Notes, Definitions and Comments on
Logic for Prelims
for students in their first year before 2008/2009

NOTES ON HODGES'S LOGIC

Volker Halbach
New College, Oxford

version of
24th July, 2008

This text is to be used only by candidates sitting Moderations in their second year, that is, Literae Humaniores students, who have studied logic from Hodges's *Logic*. This text is not be used by candidates who are in their first year in 2008/2009.

CONTENT

Preface

In these notes I aim to collect essential definitions and conventions as they are used in (Hodges, 2001). I have tried to stick as closely as possible to this book. I have also included material that goes slightly beyond Hodges' book, but which is usually treated in lectures and classes. In this case I have made an attempt to stick to Hodges' general approach and to follow widely accepted conventions.

In the margin I have given numbers such has 'H22' that refer to the corresponding page in (Hodges, 2001).

I am indebted to Stephen Blamey, Dave Leal and Hugh Rice for explaining to me Hodges' terminology and for suggesting numerous clarifications and improvements. I also thank Stephen Blamey for making the tableaux proofs more bearable by helping me with the macros.

Sebastian Sequoiah-Grayson (and some others) spotted several mistakes. I thank them for the corrections.

This pdf file contains internal links. A click on text in a red or green frame will send you to the corresponding definition or reference. These features get lost when the Manual is printed on paper.

1 Preliminaries

1.1 Sets

A set is a collection of arbitrary objects. Sets are identical if and only if they have the same members. Therefore the set of all animals having kidneys and the set of all animals having a heart are identical, because exactly those animals that have kidneys also have a heart and vice versa.[1] In contrast, the property of having a heart is distinguished from he property of having kidneys. Here we are only interested in sets, but not in properties.

The objects belonging to a set are its elements. $a \in M$ expresses that a is an element of the set M. If a is an element of M, one also says that a is in M.

There is only one set that contains no elements, namely the empty set $\emptyset$.

There are various ways to denote special sets. The set {New College, Merton College} has exactly the two colleges as its elements. The set {Merton College, New College} has the same elements. Therefore the sets are identical, that is, we have:

$$\{\text{New College, Merton College}\} = \{\text{Merton College, New College}\}$$

Thus if a set is specified by including names for the elements in curly brackets, the order of the names does not matter. The set {New College, Merton College, St. Mary of Winchester College} is again the same set because 'St. Mary of Winchester' is just another name for New College, and the set has therefore the same elements as {New College, Merton College}.

Above I have been talking about the set of all animals with a heart. This can be written more formally as:

$$\{x : x \text{ is an animal with a heart}\}$$

1.2 Ordered pairs and relations

As pointed out above, the elements of a set are not ordered by the set. {Tony Blair, George W. Bush} is the same set as´{George W. Bush, Tony Blair}. Ordered pairs, in contrast, have an order on their components. One writes

[1] I have added this footnote because there are regularly protests with respect to this example. For the example only complete and healthy animals are considered. I was told that planarians are an exception, so we would have to exclude them for the sake of the example.

⟨George W. Bush, Tony Blair ⟩ for the ordered pair with George W. Bush as the first and Tony Blair as the second component. ⟨George W. Bush, Tony Blair ⟩ and ⟨Tony Blair, George W. Bush⟩ are different ordered pairs, because the former has George W. Bush as first component, the latter Tony Blair (the second components are also different).

A set is a binary relation if and only if it contains only ordered pairs.[2] In particular, the empty set Ø is a relation, because it does not contain anything but ordered pairs.

The relation that is satisfied by objects x and y if and only if x is smaller than y is the following set:

> {⟨Munich, London ⟩, ⟨ Oxford, London⟩, ⟨Oxford, Munich⟩, ⟨Munich, Paris⟩,. . . }

Here and in the following I'll use 'iff' as an abbreviation for 'if and only if'. In the following definition $\mathscr{D}$ is an arbitrary set. $\mathscr{D}$ may be empty.

Definition 1.1. *A relation R is* H146

(i) *reflexive on $\mathscr{D}$ iff for all a in $\mathscr{D}$ $\langle a,a\rangle \in R$.*

(ii) *irreflexive iff for no a $\langle a,a\rangle \in R$.*

(iii) *non-reflexive iff R is neither reflexive nor irreflexive.* H147

(iv) *symmetric iff for all a,b: if $\langle a,b\rangle \in R$ then $\langle b,a\rangle \in R$.*

(v) *asymmetric iff for no a,b: $\langle a,b\rangle \in R$ and $\langle b,a\rangle \in R$.*

(vi) *non-symmetric iff R is neither symmetric nor asymmetric.* H149

(vii) *transitive iff for all a,b,c: if $\langle a,b\rangle \in R$ and $\langle b,c\rangle \in R$, then $\langle a,c\rangle \in R$.*

(viii) *intransitive iff for all a,b,c: if $\langle a,b\rangle \in R$ and $\langle b,c\rangle \in R$, then not $\langle a,c\rangle \in R$.*

(ix) *non-transitive iff R is neither transitive nor intransitive.* H150

(x) *connected on $\mathscr{D}$ iff for all a,b in $\mathscr{D}$ either $\langle a,b\rangle \in R$ or $\langle b,a\rangle \in R$ or $a = b$.* H155

(xi) *an equivalence relation on $\mathscr{D}$ iff it is reflexive on $\mathscr{D}$, symmetric and transitive.*

[2] There are also other notions of relations. According to another use of the term binary relations relate to sets of ordered pairs in the same way as properties are related to sets. Here a relation is identified with the corresponding set of ordered pairs. This is common practice in mathematics.

Note. Hodges (2001) does not refer explicitly to the domain in his definition of reflexivity. If the domain is empty, for instance, the empty relation Ø is reflexive; if the domain is not empty, then the empty relation Ø is *not* reflexive, because by the definition on p. 146 *every* dot must have a loop. Here I have made the reference to the underlying set explicit. The set $\mathcal{D}$ is Hodges' domain. The definition of connectedness suffers from a similar problem.

