mathcal{L}$-sentences and let ϕ be an $\mathcal{L}$-sentence. Then either $\Sigma \cup \{\phi\}$ is consistent or $\Sigma \cup \{(\neg\phi)\}$ is consistent.*

Proof. Suppose $\Sigma \cup \{\phi\} \vdash \bot$. Then, by the Deduction Theorem, $\Sigma \models (\phi \to \bot)$, i.e., $\Sigma \vdash (\neg\phi)$. Hence $\Sigma \cup \{(\neg\phi)\}$ is consistent. □

Proposition 4.18. *Let X be a consistent set of $\mathcal{L}$-sentences. There exists a complete $\mathcal{L}$-theory T such that $X \subseteq T$.*

Proof. We proceed exactly as we did in the proof of 2.52 for Propositional Logic. Let $\phi_1, \phi_2, \dots$ be an enumeration of the $\mathcal{L}$-formulas. Define $\Sigma_1 := X$. For each i, let $\Sigma_{i+1} := \Sigma_i \cup \{\phi\}$ if it is consistent and let $\Sigma_{i+1} := \Sigma_i \cup \{(\neg\phi_i)\}$ otherwise. In either case, Σ_{i+1} is consistent by 4.17. Let $\Sigma_* := \bigcup_{i\in\mathbb{N}} \Sigma_i$. Suppose for a contradiction that Σ_* is not consistent. Then $\Sigma_* \vdash \bot$. Since "Proofs are finite" there is some finite subset $\Sigma' \subseteq \Sigma_*$ such that $\Sigma' \vdash \bot$. But then, by definition of Σ_*, $\Sigma' \subseteq \Sigma_i$ for some $i \in \mathbb{N}$ and hence $\Sigma_i \vdash \bot$ which give a contradiction. Thus Σ_* is consistent. By definition, for all $\mathcal{L}$-sentences ϕ, either $\phi \in \Sigma_*$ or $(\neg\phi) \in \Sigma_*$. Therefore Σ_* is complete. □

Finishing of the proof

We prove the Model Existence Theorem.

Proof of 4.1. Let Σ be a consistent set of $\mathcal{L}$-sentences. By 4.16, there exists $\mathcal{L}^+$ an extension of $\mathcal{L}$ by constants and a set $\Sigma^+ \supseteq \Sigma$ of $\mathcal{L}^+$-sentences such that Σ^+ has a system of witnesses. By 4.18, there is a complete $\mathcal{L}^+$-theory Σ^* containing Σ^+. As we have already observed, Σ^* has a system of witnesses because Σ^+ does. Therefore $\mathrm{TmAlg}(\Sigma^*) \models \Sigma^*$. Hence the $\mathcal{L}$-reduct of $\mathrm{TmAlg}(\Sigma^*)$ is a model of Σ. □

We now reduce a version of the Model Existence Theorem for formulas to the case of sentences.

Corollary 4.19. *Let Σ be a set of $\mathcal{L}$-formulas. If $\Sigma \models \bot$ then $\Sigma \vdash \bot$.*

Proof. Full proof is omitted (at least for now). Essentially, you replace any free variables of formulas in Σ by constant symbols. Suppose $\Sigma \nvdash \bot$. Let $\mathcal{D} := \{c_i \mid i \in \mathbb{N}\}$ be a set of constant symbols. For $\sigma \in \mathrm{Fml}(\mathcal{L})$, let $\sigma(\mathcal{D})$ be the formula obtained from σ by replacing all free instances of v_i by c_i. Let $\Sigma(\mathcal{D}) := \{\sigma(\mathcal{D}) \mid \sigma \in \Sigma\}$. It follows, with some work, that $\Sigma(\mathcal{D}) \nvdash \bot$. Thus, by the Model Existence Theorem, there is an $\mathcal{L}(\mathcal{D})$-structure $\mathcal{M}$ such that $\mathcal{M} \models \Sigma(\mathcal{D})$. Now, let h be an assignment of $\mathcal{M}$ where $h(v_i) = c_i^{\mathcal{M}}$ for all $i \in \mathbb{N}$. Then h is also an assignment of the $\mathcal{L}$-reduct $\mathcal{M}^-$ of $\mathcal{M}$. With a bit of work, we can show that $\mathcal{M}^- \models \sigma[h]$ for all $\sigma \in \Sigma$. Therefore $\Sigma \not\models \bot$. □

We can now deduce the difficult direction of the Completeness Theorem exactly as we did in Propositional Logic.

Proof of the Completeness Theorem for Predicate Logic. Let Σ be a set of $\mathcal{L}$-formulas and let ϕ be an $\mathcal{L}$-formula. By the Soundness Theorem $\Sigma \vdash \phi$ implies $\Sigma \models \phi$. We now prove the converse. Suppose that $\Sigma \models \phi$. Let $\mathcal{M}$ be an $\mathcal{L}$-structure and let h be an assignment of $\mathcal{M}$. Suppose that $\mathcal{M} \models \sigma[h]$ for all $\sigma \in \Sigma$. Since $\Sigma \models \phi$, $\mathcal{M} \models \phi[h]$. Thus, $\mathcal{M} \not\models \neg\phi[h]$. Therefore $\Sigma \cup \{\neg\phi\} \models \bot$. So, by the Model Existence Theorem, $\Sigma \cup \{\neg\phi\} \vdash \bot$. By the Deduction Theorem, this implies $\Sigma \vdash (\neg\phi) \to \bot$ i.e. $\Sigma \vdash \neg\neg\phi$. Since $\neg\neg\phi \to \phi$ is an instance of **(AxProp)**, applying modus ponens gives $\Sigma \vdash \phi$ as required. □

5 Application of the Compactness Theorem and Extension by Constants

We saw a few applications of the Compactness Theorem for propositional logic but these were really just exercises in how to use the Compactness Theorem. The Compactness Theorem for predicate logic is the first theorem in Model Theory. Model Theory is a branch of mathematical logic with close ties and strong applications to the rest of pure mathematics. In particular, there are applied model theorists studying the model theory of most other parts of pure mathematics. There is an alternative model theoretic proof of the compactness theorem.

We start with some consequences of the Compactness Theorem, some of which might be viewed as negative or showing the limits of first order logic.

