1 Classical Logic and the Material Conditional

1.1 Introduction

1.1.1 The first purpose of this chapter is to review classical propositional logic, including semantic tableaux. The chapter also sets out some basic terminology and notational conventions for the rest of the book.

1.1.2 In the second half of the chapter we also look at the notion of the conditional that classical propositional logic gives, and, specifically, at some of its shortcomings.

1.1.3 The point of logic is to give an account of the notion of validity: what follows from what. Standardly, validity is defined for inferences couched in a formal language, a language with a well-defined vocabulary and grammar, the *object language*. The relationship of the symbols of the formal language to the words of the vernacular, English in this case, is always an important issue.

1.1.4 Accounts of validity themselves are in a language that is normally distinct from the object language. This is called the *metalanguage*. In our case, this is simply mathematical English. Note that 'iff' means 'if and only if'.

1.1.5 It is also standard to define two notions of validity. The first is *semantic*. A valid inference is one that *preserves truth*, in a certain sense. Specifically, every interpretation (that is, crudely, a way of assigning truth values) that makes all the premises true makes the conclusion true. We use the metalinguistic symbol '$\models$' for this. What distinguishes different logics is the different notions of interpretation they employ.

1.1.6 The second notion of validity is *proof-theoretic*. Validity is defined in terms of some purely formal procedure (that is, one that makes reference only to the symbols of the inference). We use the metalinguistic symbol '$\vdash$' for this notion of validity. In our case, this procedure will (mainly) be one employing tableaux. What distinguish different logics here are the different tableau procedures employed.

1.1.7 Most contemporary logicians would take the semantic notion of validity to be more fundamental than the proof-theoretic one, though the matter is certainly debatable. However, given a semantic notion of validity, it is always useful to have a proof-theoretic notion that corresponds to it, in the sense that the two definitions always give the same answers. If every proof-theoretically valid inference is semantically valid (so that $\vdash$ entails $\models$) the proof-theory is said to be *sound*. If every semantically valid inference is proof-theoretically valid (so that $\models$ entails $\vdash$) the proof-theory is said to be *complete*.

1.2 The Syntax of the Object Language

1.2.1 The symbols of the object language of the propositional calculus are an infinite number of propositional parameters:[1] $p_0, p_1, p_2, \ldots$; the connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\supset$ (material conditional), $\equiv$ (material equivalence); and the punctuation marks: (,).

1.2.2 The (well-formed) formulas of the language comprise all, and only, the strings of symbols that can be generated recursively from the propositional parameters by the following rule:

If A and B are formulas, so are $\neg A$, $(A \vee B)$, $(A \wedge B)$, $(A \supset B)$, $(A \equiv B)$.

1.2.3 I will explain a number of important notational conventions here. I use capital Roman letters, $A, B, C, \ldots$, to represent arbitrary formulas of the object language. Lower-case Roman letters, p, q, r, ..., represent arbitrary,

[1] These are often called 'propositional variables'.

but distinct, propositional parameters. I will always omit outermost parentheses of formulas if there are any. So, for example, I write $(A \supset (B \vee \neg C))$ simply as $A \supset (B \vee \neg C)$. Upper-case Greek letters, $\Sigma, \Pi, \ldots$, represent arbitrary sets of formulas; the empty set, however, is denoted by the (lower case) ϕ, in the standard way. I often write a finite set, $\{A_1, A_2, \ldots, A_n\}$, simply as $A_1, A_2, \ldots, A_n$.

1.3 Semantic Validity

1.3.1 An *interpretation* of the language is a function, ν, which assigns to each propositional parameter either 1 (true), or 0 (false). Thus, we write things such as $\nu(p) = 1$ and $\nu(q) = 0$.

1.3.2 Given an interpretation of the language, ν, this is extended to a function that assigns every formula a truth value, by the following recursive clauses, which mirror the syntactic recursive clauses:[2]

$\nu(\neg A) = 1$ if $\nu(A) = 0$, and 0 otherwise.
$\nu(A \wedge B) = 1$ if $\nu(A) = \nu(B) = 1$, and 0 otherwise.
$\nu(A \vee B) = 1$ if $\nu(A) = 1$ or $\nu(B) = 1$, and 0 otherwise.
$\nu(A \supset B) = 1$ if $\nu(A) = 0$ or $\nu(B) = 1$, and 0 otherwise.
$\nu(A \equiv B) = 1$ if $\nu(A) = \nu(B)$, and 0 otherwise.