Definition 1.2. *A relation is a* function *iff for all* a, b, c*: if* $\langle a, b\rangle \in R$ *and* $\langle a, c\rangle \in R$ *then* $b = c$.

For instance the relation that is satisfied by those pairs $\langle a, b\rangle$ such that b is the mother of a is a function, because nobody has two (different) mothers.

There are also three-place relations. These are sets of triples $\langle a, b, c\rangle$. Four-place relations are sets of quadruples and so on. Unary (one-place) relations are simply sets.

1.3 Arguments, validity and contradiction

In logic usually sentences are the objects that can be true or false. Of course not every sentence of English can be true: a question like 'Are there any crisps with fish flavour?' is neither true nor false. The following focuses on declarative sentences, that is, sentences that can be true or false.

A sentence can be true in one possible situation and false in others.

An argument consists in premisses and one conclusion. Premisses and conclusion are declarative sentences. The following is an example of an argument:

> There is no German who does not like crisps with paprika flavour.
> Rebecca and Johannes are German.
> *Therefore* Johannes likes crisps with paprika flavour.

The two sentences 'There is no German who does not like crisps with paprika flavour.' and 'Rebecca and Johannes are German.' are the premisses of the argument, while 'Johannes likes crisps with paprika flavour.' is its conclusion. Usually the conclusion is marked by a phrase like 'therefore' or 'it follows that', but it need not be. Sometimes the conclusion precedes the premisses:

> Johannes likes crisps with paprika flavour. *For* there is no German who does not like crisps with paprika flavour, and Rebecca and Johannes are German.

An argument may feature just one premiss or, as a degenerate case, no premiss at all. H38

An argument is valid if and only if there is no possible situation in which all the premisses of the argument are true and the conclusion is false. The arguments above are valid.

An argument is invalid if and only if there is at least one possible situation where all the premisses are true and the conclusion is false.

An argument with no premisses will be valid if and only if the conclusion is true in all possible situations. A sentence is a necessary truth if and only if it is true in all possible situations.

A sentence is consistent if and only if it is true in at least one possible situation. A set of sentences is consistent if there is at least one possible situation where all its elements are true. H1

A sentence is a contradiction, inconsistent or self-contradictory if and only if there is no possible situation where it is true. A set of sentences is inconsistent if and only if there is no possible situation in which all its elements are true.

A sentence is contingent if and only if it is true in at least one possible situation and false in at least one possible situation.

2 Propositional logic

2.1 The syntax of the language of propositional logic

H97

This exposition deviates from section 26 in (Hodges, 2001) in taking 'P', 'Q' and so on as sentence letters right from the beginning instead of 'P_0', 'P_{00}',... Hodges says later on that he uses 'P', 'Q',... instead of 'P_0', 'P_{00}',... [1].

Definition 2.1 (sentence letters)**.** *'P', 'Q', 'R', 'S', 'T', 'P_1', 'Q_1', 'R_1', 'S_1', 'T_1', 'P_2' and so on are sentence letters (or, as some authors say, propositional variables, parameters or constants).*

Definition 2.2 (formula of $\mathscr{L}_1$)**.**

(i) All sentence letters are formulae of $\mathscr{L}_1$.

(ii) If ϕ and ψ are formulae of $\mathscr{L}_1$, then '$\neg\phi$', '$[\phi \wedge \psi]$', '$[\phi \vee \psi]$', '$[\phi \rightarrow \psi]$' and '$[\phi \leftrightarrow \psi]$' are formulae of $\mathscr{L}_1$.

Of course, nothing else is a formula. This could be added explicitly as a further condition in the definition, but I introduce the convention that the definition above and in the following have to be understood in the most restrictive way. That is, I ask the reader to add the clause 'and nothing else is a formula of $\mathscr{L}_1$.'

The Greek letters 'ϕ' and 'ψ' in (ii) are not expressions of the language $\mathscr{L}_1$. For instance, '$[\phi \wedge \psi]$' is not a formula of $\mathscr{L}_1$ and it only becomes a formula of $\mathscr{L}_1$ if 'ϕ' and 'ψ' are replaced by formulae of $\mathscr{L}_1$, respectively.

Example 2.3. By (i), 'P' is a formula of $\mathscr{L}_1$. Thus by (ii) '$\neg P$' is also a formula of $\mathscr{L}_1$. By (i) again 'T_4' is a formula of $\mathscr{L}_1$. By (ii) and what has been said so far, '$[\neg P \wedge T_4]$' is a formula, and by (ii) again also '$[[\neg P \wedge T_4] \rightarrow P]$' is a formula of $\mathscr{L}_1$.

Hodges (2001) calls the symbols '$\neg$', '$\wedge$', '$\vee$', '$\rightarrow$', '$\leftrightarrow$' truth-functor symbols. Most other authors call them connectives.

[1] For the use of quotation marks see Appendix C

name	in English	symbol	alternative symbols
conjunction	and	$\wedge$	., &
disjunction	or	$\vee$	+
negation	it is not the case that	$\neg$	-, ~
arrow (material implication, conditional)	if—then—	$\rightarrow$	$\supset$
biconditional (material equivalence)	if and only if	$\leftrightarrow$	$\equiv$

The names in brackets and the symbols in the right column are used by other authors; they will not be used in exams. But it is useful to know them when you are reading other authors.

The expressions in the 'in English' column indicate how the connectives are commonly read, rather than their precise meanings.

2.2 The semantics of propositional logic

H99ff

Definition 2.4. *A $\mathscr{L}_1$-structure is a function that assigns a truth value (T or F) to some sentence letters.*

Definition 2.5. *Let* **A** *be some structure that assigns either T or F to every sentence letter in ϕ and ψ.*

(i) If **A** *assigns T to a sentence letter, then the sentence letter is true in* **A**.

(ii) If ϕ is not true in **A**, *then '$\neg\phi$' is true in* **A**.

(iii) If ϕ and ψ are true in **A**, *then '$[\phi \wedge \psi]$' is true in* **A**.