On Exercise Sheet 7, we introduced the following definition.

Definition 5.1. *We say that a class X of $\mathcal{L}$-structures is* ***axiomatisable*** *if there exists a set of $\mathcal{L}$-sentences Σ such that for all $\mathcal{L}$-structures $\mathcal{M}$,*

$$\mathcal{M} \models \phi \text{ for all } \phi \in \Sigma \text{ if and only if } \mathcal{M} \in X.$$

We call Σ an ***axiomatisation*** *of X or that Σ* ***axiomatises*** *X.*

We say an $\mathcal{L}$-structure is finite/infinite if its domain is finite/infinite. The size (or cardinality) of an $\mathcal{L}$-structure is the size of its domain.

Remark 5.2. *Let $\mathcal{L}$ be a language.*

(i) For each $n \in \mathbb{N}$, there is an $\mathcal{L}$-sentence σ_n such that $\mathcal{M} \models \sigma_n$ if and only if $\mathcal{M}$ has size n.

(ii) There is a set of $\mathcal{L}$-sentences which axiomatise the class of infinite $\mathcal{L}$-structures.

Proof. Exercise Sheets 7 and 8. □

Proposition 5.3. *Let X be an axiomatisable class of $\mathcal{L}$-structures such that for every $n \in \mathbb{N}$ there is an $\mathcal{L}$-structure $\mathcal{M} \in X$ such that the domain of $\mathcal{M}$ has more than n elements. Then X contains an infinite $\mathcal{L}$-structure.*

Proof. Let Σ be the set of $\mathcal{L}$-sentences which axiomatise X. For each $n \in \mathbb{N}$, let σ_n be an $\mathcal{L}$-sentence such that $\mathcal{M} \models \sigma_n$ if and only if $\mathcal{M}$ has size n. We will show that $\Sigma \cup \{\neg\sigma_n \mid n \in \mathbb{N}\}$ is satisfiable using the Compactness Theorem. Let $\Sigma' \subseteq \Sigma \cup \{\neg\sigma_n \mid n \in \mathbb{N}\}$ be finite. Since Σ' is finite, there exists $l \in \mathbb{N}$ such that if $\neg\sigma_n \in \Sigma'$ then $n \leq l$. Take $\mathcal{M} \in X$ such that the size of $\mathcal{M}$ has at least $l+1$ elements. Then $\mathcal{M}$ is a model of Σ and $\mathcal{M} \models \neg\sigma_n$ for all $n \leq l$. Therefore $\mathcal{M}$ is a model of Σ'. So, we have shown that all finite subsets of $\Sigma \cup \{\neg\sigma_n \mid n \in \mathbb{N}\}$ have a model. Therefore, by the Compactness Theorem, $\Sigma \cup \{\neg\sigma_n \mid n \in \mathbb{N}\}$ has a model $\mathcal{M}^*$. Since $\mathcal{M}^*$ satisfies Σ, $\mathcal{M}^* \in X$. For each $n \in \mathbb{N}$, $\mathcal{M}^* \models \neg\sigma_n$. So $\mathcal{M}^*$ is not size n for any $n \in \mathbb{N}$. Therefore $\mathcal{M}^*$ is infinite.

□

20.11.23

Corollary 5.4. *There is an infinite group G which satisfies all sentences ϕ in $\mathcal{L}_{gp} := \langle \cdot, (-)^{-1}, e \rangle$ that are true in all finite groups. We call such a group a* ***psuedofinite group***.

Proof. Let Σ be the set of $\mathcal{L}_{gp}$-sentences ϕ such that ϕ is satisfied by all finite groups. By definition, all finite groups are models of Σ. For each natural number n, the group $(\mathbb{Z}/n\mathbb{Z}, +)$ has size n. Thus, by reffiniteunbounded, Σ has an infinite model. $\square$

This might be viewed as a negative consequence of the Compactness Theorem.

Corollary 5.5. *The class of finite groups, graphs, rings, fields, etc are not axiomatisable (in the appropriate first order language).*

Proof. Each of the classes has members of unbounded finite size. So, by 5.3, if Σ is a set of sentences satisfied by all members of the class then Σ has an infinite model. $\square$

Proposition 5.6. *Let $\mathcal{L}$ be a language. There does not exist an $\mathcal{L}$-sentence which axiomatises the class of infinite $\mathcal{L}$-structures.*

Proof. Suppose for a contradiction that ϕ is an $\mathcal{L}$-sentence which axiomatises the class of infinite $\mathcal{L}$-structures. Then, $\neg\phi$ axiomatises the class of finite $\mathcal{L}$-structures. So it is enough to show that the class of finite $\mathcal{L}$-structures is not axiomatisable. For each $n \in \mathbb{N}$, let $\mathcal{M}_n$ be the $\mathcal{L}$-structure with domain $\{1, \ldots, n\}$, for every relation symbol R of $\mathcal{L}$, let $R^{\mathcal{M}_n} = \emptyset$, for every function symbol F of $\mathcal{L}$, let $F^{\mathcal{M}_n}$ be the function defined by $F^{\mathcal{M}}(m_1, \ldots, m_l) := m_1$, and for every constant symbol c of $\mathcal{L}$, let $c^{\mathcal{M}_n} = 1$. The class of $\mathcal{L}$-structures is axiomatised by the empty set of $\mathcal{L}$-sentences. So by 5.3, the set of finite $\mathcal{L}$-structures is not axiomatisable. This gives a contradiction. Therefore ϕ does not axiomatise the class of infinite $\mathcal{L}$-structures. $\square$

Warning: For some languages $\mathcal{L}$, there exist consistent $\mathcal{L}$-sentences which only have infinite models. Such sentences often use the fact that every injective function from a finite set to itself is surjective (or that every surjective function from a finite set to itself is injective). See Exercise Sheet 8.

Remark 5.7. *The previous proposition is part of a general pattern of results. Let X be a class of $\mathcal{L}$-structures and let Y be the class of all $\mathcal{L}$-structures not in X. If X is axiomatised by a single $\mathcal{L}$-sentence ϕ then Y is axiomatised by $\neg\phi$. So to show that X is not axiomatised by a single sentence, we can show that Y is axiomatisable. For reasons beyond the scope of this course, this will always work.*

Here is a simple consequence of the Compactness Theorem that doesn't involve not being able to axiomatise classes of structures with finite unbounded size.