1.3.3 Let Σ be any set of formulas (the premises); then A (the conclusion) is a *semantic consequence* of Σ ($\Sigma \models A$) iff there is no interpretation that makes all the members of Σ true and A false, that is, every interpretation that makes all the members of Σ true makes A true. '$\Sigma \not\models A$' means that it is not the case that $\Sigma \models A$.

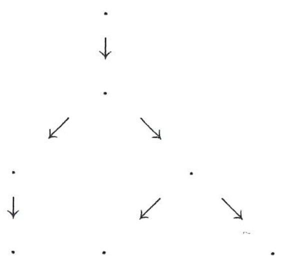
1.3.4 A is a *logical truth* (*tautology*) ($\models A$) iff it is a semantic consequence of the empty set of premises ($\phi \models A$), that is, every interpretation makes A true.

[2] The reader might be more familiar with the information contained in these clauses when it is depicted in the form of a table, usually called a *truth table*, such as the one for *conjunction* displayed:

$\wedge$	**1**	**0**
1	1	0
0	0	0

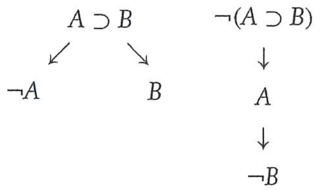
1.4 Tableaux

1.4.1 A *tree* is a structure that looks, generally, like this:[3]



The dots are called *nodes*. The node at the top is called the *root*. The nodes at the bottom are called *tips*. Any path from the root down a series of arrows as far as you can go is called a *branch*. (Later on we will have trees with infinite branches, but not yet.)

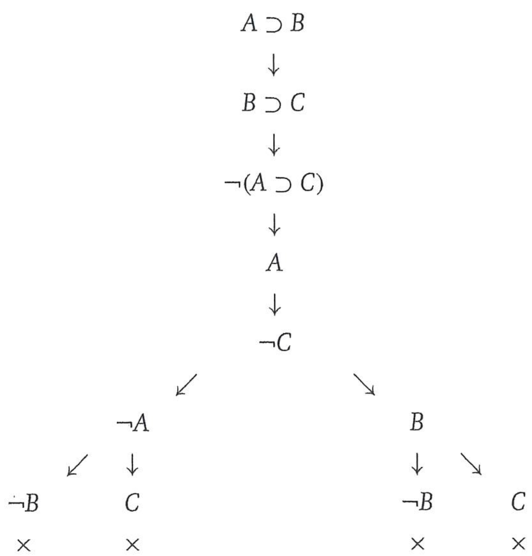
1.4.2 To test an inference for validity, we construct a tableau which begins with a single branch at whose nodes occur the premises (if there are any) and the negation of the conclusion. We will call this the *initial list*. We then apply rules which allow us to extend this branch. The rules for the conditional are as follows:



The rule on the right is to be interpreted as follows. If we have a formula $\neg(A \supset B)$ at a node, then every branch that goes through that node is extended with two further nodes, one for A and one for $\neg B$. The rule on the left is interpreted similarly: if we have a formula $A \supset B$ at a node, then every branch that goes through that node is split at its tip into two branches; one contains a node for $\neg A$; the other contains a node for B.

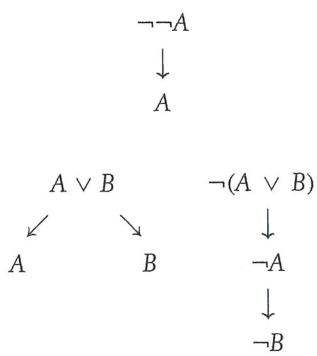
[3] Strictly speaking, for those who want the precise mathematical definition, it is a partial order with a unique maximum element, x_0, such that for any element, x_n, there is a unique finite chain of elements $x_n \leq x_{n-1} \leq \cdots \leq x_1 \leq x_0$.

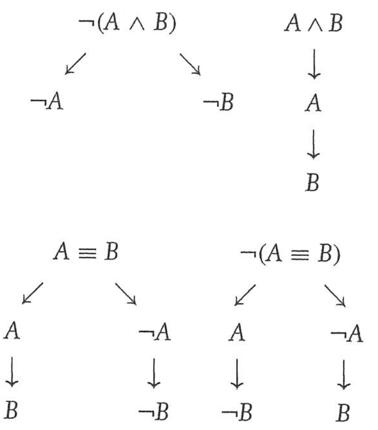
1.4.3 For example, to test the inference whose premises are $A \supset B$, $B \supset C$, and whose conclusion is $A \supset C$, we construct the following tree:



The first three formulas are the premises and negated conclusion. The next two formulas are produced by the rule for the negated conditional applied to the negated conclusion; the first split on the branch is produced by applying the rule for the conditional to the first premise; the next splits are produced by applying the same rule to the second premise. (Ignore the '×'s: we will come back to those in a moment.)