(iv) If ϕ or ψ (or both) is true in **A**, *then '$[\phi \vee \psi]$' is true in* **A**.

(v) If ϕ is not true in **A** *or ψ is true in* **A**, *then '$[\phi \rightarrow \psi]$' is true in* **A**.

(vi) If ϕ and ψ are both true or if ϕ and ψ are both false, then '$[\phi \leftrightarrow \psi]$' is true in **A**.

(vii) If ϕ is not true in **A**, *then ϕ is false.*

The assumption that **A** assigns T or F to all sentence letters in ϕ is important. If this condition is not satisfied, ϕ is neither true nor false in **A**.

The definition can be summarised in truth tables. These tables allow one to calculate whether a sentence is true or false in a structure. For instance, the first line of the table for $\wedge$ tells you that '$[\phi \wedge \psi]$' is true in the structure iff ϕ is true and ψ is true in this structure.

ϕ	$\neg\phi$
T	F
F	T

ϕ	ψ	$[\phi \wedge \psi]$
T	T	T
T	F	F
F	T	F
F	F	F

ϕ	ψ	$[\phi \vee \psi]$
T	T	T
T	F	T
F	T	T
F	F	F

ϕ	ψ	$[\phi \rightarrow \psi]$
T	T	T
T	F	F
F	T	T
F	F	T

ϕ	ψ	$[\phi \leftrightarrow \psi]$
T	T	T
T	F	F
F	T	F
F	F	T

Definition 2.5 determines whether a formula is true or false in a structure, if the structure assigns T or F to every sentence letter in the sentence. I explain this by an example.

A structure **A** assigns T to the sentence letter 'P' and F to the sentence letter 'Q'; and it is to be determined whether the formula '$[\neg[P \rightarrow Q] \rightarrow [P \wedge Q]]$' is true in this structure or not.

Since **A** assigns T to 'P', 'P' is true in **A**; and since **A** assigns F to 'Q', 'Q' is false in **A** by Definition 2.5 (i). By Definition 2.5 (v), '$[P \rightarrow Q]$' is false in **A** and thus '$\neg[P \rightarrow Q]$' is true in **A** by Definition 2.5 (ii). By Definition 2.5 (ii) the formula '$[P \wedge Q]$' is false in **A**, and therefore the entire formula '$[\neg[P \rightarrow Q] \rightarrow [P \wedge Q]]$' is false in **A** according to Definition 2.5 (v) again.

This way of writing down the calculation is awkward. It becomes much more perspicuous if written down in the following way:

P	Q	$[\neg$	$[P$	$\rightarrow$	$Q]$	$\rightarrow$	$[P$	$\wedge$	$Q]]$
T	F	T	T	F	F	**F**	T	F	F

The boldface F is the final value.

One can calculate the truth and falsehood of a formula for all possible structures that assign T or F to all sentence letters of the formula in a single truth table:

P	Q	$[\neg$	$[P$	$\rightarrow$	$Q]$	$\rightarrow$	$[P$	$\wedge$	$Q]]$
T	T	F	T	T	T	**T**	T	T	T
T	F	T	T	F	F	**F**	T	F	F
F	T	F	F	T	T	**T**	F	F	T
F	F	F	F	T	F	**T**	F	F	F

Again the column that indicates the truth and falsehood of the entire formula is in boldface letters. In the following definition I call this column the main column.

Definition 2.6.

- *A formula is a tautology if and only if there are only Ts in the main column of its truth table.*
- *A formula is semantically inconsistent if and only if there are only Fs in the main column of its truth table.*
- *A formula is a propositionally contingent if and only if there are Ts and Fs in the main column of its truth table.*

Some authors call tautologies 'logically necessary' or 'valid' formulae.

Definition 2.7. *Let X be a finite set of formulae of $\mathscr{L}_1$ and ψ be a formula of $\mathscr{L}_1$.*

(i) $X \models \phi$ *iff there is no structure such that all formulae in X are true in the structure and ϕ is false in it.*

(ii) $X \models$ *iff there is no structure such that all formulae in X are true in it.*

Hodges calls expressions of the form '$X \models$' and '$X \models \phi$' 'semantic sequents'. These formal expressions are read in English in the following way:

Definition 2.8. *(i) A set X of formulae is (semantically) inconsistent iff* $X \models$. H102

(ii) An inference from the formulae in X to ϕ is valid *iff* $X \models \phi$.

(iii) X semantically entails ϕ iff $X \models \phi$.

More precisely, one should talk about inconsistency etc. *in propositional logic*. When there is a danger of confusion this specification should be added.

Lemma 2.9. $X \models \phi$ *iff* $X, \neg\phi \models$.

Here $X, \neg\phi$ is the set with ϕ and all elements of X as elements.

Proof. Assume $X \models \phi$. Then there is no structure **A** such that all formulae in X are true in the structure and ϕ is false in **A**. Therefore there is no structure such that all formulae in X are true in the structure and '$\neg\phi$' is true in **A**. Hence there is no structure that assigns T or F to all sentence letters in $X, \neg\phi$ such that all formulae in $X, \neg\phi$ are true in **A**.

Assume $X, \neg\phi \models$. Then there is no structure **A** that assigns T or F to all sentence letters in $X, \neg\phi$ such that all formulae in $X, \neg\phi$ are true in **A**. And thus there is no structure that assigns T or F to all sentence letters in $X, \neg\phi$ such that all formulae in X are true in **A** and ϕ is false in **A**; and therefore $X \models \phi$.