Exercise 5.8. *Let $\mathcal{L}_{gp} := \langle \cdot, (-)^{-1}, e\rangle$ be the language of groups. We say a group G is* ***torsion*** *if all its elements have finite order. You should know from 1st year group theory that all finite groups are torsion. There exist (interesting) infinite torsion groups. For instance, let $U(\mathbb{C})$ be the multiplicative group of complex roots 1, i.e., the set of $z \in \mathbb{C}$ such that $z^n = 1$ for some $n \in \mathbb{N}$. Show that the class of torsion groups is not axiomatisable.*

For more sophisticated applications of the Compactness Theorem, we need to use the method of "extensions by constants". For completeness, I give the general definition of the extension of a language.

Definition 5.9. *Let $\mathcal{L} := \langle \mathcal{R}, \mathcal{F}, \mathcal{C}\rangle$ and $\mathcal{L}^+ := \langle \mathcal{R}^+, \mathcal{F}^+, \mathcal{C}^+\rangle$ be languages where $\mathcal{R}, \mathcal{R}^+$ are relation symbols, $\mathcal{F}, \mathcal{F}^+$ are function symbols and $\mathcal{C}, \mathcal{C}^+$ are constant symbols.*

We say that $\mathcal{L}^+$ is an ***extension*** *of $\mathcal{L}$ and $\mathcal{L}$ is a* ***sublanguage*** *of $\mathcal{L}^+$ if $\mathcal{R} \subseteq \mathcal{R}^+$, $\mathcal{F} \subseteq \mathcal{F}^+$ and $\mathcal{C} \subseteq \mathcal{C}^+$ (note, any relation/function symbol of both $\mathcal{L}$ and $\mathcal{L}^+$ should have the same arity).*

If $\mathcal{L}^+$ is an extension of $\mathcal{L}$ with $\mathcal{R}^+ = \mathcal{R}$ and $\mathcal{F}^+ = \mathcal{F}$ then we call $\mathcal{L}^+$ an ***extension by constants*** *of $\mathcal{L}$. In this case we write $\mathcal{L}^+ = \mathcal{L}(\mathcal{D})$ where $\mathcal{D} = \mathcal{C}^+ \backslash \mathcal{C}$.*

Definition 5.10. *If $\mathcal{M}^+$ is an $\mathcal{L}^+$-structure then there is a unique $\mathcal{L}$-structure $\mathcal{M}^+$ with the same domain as $\mathcal{M}$ and with $R^{\mathcal{M}} = R^{\mathcal{M}^+}$ for all relation symbols R of $\mathcal{L}$, $F^{\mathcal{M}} = F^{\mathcal{M}^+}$ for all function symbol F of $\mathcal{L}$ and $c^{\mathcal{M}} = c^{\mathcal{M}^+}$ for all constant symbol c of $\mathcal{L}$. The structure $\mathcal{M}$ is called the **restriction of $\mathcal{M}^+$ to $\mathcal{L}$** and $\mathcal{M}^+$ is called an **expansion of $\mathcal{M}$ to $\mathcal{L}^+$**. The structure $\mathcal{M}$ is also called a **reduct of $\mathcal{M}^+$**.*

Remark 5.11. *Let $\mathcal{L}^+$ be an extension of $\mathcal{L}$. All $\mathcal{L}$-terms (resp. $\mathcal{L}$-formulas) are $\mathcal{L}^+$-terms (resp. $\mathcal{L}^+$-formulas) in which only elements of the alphabet of $\mathcal{L}$ occur.*

This proposition could be proved by induction on the complexity of terms and formulas.

Proposition 5.12. *Let $\mathcal{L}^+$ be an extension of the language $\mathcal{L}$. If $\mathcal{M}$ is the restriction of the $\mathcal{L}^+$-structure $\mathcal{M}^+$ to $\mathcal{L}$, $t \in tm(\mathcal{L})$, $\phi \in Fml(\mathcal{L})$ and $h : Vbl \to M$ where M is the domain of $\mathcal{M}$ (hence h is an assignment of $\mathcal{M}$ and of $\mathcal{M}^+$), then*

(i) $t^{\mathcal{M}}[h] = t^{\mathcal{M}^+}[h]$ *and*

(ii) $\mathcal{M} \models \phi[h] \iff \mathcal{M}^+ \models \phi[h]$.

We now give some examples of using extensions by constants in applications of the Compactness Theorem.

For our first applications we need an important definition.

Definition 5.13. *Let $\mathcal{L}$ be a language and let $\mathcal{M}_1, \mathcal{M}_2$ be $\mathcal{L}$-structures. We say that $\mathcal{M}_1$ and $\mathcal{M}_2$ are **elementary equivalent** and write $\mathcal{M}_1 \equiv \mathcal{M}_2$ if for all $\mathcal{L}$-sentences ϕ, $\mathcal{M}_1 \models \phi$ if and only if $\mathcal{M}_2 \models \phi$.*

This is a very weak version of the Upwards Löwenheim-Skolem Theorem which will be covered in the 4th year material.

Proposition 5.14. *Let $\mathcal{L}$ be a language and $\mathcal{M}$ be an infinite $\mathcal{L}$-structure. For all cardinalities κ, there is an $\mathcal{L}$-structure $\mathcal{M}'$ such that the domain of $\mathcal{M}'$ has cardinality greater than κ and $\mathcal{M} \equiv \mathcal{M}'$.*

Alternative formulation: *Let $\mathcal{L}$ be a language and $\mathcal{M}$ be an infinite $\mathcal{L}$-structure. For all sets κ, there is an $\mathcal{L}$-structure $\mathcal{M}'$ such that the domain of $\mathcal{M}'$ has size greater than κ and $\mathcal{M} \equiv \mathcal{M}'$.*

Proof. Let $\mathcal{L}^+$ be an extension of $\mathcal{L}$ with a constant symbol c_i for every $i \in \kappa$. Let Σ be the set of $\mathcal{L}$-sentences satisfied by $\mathcal{M}$. We show that

$$\Sigma \cup \{c_i \neq c_j \mid i \neq j\}$$

has a model.