1.4.4 The other connectives also have rules, which are as follows.





Intuitively, what a tableau means is the following. If we apply a rule to a formula, then if that formula is true in an interpretation, so are the formulas below on at least one of the branches that the rule generates. (Of course, there may be only one such branch.) This is a useful mnemonic for remembering the rules. It must be stressed, though, that officially the rules are purely formal.

1.4.5 A tableau is *complete* iff every rule that can be applied has been applied. By applying the rules over and over, we may always construct a complete tableau. In the present case, the branches of a completed tableau are always finite,[4] but in the tableaux of some subsequent chapters they may be infinite.

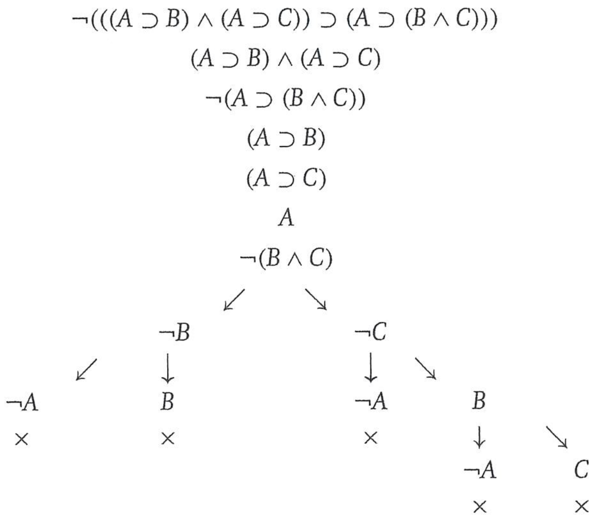
1.4.6 A branch is *closed* iff there are formulas of the form A and $\neg A$ on two of its nodes; otherwise it is *open*. A closed branch is indicated by writing an $\times$ at the bottom. A *tableau* itself is closed iff every branch is closed; otherwise it is open. Thus the tableau of 1.4.3 is closed: the leftmost branch contains A and $\neg A$; the next contains A and $\neg A$ (and C and $\neg C$); the next contains B and $\neg B$; the rightmost contains C and $\neg C$.

1.4.7 A is a proof-theoretic consequence of the set of formulas $\Sigma (\Sigma \vdash A)$ iff there is a complete tree whose initial list comprises the members of Σ and the negation of A, and which is closed. We write $\vdash A$ to mean that $\phi \vdash A$,

[4] This is not entirely obvious, though it is not difficult to prove.

that is, where the initial list of the tableau comprises just $\neg A$. '$\Sigma \nvdash A$' means that it is not the case that $\Sigma \vdash A$.[5]

1.4.8 Thus, the tree of 1.4.3 shows that $A \supset B, B \supset C \vdash A \supset C$. Here is another, to show that $\vdash ((A \supset B) \wedge (A \supset C)) \supset (A \supset (B \wedge C))$. To save space, we omit arrows where a branch does not divide.



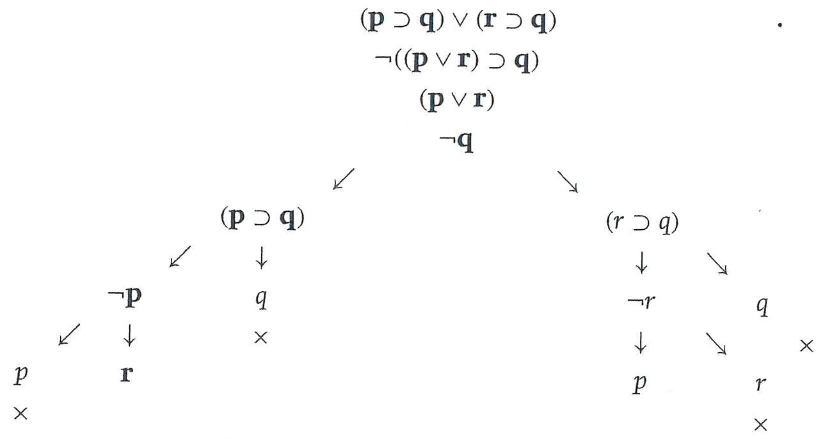
Note that when we find a contradiction on a branch, there is no point in continuing it further. We know that the branch is going to close, whatever else is added to it. Hence, we need not bother to extend a branch as soon as it is found to close. Notice also that, wherever possible, we apply rules that do not split branches before rules that split branches. Though this is not essential, it keeps the tableau simpler, and is therefore useful practically.