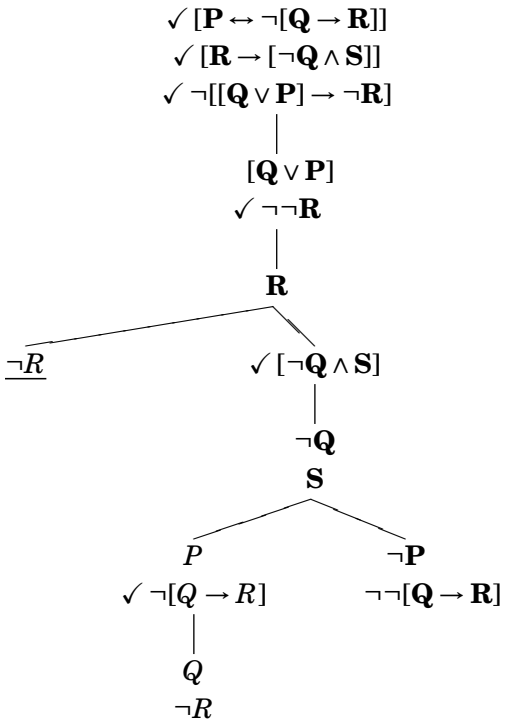
2.3 Proofs

H115ff

A tree (of formula of $\mathscr{L}_1$) has some formula at the top and it is branching to the bottom (it is not allowed that the branches of the tree merge again).

A branch in a tree is a sequence of formulae with a formula on the bottom as the last element and the topmost formula as the first element and all formulae in between (in their order) as further elements in between.

Example 2.10. In the following tree the bold-face formulae constitute a branch in the tree.



The particular shape of the formulae does not matter at the moment. You should just check that the sequence of formulae in boldface satisfies the definition of a branch: '$\neg\neg[Q \to R]$' is the end point of the branch. Taking this formula as the last formula and the topmost formula and all formulae in between yields a branch within the tree.

There are several rules that can be used to extend an existing tree; they allow one to write something at the end of one or several branches. A branch is closed if and only if the there is a bar at the bottom of the branch; otherwise the branch is unclosed (How branches can be closed will be explained below).

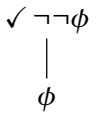
I pick one of the rules at random. Assume that a formula of the form '$\neg\neg\phi$' occurs in a tree and the formula is not yet ticked. Then there is a rule

that allows one to write the formula ϕ at the end of any unclosed branch that contains this occurrence of '$\neg\neg\phi$' and to tick the occurrence of '$\neg\neg\phi$'.

You must write ϕ at the end of *any* unclosed branch that contains the occurrence of '$\neg\neg\phi$' in question. It is also mandatory that you tick the occurrence of '$\neg\neg\phi$'.

I have been talking about occurrences of formulae. The reason is that a formula may occur several times in a tree. The rule focuses on just one occurrence of the formula. If there is a second occurrence of '$\neg\neg\phi$' on another branch you do not have to add ϕ to this branch.

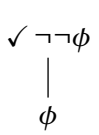
In the following I shall abbreviate this rule by writing:

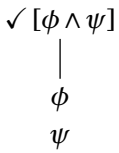


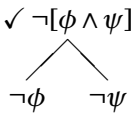
Of course, there may be formulae on the branch between '$\neg\neg\phi$' and ϕ.

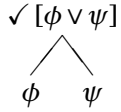
It is a common mistake to apply the rule not to the entire formula in a line, but only to parts of a formula. For instance, the formula does *not* allow you to pass from '$[P \wedge \neg\neg Q]$' to '$[P \wedge Q]$'. The rule cannot be applied to '$[P \wedge \neg\neg Q]$' at all, because this formula is of the form '$[\phi \wedge \psi]$' and not of the form '$\neg\neg\phi$'.

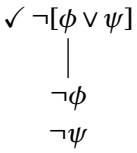
Other rules allow you to add two formulae at the end of branches. Still others allow you to add two formulae but on two different branches. Here is a list of all rules:

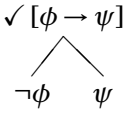


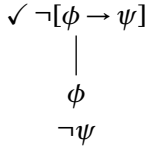


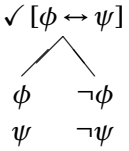


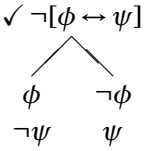












You may draw a line at the bottom of every branch on which a formula occurs together with its negation.

Definition 2.11. *A tree is a tableau if and only if all its branchings follow one of the above derivation rules.*

A branch is closed if and only if there is a line at the bottom of the branch; otherwise it is open. A tableau is closed if and only if all its branches are closed.

Definition 2.12. *Assume X is a finite set of formulae of $\mathscr{L}_1$. Then $X \vdash$ if and only if there is a closed tableau with all formulae in X at the top.*

Definition 2.13. *$X \vdash \phi$ if and only if $X, \neg\phi \vdash$.*

Thus we have $X \vdash \phi$ iff there is a closed tableau with all formulae in X and '$\neg\phi$' at the top. The elements of X are called *premisses*.

Instead of writing '$\{\psi_1, \ldots, \psi_n\} \vdash \phi$' one may also write '$\psi_1, \ldots, \psi_n \vdash \phi$'. That is, one may drop the set brackets around an enumeration of the premisses. Below '$X, \neg\phi$' stands for the set of all formulae in X plus the formula ϕ.

'$X \vdash \phi$' is read as 'X syntactically entails ϕ' or as 'ϕ is provable from the premisses in X', and '$\vdash \phi$' is read as 'ϕ is a (syntactic) theorem'.

X is allowed to be the empty set $\emptyset$ and the same conventions apply. Thus $\vdash \phi$ if and only if there is a closed tableau with '$\neg\phi$' at the top. In this case ϕ is called *provable*.

'$X \vdash \phi$' is called a 'syntactic sequent'. H116

Finally I mention a result without proof that relates $\models$ and $\vdash$: H118

Theorem 2.14 (Adequacy)**.** *For all sets X of formulae of $\mathscr{L}_1$ and formulae ϕ the following holds:*

$$X \models \phi \textit{ iff } X \vdash \phi$$

The left-to-right direction is called the Completeness Theorem (for propositional logic); the right-to-left direction is called the Soundness Theorem (for propositional logic).

3 Predicate logic

3.1 The syntax of the language of predicate logic

Hodges (2001) does not specify the syntax of predicate logic.

In general Hodges does not carefully distinguish between the formal and the natural language.