For each $X \subseteq \kappa$, define $\mathcal{M}_X$ to be an expansion of $\mathcal{M}$ to an $\mathcal{L}^+$-structure by picking $c_i^{\mathcal{M}_X} \in \mathcal{M}$ so that $c_i^{\mathcal{M}_X} \neq c_j^{\mathcal{M}_X}$ for all $i, j \in X$ with $i \neq j$ and picking $c_i^{\mathcal{M}_X} \in \mathcal{M}$ for $i \in \kappa \backslash X$ arbitrarily.

Then $\mathcal{M}_X$ is a model of

$$\Sigma \cup \{c_i \neq c_j \mid i, j \in X \text{ and } i \neq j\}.$$

It follows that every finite subset of $\Sigma \cup \{c_i \neq c_j \mid i \neq j\}$ has a model. So, by the Compactness Theorem $\Sigma \cup \{c_i \neq c_j \mid i \neq j\}$ has a model $\mathcal{M}_\kappa$. Since $c_i^{\mathcal{M}_\kappa} \neq c_j^{\mathcal{M}_\kappa}$ for all $i \neq j$ in κ, $\mathcal{M}_\kappa$ has size at least κ. $\square$

Definition 5.15.

1. *A set X together with a binary relation $\leq$ is a total order if for all $a, b, c \in X$,*

(i) $a \leq b$ and $b \leq c$ implies $a \leq c$;

(ii) $a \leq a$;

(iii) $a \leq b$ and $b \leq a$ implies $a = b$; and

(iv) $a \leq b$ or $b \leq a$.

2. *An **ordered ring** is a ring R together with a total order $\leq$ which satisfies the following properties:*

   (i) For all $a, b, c \in R$, if $b \leq c$ then $a + b \leq a + c$.

   (ii) For all $a, b, c \in R$, if $0 \leq a$ and $b \leq c$ then $ab \leq ac$.

3. *The language of ordered rings is $\mathcal{L}_{\text{o-ring}} := \langle \geq, +, \cdot, -, 0, 1 \rangle$. where $\geq$ is a binary relation symbol, $+$ and $\cdot$ are binary function symbols, $-$ is a unary function symbol and 0 and 1 are constant symbols.*

Note that the class of ordered rings is axiomatisable in $\mathcal{L}_{\text{o-ring}}$.

You already know lots of examples of ordered rings. For example, $\mathbb{Z}$, $\mathbb{R}$ and $\mathbb{Q}$ are all ordered rings when equipped with their usual order. Any ordered ring can be viewed as an $\mathcal{L}_{\text{o-ring}}$-structure in the obvious way. If R is an ordered ring then for all $n \in \mathbb{N}$,

$$\underbrace{1 + \ldots + 1}_{\text{n times}} \neq 0.$$

A (positive) **infinitesimal element** of an ordered ring R is an element $\epsilon \in R$ such that for all $n \in \mathbb{N}$, $0 < n\epsilon \leq 1$. If R is a field then we may rewrite this as $0 < \epsilon \leq 1/n$ for all $n \in \mathbb{N}$.

Example 5.16. *Let $\mathcal{L}_{\text{o-ring}} := \langle \geq, +, \cdot, -, 0, 1 \rangle$. There exists an ordered ring which has infinitesimal elements and is elementary equivalent to $\mathbb{R}$ as an ordered ring.*

Let $\mathcal{L}_{\text{o-ring}}(\epsilon)$ be the extension of $\mathcal{L}_{\text{o-ring}}$ by the constant symbol ϵ. We want to write down a $\mathcal{L}_{\text{o-ring}}(\epsilon)$-sentence which, for $n \in \mathbb{N}$, says that $0 < n\epsilon \leq 1$. Technically "n" is not part of our language. For each $n \in \mathbb{N}$, we define an $\mathcal{L}_{\text{o-ring}}$-term t_n by induction. Let $t_1 := 1$. For all $n \in \mathbb{N}$, let $t_{n+1} := (t_n + 1)$. Note that this means that for all $\mathcal{L}_{\text{o-ring}}$-structures $\mathcal{A}$,

$$t_n^{\mathcal{A}} = \underbrace{1^{\mathcal{A}} + \ldots + 1^{\mathcal{A}}}_{n \text{ times}}.$$

For each $n \in \mathbb{N}$, let ϕ_n be the $\mathcal{L}_{\text{o-ring}}(\epsilon)$-sentence

$$0 < \epsilon \wedge t_n \cdot \epsilon \leq 1.$$

Let T be the set of $\mathcal{L}_{\text{o-ring}}$-sentences satisfied by $\mathbb{R}$. We will show that the set of $\mathcal{L}_{\text{o-ring}}(\epsilon)$-sentences $T \cup \{\phi_n \mid n \in \mathbb{N}\}$ has a model. Let $X \subseteq T \cup \{\phi_n \mid n \in \mathbb{N}\}$ be finite. Let $m \in \mathbb{N}$ be greatest such that $\phi_m \in X$. Let $\mathcal{M}_X$ be the $\mathcal{L}_{\text{o-ring}}(\epsilon)$-structure which is an expansion of $\mathbb{R}$ as an $\mathcal{L}_{\text{o-ring}}$-structure to an $\mathcal{L}_{\text{o-ring}}(\epsilon)$-structure by setting $\epsilon^{\mathcal{M}_X} := 1/(m+1)$. Then $t_m \cdot \epsilon^{\mathcal{M}_X} = m/m+1$ and $0 < m/m+1 \leq 1$. So $\mathcal{M}_X$ is a model of X. Thus, by the Compactness Theorem, there exists an $\mathcal{L}_{\text{o-ring}}(\epsilon)$-structure $\mathcal{M}$ which satisfies $T \cup \{\phi_n \mid n \in \mathbb{N}\}$. Since $\mathcal{M} \models T$, the $\mathcal{L}$-reduct of $\mathcal{M}$ is elementary equivalent to $\mathbb{R}$ as an ordered ring. For all $n \in \mathbb{N}$, $\mathcal{M} \models \phi_n$. Therefore $\epsilon^{\mathcal{M}}$ is an infinitesimal. Therefore the $\mathcal{L}$-reduct of $\mathcal{M}$ has the required properties.