1.4.9 In practice, it is also a useful idea to put a tick at the side of a formula once one has applied a rule to it. Then one knows that one can forget about it.

[5] There may, in fact, be several completed trees for an inference, depending upon the order of the premises in the initial list and the order in which rules are applied. Fortunately, they all give the same result, though this is not entirely obvious. See 1.14, problem 5.

1.5 Counter-models

1.5.1 Here is another example, to show that $(p \supset q) \vee (r \supset q) \nvdash (p \vee r) \supset q$.



The tableau has two open branches. The leftmost one is emphasised in bold for future reference.

1.5.2 The tableau procedure is, in effect, a systematic search for an interpretation that makes all the formulas on the initial list true. Given an open branch of a tableau, such an interpretation can, in fact, be read off from the branch.[6]

1.5.3 The recipe is simple. If the propositional parameter, p, occurs at a node on the branch, assign it 1; if $\neg p$ occurs at a node on the branch, assign it 0. (If neither p nor $\neg p$ occurs in this way, it may be assigned anything one likes.)

1.5.4 For example, consider the tableau of 1.5.1 and its (bolded) leftmost open branch. Applying the recipe gives the interpretation, ν, such that $\nu(r) = 1$, and $\nu(p) = \nu(q) = 0$. It is simple to check directly that $\nu((p \supset q) \vee (r \supset q)) = 1$ and $\nu((p \vee r) \supset q) = 0$. Since p is false, $p \supset q$ is true, as is $(p \supset q) \vee (r \supset q)$. Since r is true, $p \vee r$ is true; but q is false; hence, $(p \vee r) \supset q$ is false.

[6] If one thinks of constructing a tableau as a search procedure for a counter-model, then the soundness and completeness theorems constitute, in effect, a proof that the procedure always gives the right result, that is, which *verifies* the algorithm in question.

1.5.4a Note that the tableau of 1.4.8 shows that *any* inference of the form in question is valid. That is, A, B and C can be *any* formulas. To show that an inference is invalid, we have to construct a counter-model, and this means assigning truth values to *particular* formulas. This is why the example just given uses 'p', 'q' and 'r', not 'A', 'B' and 'C'. One may say that an inference expressed using schematic letters ('A's and 'B's) is invalid, but this must mean that there are some formulas that can be substituted for these letters to make it so. Thus, we may write $A \nvDash B$, since $p \nvDash q$. But note that this does not rule out the possibility that some inferences of that form are valid, e.g., $p \vDash q \vee \neg q$.

1.5.5 As one would hope, the tableau procedure we have been looking at is sound and complete with respect to the semantic notion of consequence, i.e., if Σ is a finite set of sentences, $\Sigma \vdash A$ iff $\Sigma \models A$. That is, the search procedure really works. If there is an interpretation that makes all the formulas on the initial list true, the tableau will have an open branch which, in effect, specifies one. And if there is no such interpretation, every branch will close. These facts are not obvious. The proof is in 1.11.[7]

1.6 Conditionals

1.6.1 In the remainder of this chapter, we look at the notion of conditionality that the above, classical, semantics give us, and at its inadequacy. But first, what is a conditional?

1.6.2 Conditionals relate some proposition (the *consequent*) to some other proposition (the *antecedent*) on which, in some sense, it depends. They are expressed in English by 'if' or cognate constructions:

> If the bough breaks (then) the cradle will fall.
> The cradle will fall if the bough breaks.
> The bough breaks only if the cradle falls.

[7] The restriction to finite Σ is due to the fact that tableaux have been defined only for finite sets of premises. It is possible to define tableaux for infinite sets of premises as well (not putting all the premises at the start, but introducing them, one by one, at regular intervals down the branches). If one does this, the soundness and completeness results generalise to arbitrary sets of premises. We will take up this matter again in Chapter 12 (Part II), where the matter assumes more significance.

If the bough were to break the cradle would fall.
Were the bough to break the cradle would fall.