Definition 3.1 (predicate letters)**.** *All upper case Latin letters 'A^0', 'B^0', 'C^0', 'A^0', 'B^0_0', 'C^0_0',...,'A^{24}_8',... with an arbitrary natural number or no number as subscript and with some number as superscript are predicate letters.*

The value of the upper index of a predicate letter is called its arity. The predicate letter 'P^3_4', for example, has arity 3. A predicate letter of arity 3, for example, is sometimes called a 3-place predicate letter.

Definition 3.2 (variables)**.** *'x', 'y', 'z', 'x_1', 'y_1', 'z_1', 'x_2',... are variables.*

Definition 3.3 (individual constants)**.** *'a', 'b', 'c', 'd', 'a_1', 'b_1', 'c_4', 'd_1', 'a_2',... are individual constants.*

Hodges calls the individual constants 'designators' and 'proper names'.

Definition 3.4 (atomic formulae)**.** *If P is a predicate letter of arity n and $t_1,\ldots,t_n$ are variables or individual constants, then '$Pt_1\ldots t_n$' is an atomic formula. If t_1 and t_2 are variables or individual constants, then '$t_1 = t_2$' is also an atomic formula.*

'$G^3_5xd_4y$', '$x = a$' and 'G^2xx', for instance, are atomic formulae.

In an atomic formula the arity of a predicate letter is obvious: it is the numbers of variables and individual constants following the predicate letter. Therefore we adopt the following convention:

> Superscripts of predicate letters may be dropped in atomic formulae.

Definition 3.5. *A quantifier is an expression '$\forall v$' or '$\exists v$' where v is a variable.*

There are alternative symbols for '$\forall$' and '$\exists$' (which will not be used here or in examinations). '$\bigwedge v$', 'Πv' and '(v)' are sometimes used instead of '$\forall v$', and '$\bigvee v$' and 'Σv' instead of '$\exists v$'.

$\mathscr{L}_1$ was used for the language of propositional logic; the language of predicate logic is labelled $\mathscr{L}_2$.

Definition 3.6 (formula of $\mathscr{L}_2$)**.**

(i) All atomic formulae are formulae of $\mathscr{L}_2$.

(ii) If ϕ and ψ are formulae of $\mathscr{L}_2$, then '$\neg\phi$', '$[\phi \wedge \psi]$', '$[\phi \vee \psi]$', '$[\phi \to \psi]$' and '$[\phi \leftrightarrow \psi]$' are formulae of $\mathscr{L}_2$.

(iii) If v is a variable and ϕ is a formula with an occurrence of v but without an occurrence of a quantifier '$\forall v$' or '$\exists v$', then '$\forall v\phi$' and '$\exists v\phi$' are formulae of $\mathscr{L}_2$.

Examples of formulae of the language $\mathscr{L}_2$ of predicate logic are:

- $\forall x[P^2xa \to Q^1x]$
- $\neg[\forall x \forall y[P^3_2axx \wedge \exists z P^3_2zyc] \wedge P^0]$
- P^3xya
- $[[\forall z \exists y Rzy \leftrightarrow \exists z Rzz] \wedge \forall z Pz]$ This is a formula in accordance with Definition 3.6 (iii), although there are three occurrences of quantifiers involving z.

Do not try to understand these expressions, just check that these expressions are formulae of $\mathscr{L}_2$ according to the definition above.

The following two displayed formulae are *not* formulae of $\mathscr{L}_2$:

$$\forall x[Py \to Qy]$$

This is not a formula, because the variable x has no occurrence in '$[Py \to Qy]$'.

$$\exists x[Pxa \wedge \forall x[\neg Gx \vee Qx]]$$

This is not a formula, because the quantifier '$\forall x$' occurs in '$[Pxa \wedge \forall x[\neg Gx \vee Qx]]$', which is excluded in Definition 3.6 (iii).

Many other authors use a slightly different definition of a formula according to which the two expressions above would be formulae.

Again superscripts of predicate letters may be skipped. So for the first formula one would usually write '$\forall x[Pxa \to Qx]$'.

H176ff

Definition 3.7 (scope of a quantifier)**.** *The scope of a quantifier within a given formula ϕ is the smallest formula within ϕ that contains this quantifier.*[1]

[1] To be precise, I should talk about occurrences of quantifiers and formulae. In the formula '$[\exists xPx \wedge \exists xQx]$' the quantifier '$\exists x$' occurs twice. The scopes of the two occurrences are obviously different. However, I shall suppress the reference to particular occurrences if it is clear which occurrence is discussed.

Definition 3.8 (bound occurrence of a variable)**.** *An occurrence of a variable v is bound if and only if it is in the scope of a quantifier '$\forall v$' or '$\exists v$'.*

An occurrence of a variable is free if and only if it is not bound.

Definition 3.9 (closed formulae)**.** *A formula is closed if and only if all occurrences of variables are bound in the formula.*

I list some examples. The underlined occurrences of variables are free in the respective formulae.

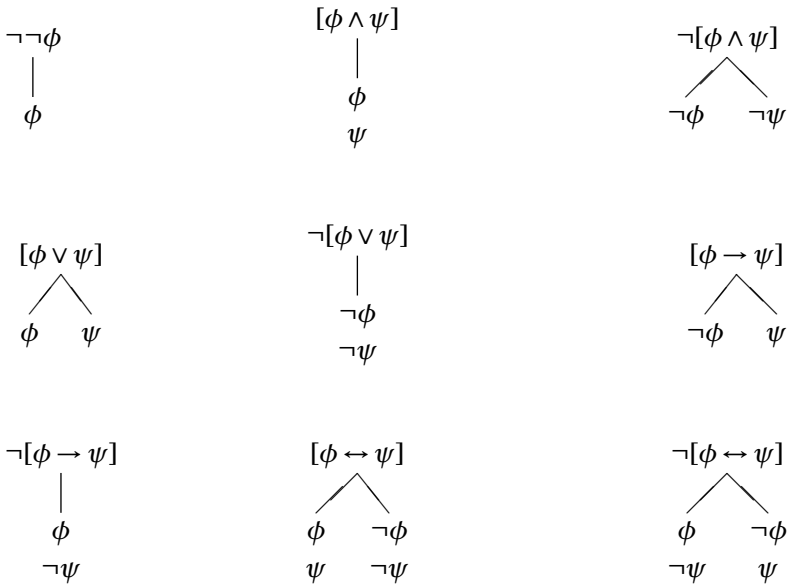
- $[\forall x[Px\underline{y} \rightarrow Qax] \leftrightarrow P\underline{x}\underline{y}]$
- $\exists z \neg [Pa\underline{x} \wedge \neg [\exists x Px \vee Qz\underline{x}]]$
- $[[P\underline{x} \wedge A\underline{x}\underline{y}] \wedge R\underline{x}\underline{y}]$

The occurrences of variables that immediately follow the quantifier symbols '$\forall$' and '$\exists$' are usually not considered. One can take all of them as bound occurrences or as neither bound nor free.