The next example we give is also to ordered structures. This time we use a strict order.

Example 5.17. *Let $\mathcal{L} := \langle < \rangle$ where $<$ is a binary relation symbol. For each $n \in \mathbb{N}$, let $\mathcal{A}_n$ be the $\mathcal{L}$-structure with domain $\{1, \dots, n\}$ with $<^{\mathcal{A}_n}$ defined as the usual strict order on $\{1, \dots, n\}$. Let T be the set of $\mathcal{L}$-sentences satisfied by all $\mathcal{A}_n$. There is an $\mathcal{L}$-structure $\mathcal{A}$ with domain A such that $\mathcal{A} \models T$ and for all $r \in \mathbb{R}$, there exists $a_r \in A$ such that for all $r, s \in \mathbb{R}$,*

$$a_r < a_s \text{ if and only if } r < s \text{ in } \mathbb{R}$$

To show this we extend our language by $\mathbb{R}$-many constant symbols. Let $\mathcal{L}_{\mathbb{R}} := \langle <, (c_r)_{r \in \mathbb{R}} \rangle$ where $<$ is the binary relation symbol from $\mathcal{L}$ and c_r is a constant symbol for all $r \in \mathbb{R}$. For each pair of rational numbers r, s, let $\phi_{r<s}$ be the $\mathcal{L}_{\mathbb{R}}$-formula $c_r < c_s$.

For $X \subseteq \mathbb{R}$, let

$$\Sigma_X := \{\phi_{r<s} \mid r, s \in X \text{ and } r < s\} \cup \{(\neg \phi_{r<s}) \mid r, s \in X \text{ and } r \not< s\}.$$

We will show that for all finite $Y \subseteq \mathbb{R}$, $\Sigma_Y \cup T$ has a model. Let $Y = \{r_1, \dots, r_n\}$ with $r_1 < r_2 < \dots < r_n$. We expand $\mathcal{A}_n$ to an $\mathcal{L}_{\mathbb{R}}$- structure $\mathcal{A}_{n,Y}$ which is a model of Σ_Y. For each r_i, let $c_{r_i}^{\mathcal{A}_{n,Y}}$ be i and for $r \notin Y$, let $c_r^{\mathcal{A}_{n,Y}} = 1$. This means that for all $r, s \in Y$, $r < s$ if and only if $c_r <^{\mathcal{A}_n} c_s$. So $\mathcal{A}_n$, as an $\mathcal{L}_{\mathbb{R}}$-structure, is a model of Σ_Y. Finally, $\mathcal{A}_{n,Y}$, as an $\mathcal{L}_{\mathbb{R}}$-structure, is a model of T because, by definition of T, $\mathcal{A}_{n,Y}$ is $\mathcal{A}_n$ as an $\mathcal{L}$-structure.

If $S \subseteq \Sigma_{\mathbb{R}} \cup T$ is finite then there exists a finite subset $Y \subseteq \mathbb{R}$ such that $S \subseteq \Sigma_Y \cup T$. So, for any such S, $\mathcal{A}_{n,Y}$ is a model of S. Therefore, by the Compactness Theorem, $\Sigma_{\mathbb{R}} \cup T$ has a model $\mathcal{A}$. Since $\mathcal{A} \models \Sigma_{\mathbb{R}}$,

$$c_r^{\mathcal{A}} <^{\mathcal{A}} c_s^{\mathcal{A}} \text{ if and only if } r < s.$$

So setting $a_r := c_r^{\mathcal{A}}$ and viewing $\mathcal{A}$ as an $\mathcal{L}$-structure, gives a model of T with the required properties.

6 Universal Algebra: Substructures and Homomorphisms

The material in this section will not appear on the 3rd year exam. However, 4th years need to know what a substructure is and what an isomorphism of $\mathcal{L}$-structures is.

In this section we give a general definitions of substructures and homomorphisms for $\mathcal{L}$-structures. These definitions will mostly match up with ones you already know if we have picked the correct language. Warning: This will not always be the case!

Definition 6.1. *Let $\mathcal{A}, \mathcal{B}$ be $\mathcal{L}$-structures with domains A and B respectively. A **map from $\mathcal{A}$ to $\mathcal{B}$** is a function $f : A \to B$. We will write $f : \mathcal{A} \to \mathcal{B}$ instead of $f : A \to B$.*

Definition 6.2. *Let $\mathcal{L}$ be a language and let $\mathcal{A}, \mathcal{B}$ be $\mathcal{L}$-structures with domains A and B respectively. A map $g : \mathcal{A} \to \mathcal{B}$ is a **homomorphism of $\mathcal{L}$-structures** if*

1. *for all relation symbols R of $\mathcal{L}$ of arity n and elements $a_1, \dots, a_n$, $(a_1, \dots, a_n) \in R^{\mathcal{A}}$ implies $(g(a_1), \dots, g(a_n)) \in R^{\mathcal{B}}$;*
2. *for all function symbols F of $\mathcal{L}$ of arity n and elements $a_1, \dots, a_n$,*

$$g(F^{\mathcal{A}}(a_1, \dots, a_n)) = F^{\mathcal{B}}(g(a_1), \dots, g(a_n));$$

and

3. *for all constant symbols c of $\mathcal{L}$, $g(c^{\mathcal{A}}) = c^{\mathcal{B}}$.*

*A homomorphism $g : \mathcal{A} \to \mathcal{B}$ is an **embedding** if g is injective and for all relation symbols R of arity n,*

$$(a_1, \dots, a_n) \in R^{\mathcal{A}} \text{ if and only if } (g(a_1), \dots, g(a_n)) \in R^{\mathcal{B}}.$$

*An embedding $g : \mathcal{A} \to \mathcal{B}$ is an **isomorphism** if it is bijective.*

Examples.

(i) Let $\mathcal{L}_\emptyset$ be the empty language. Any function $f : A \to B$ where A and B are the domains of $\mathcal{L}_\emptyset$-structures $\mathcal{A}$ and $\mathcal{B}$.