1.6.3 Note that the grammar of conditionals imposes certain requirements on the tense (past, present, future) and mood (indicative, subjunctive) of the sentences expressing the antecedent and consequent within it. These may be different when the antecedent and consequent stand alone. To see this, just consider the following applications of *modus ponens* (if A then B; A; hence B):

If he *takes* a plane he will get there quicker.
He *will take* a plane.
Hence, he will get there quicker.

If he *had come* in the window there *would have been* foot-marks.
He *did come* in the window.
So, there *are* foot-marks.

1.6.4 Note, also, that not all sentences using 'if' are conditionals; consider, for example, 'If I may say so, you have a nice ear-ring', '(Even) if he was plump, he could still run fast', or 'If you want a banana, there is one in the kitchen.' A rough and ready test for 'if A, B' to be a conditional is that it can be rewritten equivalently as 'that A implies that B'.

1.7 The Material Conditional

1.7.1 The connective $\supset$ is usually called the *material conditional* (or *material implication*). As its truth conditions show, $A \supset B$ is logically equivalent to $\neg A \vee B$. It is true iff A is false or B is true. Thus, we have:

$$B \models A \supset B$$

$$\neg A \models A \supset B$$

These are sometimes called the 'paradoxes of material implication'.

1.7.2 People taking a first course in logic are often told that English conditionals may be represented as $\supset$. There is an obvious objection to this claim, though. If it were correct, then the truth conditions of $\supset$ would ensure the truth of the following, which appear to be false:

If New York is in New Zealand then $2 + 2 = 4$.
If New York is in the United States then World War II ended in 1945.
If World War II ended in 1941 then gold is an acid.

1.7.3 It is possible to reply to this objection as follows. These examples are, indeed, true. They strike us as counterintuitive, though, for the following reason. Communication between people is governed by many pragmatic rules of conversation, for example 'be relevant', 'assert the strongest claim you are in a position to make'. We often use the fact that these rules are in place to draw conclusions. Consider, for example, what you would infer from the following questions and replies: 'How do you use this drill?', 'There's a book over there.' (It is a drill manual. *Relevance*.) 'Who won the 3.30 at Ascot?', 'It was a horse named either Blue Grass or Red Grass.' (The speaker does not know which. *Assert the strongest information*.) These inferences are inferences, not from the *content* of what has been said, but from the fact that *it has been said*. The process is often dubbed 'conversational implicature'. Now, the claim goes, the examples of 1.7.2 strike us as odd since anyone who asserted them would be violating the rule *assert the strongest*, since, in each case, we are in a position to assert either the consequent or the negation of the antecedent (or both).

1.8 Subjunctive and Counterfactual Conditionals

1.8.1 A harder objection to the correctness of the material conditional is to the effect that there are pairs of conditionals which appear to have the same antecedent and consequent, but which clearly have different truth values. They cannot both, therefore, be material conditionals. Consider the examples:

(1) If Oswald didn't shoot Kennedy someone else did. (True)
(2) If Oswald hadn't shot Kennedy someone else would have. (False)

1.8.2 In response to this kind of example, it is not uncommon for philosophers to distinguish between two sorts of conditionals: conditionals in which the consequent is expressed using the word 'would' (called 'subjunctive' or 'counterfactual'), and others (called 'indicative'). Subjunctive conditionals, like (2), cannot be material: after all, (2) is false, though its antecedent is false (assuming the results of the Warren Commission!). But indicative conditionals may still be material.

1.8.3 The claim that the English conditional is ambiguous between subjunctive and indicative is somewhat dubious, though. There appears to be no grammatical justification for it, for a start. In (1) and (2) the 'if's are grammatically identical; it is the tenses and/or moods of the verbs involved which make the difference.

1.8.4 What these differences seem to do is to get us to evaluate the truth values of conditionals from different points in time. Thus, we evaluate (1) as true from the present, where Kennedy has, in fact, been shot. The difference of tense and mood of (2) asks us to evaluate the conditional 'If Oswald doesn't shoot Kennedy, someone else will' from the perspective of a time just before Kennedy was shot. It is, in a certain sense, the past tense of that conditional. Notice that no difference of the kind between (1) and (2) arises in the case of present-tense conditionals. There is no major difference between 'If I shoot you, you will die' and 'If I were to shoot you, you would die.'

1.9 More Counter-examples

1.9.1 There are more fundamental objections against the claim that the indicative English conditional (even if it is distinct from the subjunctive) is material. It is easy to check that the following inferences are valid.

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

If the English indicative conditional were material, the following inferences would, respectively, be instances of the above, and therefore valid, which they are clearly not.