3.2 Proofs

Most definitions of section 2.3 carry over from propositional logic. Formulae of $\mathscr{L}_1$ are replaced by closed formulae of the language $\mathscr{L}_2$ of predicate logic. Moreover, the formulae are not ticked and it is not necessary to extend every branch containing the occurrence of the formula in question. H190ff

The rules for quantifiers look as follows:



There are additional rules for identity:

ϕ
$D = E$
ψ

provided that the individual constant D occurs in ϕ, and ψ is the result of replacing one or more occurrences of D in ϕ by occurrences of E.

Here it is only required that ϕ and '$D = E$' occur on the branch; it is not required that '$D = E$' has ϕ directly above it. It is also allowed that '$D = E$' occurs before ϕ on the branch.

The following rule differs from the first only in the order of the individual constants D and E.

ϕ
$E = D$
ψ

provided that the individual constant D occurs in ϕ, and ψ is the result of replacing one or more occurrences of D in ϕ by occurrences of E.

The remaining rules concern the quantifiers:

$\forall v\phi$
ψ

provided that there is an individual constant D which has already occurred in the branch above ψ, and ψ is the result of replacing every free occurrence of the variable v in ϕ by D.

$\exists v\phi$
ψ

provided that ψ is the result of replacing every free occurrence of v in ϕ by the individual constant D and D has not occurred anywhere in the branch above ψ.

$\neg\forall v\phi$
$\exists v\neg\phi$

$\neg\exists v\phi$
$\forall v\neg\phi$

Hodges (2001) mentions a rule VII, but he does not allow it to be used. This rule must *not* be used according to the examination regulations, unless you are explicitly allowed to do so.

The rule for drawing bars at the end of branches is as follows:

> You may draw a line at the bottom of every branch on which a formula occurs together with its negation or on which '$\neg D = D$' occurs for some individual constant D.

The definition of a tableau, a closed tree and so on are as in section section 2.3.

Definition 3.10. *Assume X is a finite set of formulae of $\mathscr{L}_2$ and ϕ is a formula of $\mathscr{L}_2$.*

(i) $X \vdash$ *if and only if there is a closed tableau with all formulae in X at the top.*

(ii) $X \vdash \phi$ *if and only if* $X, \neg\phi \vdash$.

Thus $X \vdash \phi$ if and only if there is a closed branch with all formulae in X and '$\neg\phi$' at the top. The elements of X are called *premisses*. '$X \vdash \phi$' is read as 'ϕ is provable from the premisses in X'.

3.3 Interpretations and counterexamples

In propositional logic the tableau method yields a decision procedure for deciding whether a syntactic sequent is correct or not, that is, for deciding whether a formula is provable from a finite set of formulae.[2] If all complex formulae are ticked and the tableau is not yet closed, we have done everything we can do. Therefore the sequent cannot shown to be correct in this case. And from Theorem 2.14 we know that the corresponding semantic sequent is also not correct.

If a tree does not close in predicate logic, that doesn't show anything (apart from some special cases). It could still close within the next 100 steps. The reason is that we could still try another individual constant.

For predicate logic Hodges does not define correctness for semantic sequents, that is, '$X \models \phi$' is not defined where ϕ and all elements of X are sentences of $\mathscr{L}_2$. Counterexamples appeal to validity of arguments in English. The idea is that a syntactic sequent '$X \vdash \phi$' (X a set of formulae of $\mathscr{L}_2$, ϕ a formula of $\mathscr{L}_2$) is not correct if there is a translation of all the formulae into English such that the resulting argument in English is not valid.

In Hodges' terminology a (predicate) interpretation is a translation of some predicate letters and individual constants into corresponding expressions in English. Thus, e.g., a binary predicate letter will be translated as binary predicate expressions of English (and similarly for predicate letters of arbitrary arity), and individual constants will be translated as designators of English. An interpretation is an interpretation for a particular formula if and only if the interpretation comprises translations for all predicate letters and individual constants occurring in the formula. H187

An interpretation specifies translations for the predicate letters and individual constants, but not for the identity symbol because the identity symbol is always translated in the usual way:

$$x = y \quad : \quad x \text{ is (identical with) } y$$

[2] Here it is important that Hodges allows only finitely many premisses in a sequent.

I give an example of an interpretation.

Example 3.11. The following is an interpretation for the formula $\forall x[Px \rightarrow Rxa]$:

Px	:	x is a tree on Christ Church meadow
Rxy	:	x is in y
a	:	Oxford

This example shows that a predicate letter might receive a fairly complex predicate expression of English as translation.

Given the above interpretation one can translate the entire formula '$\forall x[Px \rightarrow Rxa]$' into English as

> All trees on Christ Church meadow are in Oxford.

Thus under the interpretation of Example 3.11 the formula '$\forall x[Px \rightarrow Rxa]$' becomes a true English sentence. It is not hard to think of an interpretation that renders '$\forall x[Px \rightarrow Rxa]$' a false sentence.