(ii) Let $\mathcal{L}_{gp} := \langle \cdot, (-)^{-1}, e \rangle$ be the language of groups. If $\mathcal{A}$ and $\mathcal{B}$ are $\mathcal{L}_{gp}$-structures which are groups then an $\mathcal{L}_{gp}$-homomorphism $g : \mathcal{A} \to \mathcal{B}$ is just a group homomorphism in the usual sense. Note that this requires a proof and the same statement is true if we instead consider groups in the language with just the binary operation "$\cdot$".

Remark. *A homomorphism $g : \mathcal{A} \to \mathcal{B}$ of $\mathcal{L}$-structures is an isomorphism if and only if there is a homomorphism $f : \mathcal{B} \to \mathcal{A}$ such that $fg = Id_{\mathcal{A}}$ and $gf = Id_{\mathcal{B}}$.*

Definition 6.3. *Let $\mathcal{L}$ be a language and let $\mathcal{A}, \mathcal{B}$ be $\mathcal{L}$-structures with domains A and B respectively. If $A \subseteq B$ and the inclusion map is an embedding of $\mathcal{L}$-structures then we call $\mathcal{A}$ a substructure of $\mathcal{B}$.*

Examples. *1. Let $\mathcal{L}_\emptyset$ be the empty language. Let $\mathcal{B}$ be an $\mathcal{L}_\emptyset$-structure. Then any non-empty subset A is the domain of a substructure of $\mathcal{B}$.*

2. Let $\mathcal{L}_{gp}$ be the language of groups. If $\mathcal{B}$ is an $\mathcal{L}_{gp}$-structure which is a group then the $\mathcal{L}_{gp}$-substructures of $\mathcal{B}$ are exactly the subgroups of $\mathcal{B}$.

3. Let $\mathcal{L} := \langle \cdot \rangle$. If $\mathcal{B}$ is an $\mathcal{L}$-structure which is a group under $\cdot^{\mathcal{B}}$ then the $\mathcal{L}$-substructures of $\mathcal{B}$ are not necessarily subgroups.

Remark 6.4. *Let $\mathcal{L}$ be a language and let $\mathcal{A}, \mathcal{B}$ be $\mathcal{L}$-structures with domains A and B respectively. Suppose that $A \subseteq B$. Then $\mathcal{A}$ is a **substructure** of $\mathcal{B}$ if and only if the following conditions hold:*

(i) For all relation symbols R of $\mathcal{L}$ of arity n and $a_1, \dots, a_n \in A$, $(a_1, \dots, a_n) \in R^{\mathcal{A}}$ if and only if $(a_1, \dots, a_n) \in R^{\mathcal{B}}$.

(ii) For all function symbols F of $\mathcal{L}$ of arity n and $a_1, \dots, a_n \in A$, $F^{\mathcal{A}}(a_1, \dots, a_n) = F^{\mathcal{B}}(a_1, \dots, a_n)$.

(iii) For all constant symbols c of $\mathcal{L}$, $c^{\mathcal{A}} = c^{\mathcal{B}}$.

This allows the following definition.

Definition 6.5. *Let $\mathcal{M}$ be an $\mathcal{L}$-structure with domain M. Let $A \subseteq M$ be such that for all function symbols F of $\mathcal{L}$ and $a_1, \dots, a_n \in A$, $F^{\mathcal{M}}(a, \dots, a_n) \in A$ and for all constant symbols c of $\mathcal{L}$, $c^{\mathcal{M}} \in A$. Then the **substructure of $\mathcal{M}$ induced on** A is the unique substructure $\mathcal{A}$ of $\mathcal{M}$ with domain A. It follows from the definition of substructure that*

(i) $R^{\mathcal{A}} = R^{\mathcal{M}} \cap A^n$ for all relation symbols R of arity n;

(ii) $F^{\mathcal{A}}(a_1, \dots, a_n) = F^{\mathcal{M}}(a_1, \dots, a_n)$ for all functions symbols F of arity n; and

(iii) $c^{\mathcal{A}} = c^{\mathcal{M}}$ for all constant symbols c.

7 Further Model Theory (4th year material)

I will just write skeleton notes for this section with reference to books. This is possible because the material is no longer so sensitive to presentation. On the other hand, do let me know if you don't understand something in one of the books, in particular, if you don't understand some notation or terminology. I very strongly suggest you try the exercises on Sheet 10 related to this section.

7.1 Definable Sets

Definition 7.1. *Let $\mathcal{L}$ be a language and $\mathcal{A}$ an $\mathcal{L}$-structure with domain A. We say a subset $S \subseteq A^n$ for some $n \in \mathbb{N}$ is **definable** if there exists an $\mathcal{L}$-formula $\phi(v_1, \dots, v_n)$ such that for all $(a_1, \dots, a_n) \in A^n$*

$$\mathcal{A} \models \phi[a_1, \dots, a_n] \text{ if and only if } (a_1, \dots, a_n) \in S.$$

Examples 7.2. *(i) If $\mathcal{L}_{gp}$ is the language of groups and $\mathcal{G}$ is an $\mathcal{L}_{gp}$-structure which is a group then the centre $Z(\mathcal{G})$ of $\mathcal{G}$ is defined by the formula $\forall v_2 \, v_1 \cdot v_2 = v_2 \cdot v_1$.*

(ii)

Definition 7.3. *Let $\mathcal{L}$ be a language and $\mathcal{A}, \mathcal{B}$ an $\mathcal{L}$-structures with domain A and B respectively. An **embedding** of $\mathcal{A}$ in $\mathcal{B}$ is an injective function $\alpha : A \to B$ such that*

(i) for all relation symbols R of $\mathcal{L}$ and $(a_1, \dots, a_n) \in A^n$,

$$(a_1, \dots, a_n) \in R^{\mathcal{A}} \text{ if and only if } (\alpha(a_1), \dots, \alpha(a_n)) \in R^{\mathcal{B}};$$

(ii) for all function symbols F of $\mathcal{L}$ and $(a_1, \dots, a_n) \in A^n$

$$\alpha(F^{\mathcal{A}}(a_1, \dots, a_n)) = F^{\mathcal{B}}(\alpha(a_1), \dots, \alpha(a_n));$$

and

(iii) for all constant symbols c of $\mathcal{L}$, $\alpha(c^{\mathcal{A}}) = c^{\mathcal{B}}$.