A domain is some arbitrary set. An English sentence is true in a domain, if and only if the sentence is true if only elements in the domain are considered. For instance, in the domain $\{2,4\}$ the sentence 'All numbers are even' is true. If the domain is the set of all integers the sentence is false. However, we must presuppose that the English sentence contains only designators designating objects in the domain. For instance, you may not specify 'Brazil' as the translation of the individual constant c and then take the set of all European countries as the domain of quantification. Thus the objects denoted by the translations of the individual constants must be in the domain of quantification. In the following it is always assumed that the interpretations respect this restriction.

In order to refute $X \vdash \phi$, one can specify a domain $\mathscr{D}$ plus an interpretation for ϕ and all formulae in X, such that all formulae in X are true under this interpretation in $\mathscr{D}$ and ϕ is false under this interpretation in $\mathscr{D}$.

> A counterexample to the claim that $X \vdash \phi$ (or, for short, a counterexample to '$X \vdash \phi$') consists in a specification of the following:
>
> - a domain of quantification (any set is allowed as domain of quantification; in particular the domain can be the empty set)
> - translations of all predicate letters involved in the argument to English predicates of the same arity.
> - translations of all individual constants involved in the argument (the translations must designate only objects in $\mathscr{D}$)

where all formulae in X are true under this interpretation in the domain and ϕ is false under this interpretation in the domain.

Of course some sort of justification is required for the claim that a counterexample can be used to refute a claim like $X \vdash \phi$. The claim cannot be proved formally because it involves informal notions like the truth of English sentences in a given domain. But an inspection of the tableau system should show that the if all the translations of elements of X are true, then ϕ cannot be false, if $X \vdash \phi$.

Example 3.12. The following is a counterexample to '$\forall x \exists y Gxy \vdash \exists y \forall x Gxy$':

Gxy	:	x is separated by the sea from y
domain:	:	{Canada, U.S.A., Germany, France}

The translation of the premiss '$\forall x \exists y Gxy$' is then 'Everything is separated by the sea from something', which is true if we focus on the countries in the domain: Canada is separated by the sea, e.g., from France. The translation of the conclusion '$\exists y \forall x Gxy$' is 'Something is separated by the sea from everything.' is clearly false, because, e.g., France is not separated from Germany by the sea.

There are also simpler counterexamples like the following:

Gxy	:	x is identical to y
domain:	:	$\{1,2\}$

This is a counterexample because 1 and 2 are both identical to something in the domain, respectively (namely to themselves); but neither 1 nor 2 are identical to everything in the domain.

This means in practice that if one wonders whether an argument is valid in predicate logic, one can try to prove $X \vdash \phi$ by the tableau method in order to prove that the argument is valid or one can try to find a counterexample in order to show that the argument is not valid.

Example 3.13. There is a counterexample to

$$\forall x[Px \to \exists y Rxy], \forall x[Rxa \to Qx] \vdash \forall x[Px \to Qx].$$

Px	:	is a European capital city
Rxy	:	x is the capital of y
a	:	Italy
Qx	:	is in Italy
domain:	:	the set of all capital cities and all European countries

The translations of the two premisses and the conclusion are then:

1. Every European capital is the capital of something.

2. Every capital of Italy is in Italy.

3. Every European capital is in Italy.

Sometimes it may be easier to specify a structure rather than a translation in order to refute the validity of an argument.

Basically the idea is this. Instead of providing a translation for each predicate letter, one assigns a truth value to each 0-place predicate letter, a set of objects to each 1-place predicate letter, and a set of ordered n-tuples to each n-place predicate letter, where $n > 1$.

In the case of Example 3.13 one could write:

P	:	{Rome, Berlin}
R	:	{⟨Rome, Italy⟩}, {⟨Berlin, Germany⟩}
a	:	Italy
Q	:	{Rome}
domain:	:	{Rome, Berlin, Italy, Germany}

All the objects (Rome, Berlin, Italy, Germany) in the relations must be in the domain. The structure follows the idea of the above translation with the restriction to two capitals and two countries.

This approach is allowed and sometimes sensible.

It is possible to assign a predicate letter the empty relation (i.e., the empty set). It is also possible to assign the same relation to two different predicate letters of the same arity.

Note. In counterexamples you should not presuppose any particular knowledge only you have. For instance, taking the set of the objects on your desk as domain is not a good idea. It is unlikely that the examiner knows the objects on your desk. Usually it is also sensible to choose a simple counterexample if possible.

Appendix A: Rules for dropping brackets

Most logicians employ certain rules for dropping brackets. For instance, they do not write '$[[P \wedge Q] \wedge R]$', but rather simply '$P \wedge Q \wedge R$'. The conventions explained here do not form part of the syllabus, but they should be useful when reading texts not following Hodges' somewhat idiosyncratic notation.

It is not recommended that you use these conventions. The bracketing conventions introduce more possibilities for mistakes. If brackets are missing from your formulae, the conventions below will be applied.

The conventions below concern the language $\mathscr{L}_1$ of propositional logic and the language $\mathscr{L}_2$ of predicate logic.

The expression that is obtained from dropping brackets is not itself a formula but rather a mere abbreviation of the original formula.

Bracketing Convention 1. *Brackets surrounding the whole formula may be dropped.*

For instance, one may write '$P \rightarrow [Q \vee P]$' instead of '$[P \rightarrow [Q \vee P]]$'. However, this convention does not allow to drop any brackets from '$\neg[P \rightarrow [Q \vee P]]$', because only a part and not the whole formula is surrounded by the brackets.

This convention is a possible source for errors. Assume the formula '$P \wedge Q$' is to be negated. Then it seems natural to write '$\neg P \wedge Q$'. But the latter formula is *not* the negation of the first. By Convention 1, '$P \wedge Q$' is short for '$[P \wedge Q]$' and thus its negation is '$\neg[P \wedge Q]$'. From the last formula no brackets can be dropped.

Bracketing Convention 2. *If '$[\phi \wedge \psi] \wedge \chi$' is a part of a formula, the occurrences of the brackets may be dropped and '$\phi \wedge \psi \wedge \chi$' may be written. An analogous convention applies to '$\vee$'.*

According to this convention one may write '$[Ga \wedge Gb \wedge Gc]$' instead of '$[[Ga \wedge Gb] \wedge Gc]$', for instance. Using the Convention 1 this may even be shortened to '$Ga \wedge Gb \wedge Gc$'. Convention 2 also allows to write '$\forall x[Px \rightarrow [Qx \vee Rx \vee Sxa]]$' instead of '$\forall x[Px \rightarrow [[Qx \vee Rx] \vee Sxa]]$'.

In mathematics one may write '$3 \cdot 5 + 4$' instead of '$(3 \cdot 5) + 4$', because '$\cdot$' binds more strongly that '$+$'. Similar conventions are adopted in logic.

We first fix which truth-functor symbol (connective) binds more strongly. A connective $\circ$ binds more strongly than another connective $\bullet$ if and only if $\circ$

stands further to the left than • in the following line:

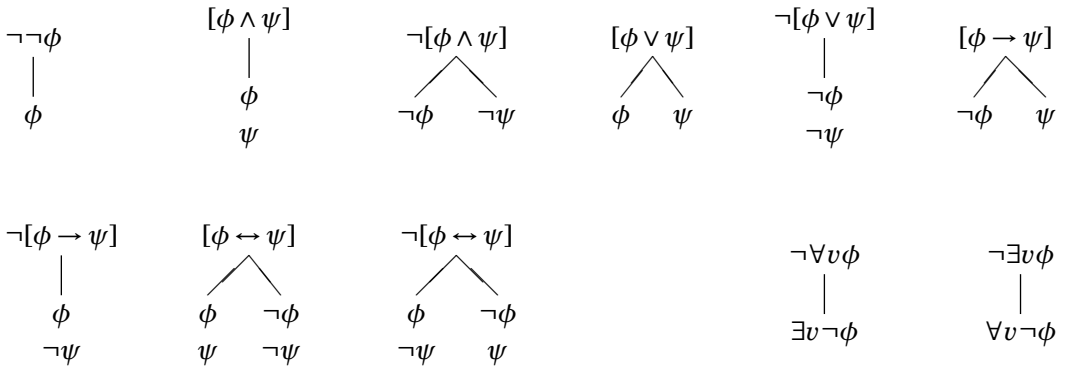
$$\wedge \quad \vee \quad \rightarrow \quad \leftrightarrow$$

The last bracketing convention allows one to drop brackets where the grouping of the symbols is already clear from the above list.

Bracketing Convention 3. *Assume* $\circ$ *and* $\bullet$ *are truth-functor symbols (connectives). If a formula contains an expression of the form* '$[\phi \circ \psi] \bullet \chi$' *(or* '$\phi \bullet [\psi \circ \chi]$'*) and* $\circ$ *binds more strongly than* $\bullet$*, one may write* '$\phi \circ \psi \bullet \chi$' *(or* '$\phi \bullet \psi \circ \chi$' *in the latter case).*

'$[P \wedge Q] \rightarrow R$' (Convention 1 has already been used) may be shortened to '$P \wedge Q \rightarrow R$' because '$\wedge$' binds more strongly than '$\rightarrow$'.

Appendix B: tableaux rules



$$\begin{array}{c}\phi\\ D = E\\ \hline \psi\end{array}$$

provided that the individual constant D occurs in ϕ, and ψ is the result of replacing one or more occurrences of D in ϕ by occurrences of E.

$$\begin{array}{c}\phi\\ E = D\\ \hline \psi\end{array}$$

provided that the individual constant D occurs in ϕ, and ψ is the result of replacing one or more occurrences of D in ϕ by occurrences of E.

$$\begin{array}{c}\forall v\phi\\ \hline \psi\end{array}$$

provided that there is an individual constant D which has already occurred in the branch above ψ, and ψ is the result of replacing every free occurrence of the variable v in ϕ by D.

$$\begin{array}{c}\exists v\phi\\ \hline \psi\end{array}$$

provided that ψ is the result of replacing every free occurrence of v in ϕ by the individual constant D and D has not occurred anywhere in the branch above ψ.

You may draw a line at the bottom of every branch on which a formula occurs together with its negation or on which '$\neg D = D$' occurs for some individual constant D.

Appendix C: Quotation

In logic one talks about expressions in natural and formal languages. By enclosing an expression in quotation marks one obtains a term designating that expression. For instance, the expression

'Italy'

refers to the word that begins with an 'I' followed by 't', 'a', 'l' and 'y'.

When expressions are displayed, quotation marks are usually skipped.

Often in logic one does not only intend to talk about a single expression, but about many expressions of a certain form simultaneously. For instance, one tries to affirm the followiong claim:

> A conjunction of two sentences is true if and only if both sentences are true.

This can be more formally expressed in the following way:

(K1) If ϕ and ψ are sentences, then the expression beginning with ϕ followed by '$\wedge$' and ψ is true if and only if ϕ and ψ are true .

Here we do not talk about 'ϕ' and 'ψ', rather 'ϕ' and 'ψ' are used as variables ranging over sentences. These variables belong to the language we are actually using, that is, in English enriched by some symbols. Therefore there is no need for quotation marks enclosing 'ϕ' and 'ψ', respectively. But we talk about the conjunction symbol '$\wedge$', so it has to be enclosed in quotation marks.

We adopt a convention for abbreviating claims like (K1). According to this convention (K1) can be rephrased in the following way:

(K2) If ϕ and ψ are sentences, then '$\phi \wedge \psi$' is true if and only if ϕ and ψ are true .

We still do not intend to talk about the Greek letters 'ϕ' and 'ψ', but rather about sentences of a language that obtained by replacing sentences for the Greek letters 'ϕ' and 'ψ' in '$\phi \wedge \psi$'.

The convention has been introduced here only by way of example, and this should be sufficient for the understanding of the text here and Hodges' book. The details of this convention are tricky and quotation has puzzled logicians and philosophers. There are ways to avoid quotation marks altogether, but this is usually on he cost of readability.

BIBLIOGRAPHY

Hodges, Wilfrid (2001), *Logic: An Introduction to Elementary Logic*, second edn, Penguin Books, London.