Definition 7.4. *Let $\mathcal{L}$ be a language and $\mathcal{A}$ an $\mathcal{L}$-structure. An **automorphism** of $\mathcal{A}$ is a bijective embedding $\alpha : \mathcal{A} \to \mathcal{A}$.*

The next theorem shows that "automorphisms preserve definable sets".

Theorem 7.5. *Let $\mathcal{L}$ be a language and $\mathcal{A}$ an $\mathcal{L}$-structure. Let $\alpha : \mathcal{A} \to \mathcal{A}$ be an automorphism. For all $\mathcal{L}$-formulas $\phi(x_1, \dots, x_n)$ and $(a_1, \dots, a_n) \in A^n$,*

$$\mathcal{A} \models \phi[a_1, \dots, a_n] \text{ if and only if } \mathcal{A} \models \phi[\alpha(a_1), \dots, \alpha(a_n)].$$

For examples using this theorem to show that a set is not definable in an $\mathcal{L}$-structure see Section 4.2 of "An invitation to model theory" by Jonathan Kirby.

7.2 Elementary Substructures

Definition 7.6. *A map $f : \mathcal{A} \to \mathcal{B}$ is an* ***elementary embedding*** *if for all $\mathcal{L}$-formulas $\phi(x_1, \ldots, x_n)$ and $a_1, \ldots, a_n \in A$, $\mathcal{A} \models \phi[a_1, \ldots, a_n]$ if and only if $\mathcal{B} \models \phi[f(a_1), \ldots, f(a_n)]$.*

If $A \subseteq B$ and the inclusion map is an elementary embedding then we call $\mathcal{A}$ an ***elementary substructure*** *of $\mathcal{B}$ and $\mathcal{B}$ an* ***elementary extension*** *of $\mathcal{A}$. We write $\mathcal{A} \preceq \mathcal{B}$ to mean $\mathcal{A}$ is an elementary substructure of $\mathcal{B}$.*

Remark 7.7. *If $\mathcal{A}$ is an elementary substructure of $\mathcal{B}$ then $\mathcal{A} \equiv \mathcal{B}$.*

Proposition 7.8. *Isomorphisms are elementary embeddings. In particular, isomorphic structures are elementary equivalent.*

Proof. This is proved by induction on the complexity of formulas. Note this was previously stated for automorphisms. □

Lemma 7.9. *If $\mathcal{A}$ is an elementary substructure of $\mathcal{B}$ then $\mathcal{A}$ is a substructure of $\mathcal{B}$.*

Proof. This is exercise 5 of sheet 10. □

Theorem 7.10 (Tarski-Vaught Test). *Let $\mathcal{M}$ be an $\mathcal{L}$-structure with domain M and let $A \subseteq M$. The following are equivalent:*

(i) A is the domain of an elementary substructure of $\mathcal{M}$. In this situation we will often say A is an elementary substructure of $\mathcal{M}$.

(ii) For every $\mathcal{L}$-formula $\phi(x, y_1, \ldots, y_n)$ and $a_1, \ldots, a_n \in A$, if $\mathcal{M} \models (\exists x \phi)[a_1, \ldots, a_n]$ then there is some $b \in A$ such that $\mathcal{M} \models \phi[b, a_1, \ldots, a_n]$.

Proof. (i)⇒(ii) Suppose that $a_1, \ldots, a_n \in A$ and $\mathcal{M} \models (\exists x \phi)[a_1, \ldots, a_n]$. Then by (i), $\mathcal{A} \models (\exists x \phi)[a_1, \ldots, a_n]$. By Tarski's Truth Definition, this implies there exists $b \in A$ such that $\mathcal{A} \models \phi[b, a_1, \ldots, a_n]$ as required.

(ii)⇒(i) The first thing we need to show is that A is the domain of a substructure. Let c be a constant symbol of $\mathcal{L}$ and take ϕ to be the formula $c = v_1$. Then $\mathcal{M} \models \exists v_1 \phi$. So, by (ii), $\mathcal{M} \models \phi[b]$ for some $b \in \mathcal{A}$. Therefore $c^{\mathcal{M}} = b$. So $c^{\mathcal{M}} \in A$. Now suppose that F is an n-ary function symbol and $a_1, \ldots, a_n \in A$. Let $\phi(x, y_1, \ldots, y_n)$ be the formula, $F(y_1, \ldots, y_n) = x$. Then $\mathcal{M} \models (\exists x \phi)[a_1, \ldots, a_n]$. So, by (ii), $\mathcal{A} \models \phi[b, a_1, \ldots, a_n]$ for some $b \in A$. Therefore $F^{\mathcal{M}}(a_1, \ldots, a_n) = b$. Thus A is the domain of a substructure of $\mathcal{M}$.

We now need to show that this substructure is elementary. We show, by induction on the complexity of the $\mathcal{L}$-formula $\phi(y_1, \ldots, y_n)$, that for all $a_1, \ldots, a_n \in A$,

$$\mathcal{M} \models \phi[a_1, \ldots, a_n] \text{ if and only if } \mathcal{A} \models \phi[a_1, \ldots, a_n]. \qquad (\dagger)$$

Suppose ϕ is complexity 0. The case where ϕ is $\perp$ is trivial. Suppose ϕ is of the form $t_1 = t_2$ where t_1 and t_2 are terms. It follows by induction on the complexity of terms, just using the fact that $\mathcal{A}$ is a substructure of $\mathcal{M}$, that if h is a valuation with $h(y_i) := a_i$ and $h(z) \in A$ for all variables $z \in \mathrm{Vbl}$ that $t_1^{\mathcal{M}}[h] = t_2^{\mathcal{M}}[h]$ if and only if $t_1^{\mathcal{A}}[h] = t_2^{\mathcal{A}}[h]$. Thus $\mathcal{M} \models t_1 = t_2[a_1, \ldots, a_n]$ if and only if $\mathcal{A} \models t_1 = t_2[a_1, \ldots, a_n]$. The case of atomic formulas of the form $R(t_1, \ldots, t_n)$ is similar.

We now prove the inductive step. If the equivalence holds for ϕ_1 and ϕ_2 then it is easy to show that the equivalence (†) holds for $(\phi_1 \to \phi_2)$. This implies that if the equivalence (†) holds for ϕ then it holds for $\neg\phi$. It's easy to check that for any formula ϕ, $\neg\exists x\,\neg\phi \equiv \forall x\,\phi$. Thus in order to show that the equivalence (†) is true of $\forall x\,\phi$, we just need to show it is true for $\neg\exists x\,\neg\phi \equiv \forall x\,\phi$.

Suppose that (†) holds for $\phi(x, y_1, \ldots, y_n)$. By the discussion above, (†) holds of $\neg\phi$. Suppose $\mathcal{M} \models (\exists x\neg\phi)[a_1, \ldots, a_n]$ for some $a_1, \ldots, a_n \in A$. Then, by (ii), $\mathcal{M} \models \neg\phi[b, a_1, \ldots, a_n]$ for some $b \in A$. Since (†) is true for $\neg\phi$, $\mathcal{A} \models \neg\phi[b, a_1, \ldots, a_n]$. Thus, by Tarski's Truth Definition, $\mathcal{A} \models (\exists x\neg\phi)[a_1, \ldots, a_n]$. Thus the forward direction of (†) is true for $\exists x\neg\phi$. Now suppose that $\mathcal{A} \models \exists x\neg\phi[a_1, \ldots, a_n]$ for $a_1, \ldots, a_n \in A$. By Tarski's Truth Definition, there exists $b \in A$ such that $\mathcal{A} \models \neg\phi[b, a_1, \ldots, a_n]$. So, by (†), $\mathcal{M} \models \neg\phi[b, a_1, \ldots, a_n]$. So, applying Tarski's Truth Definition again, we get that $\mathcal{M} \models (\exists x\neg\phi)[a_1, \ldots, a_n]$. Therefore, (†) holds of $(\exists x\neg\phi)$. But then, by the discussion above, (†) holds for $\neg\exists x\neg\phi$. Hence (†) holds for $\forall x\,\phi$.

Thus, by induction on complexity of formulas, (†) holds for all formulas ϕ. This implies (i). □

7.3 Downwards Löwenheim-Skolem Theorem

Definition 7.11. *The* ***cardinality of a language*** *$\mathcal{L}$, written $card(\mathcal{L})$, is the cardinality of $\mathcal{R} \cup \mathcal{F} \cup \mathcal{C}$ where $\mathcal{R}$ is the set of relation symbols of $\mathcal{L}$, $\mathcal{F}$ is the set of function symbols of $\mathcal{L}$ and $\mathcal{C}$ is the set of constant symbols of $\mathcal{L}$.*

Theorem 7.12 (The Downwards Löwenheim-Skolem Theorem). *Let $\mathcal{M}$ be an $\mathcal{L}$-structure with domain M and let $A \subseteq M$. Then there is an elementary substructure $\mathcal{N}$ with domain N of $\mathcal{M}$ such that $A \subseteq N$ and $card(\mathcal{N}) \leq \max\{\aleph_0, card(A), card(\mathcal{L})\}$.*

For more details see "An invitation to Model Theory" section 12.3.

7.4 Upwards Löwenheim-Skolem Theorem

Theorem 7.13 (The Upwards Löwenheim-Skolem Theorem). *Let $\mathcal{M}$ be an infinite $\mathcal{L}$-structure and let κ be a cardinal of size greater than or equal to $card(\mathcal{M})$ and $card(\mathcal{L})$. Then there is an elementary extension $\mathcal{N} \succeq \mathcal{M}$ of cardinality κ.*

For more details see "An invitation to Model Theory" section 13.2.

7.5 Categoricity

Definition 7.14. *Recall that an $\mathcal{L}$-**theory** is a consistent set of $\mathcal{L}$-sentences. If X is a class of $\mathcal{L}$-structures then **the theory of** X is the set of all $\mathcal{L}$-sentences satisfied by all $\mathcal{L}$-structures in X. The $\mathcal{L}$-theory of a single $\mathcal{L}$-structure is the set of $\mathcal{L}$-sentences satisfied by that $\mathcal{L}$-structure.*

Definition 7.15. *(i) Let κ be an infinite cardinal. A theory T is κ-**categorical** if all models of T of cardinality κ are isomorphic.*

*(ii) A theory T is **categorical** if for some infinite cardinal $\kappa \geq card(\mathcal{L})$, all models of T of cardinality κ are isomorphic.*

Example. *Let $\mathcal{L}_\emptyset$ be the empty language and let $T_\emptyset$ be the empty $\mathcal{L}_\emptyset$-theory. If $\mathcal{A}$ and $\mathcal{B}$ are $\mathcal{L}_\emptyset$-structures of the same cardinality then, by definition, there is a bijection $f : A \to B$. Then f is an isomorphism of $\mathcal{L}_\emptyset$-structures. Therefore $T_\emptyset$ is κ-categorical for every infinite cardinal κ.*

Theorem 7.16 (Categoricity Theorem). *Let T be a theory without finite models. If T is categorical then all models of T are elementary equivalent. In particular, if T is categorical and deductively closed (i.e. for all $\mathcal{L}$-sentences ϕ, $T \vdash \phi$ implies $\phi \in T$) then T is complete.*

Proof. Suppose T is categorical. Let κ be such that $\kappa \geq \text{card}(\mathcal{L})$ and all models of T of size κ are isomorphic. Let $\mathcal{M}$ and $\mathcal{N}$ be models of T. By the LS Theorems, any, by assumption infinite, model of T either has an elementary extension of cardinality κ or an elementary restriction of cardinality κ. Therefore there are models $\mathcal{M}'$ and $\mathcal{N}'$ of cardinality κ such that $\mathcal{M} \equiv \mathcal{M}'$ and $\mathcal{N} \equiv \mathcal{N}'$. By assumption $\mathcal{N}'$ is isomorphic to $\mathcal{M}'$ and hence $\mathcal{N} \equiv \mathcal{N}' \equiv \mathcal{M}' \equiv \mathcal{M}$.

□

For important examples of (κ-)categorical theories, see your exercise sheet (and Jonathan Kirby's book "An Invitation to Model Theory".)