(1) If you close switch x and switch y the light will go on. Hence, it is the case either that if you close switch x the light will go on, or that if you close switch y the light will go on. (Imagine an electrical circuit where switches x and y are in series, so that both are required for the light to go on, and both switches are open.)

(2) If John is in Paris he is in France, and if John is in London he is in England. Hence, it is the case either that if John is in Paris he is in England, or that if he is in London he is in France.

(3) It is not the case that if there is a good god the prayers of evil people will be answered. Hence, there is a god.

1.9.2 Notice that all these conditionals are indicative. Note, also, that appealing to conversational rules cannot explain why the conclusions appear odd, as in 1.7.3. For example, in the first, it is not the case that we already know which disjunct of the conclusion is true: *both* appear to be false.

1.9.3 It might be pointed out that the above arguments are valid if 'if' is understood as $\supset$. However, this just concedes the point: 'if' in English is not understood as $\supset$.

1.10 Arguments for $\supset$

1.10.1 The claim that the English conditional (or even the indicative conditional) is material is therefore hard to sustain. In the light of this it is worth asking why anyone ever thought this. At least in the modern period, a large part of the answer is that, until the 1960s, standard truth-table semantics were the only ones that there were, and $\supset$ is the only truth function that looks an even remotely plausible candidate for 'if'.

1.10.2 Some arguments have been offered, however. Here is one, to the effect that 'If A then B' is true iff '$A \supset B$' is true.

1.10.3 First, suppose that 'If A then B' is true. Either $\neg A$ is true or A is. In this first case, $\neg A \vee B$ is true. In the second case, B is true by *modus ponens*. Hence, again, $\neg A \vee B$ is true. Thus, in either case, $\neg A \vee B$ is true.

1.10.4 The converse argument appeals to the following plausible claim:

(*) 'If A then B' is true if there is some true statement, C, such that from C and A together we can deduce B.

Thus, we agree that the conditional 'If Oswald didn't kill Kennedy, someone else did' is true because we can deduce that someone other than Oswald killed Kennedy from the fact that Kennedy was murdered and Oswald did not do it.

1.10.5 Now, suppose that $\neg A \vee B$ is true. Then from this and A we can deduce B, by the *disjunctive syllogism*: $A, \neg A \vee B \vdash B$. Hence, by (*), 'If A then B' is true.

1.10.6 We will come back to this argument in a later chapter. For now, just note the fact that it uses the disjunctive syllogism.

1.11 *Proofs of Theorems

1.11.1 Definition: Let ν be any propositional interpretation. Let b be any branch of a tableau. Say that ν is *faithful* to b iff for every formula, A, on the branch, $\nu(A) = 1$.

1.11.2 Soundness Lemma: If ν is faithful to a branch of a tableau, b, and a tableau rule is applied to b, then ν is faithful to at least one of the branches generated.

Proof:
The proof is by a case-by-case examination of the tableau rules. Here are the cases for the rules for $\supset$. The other cases are left as exercises. Suppose that ν is faithful to b, that $\neg(A \supset B)$ occurs on b, and that we apply a rule to it. Then only one branch eventuates, that obtained by adding A and $\neg B$ to b. Since ν is faithful to b, it makes every formula on b true. In particular, $\nu(\neg(A \supset B)) = 1$. Hence, $\nu(A \supset B) = 0$, $\nu(A) = 1$, $\nu(B) = 0$, and so $\nu(\neg B) = 1$. Hence, ν makes every formula on b true. Next, suppose that ν is faithful to b, that $A \supset B$ occurs on b, and that we apply a rule to it. Then two branches eventuate, one extending b with $\neg A$ (the left branch); the other extending b with B (the right branch). Since ν is faithful to b, it makes every formula on b true. In particular, $\nu(A \supset B) = 1$. Hence, $\nu(A) = 0$, and so $\nu(\neg A) = 1$, or $\nu(B) = 1$. In the first case, ν is faithful to the left branch; in the second, it is faithful to the right. ■

1.11.3 Soundness Theorem: For finite Σ, if $\Sigma \vdash A$ then $\Sigma \models A$.

Proof:
We prove the contrapositive. Suppose that $\Sigma \not\models A$. Then there is an interpretation, ν, which makes every member of Σ true, and A false – and hence makes $\neg A$ true. Now consider a completed tableau for the inference. ν is faithful to the initial list. When we apply a rule to the list, we can, by the

Soundness Lemma, find at least one of its extensions to which ν is faithful. Similarly, when we apply a rule to this, we can find at least one of *its* extensions to which ν is faithful; and so on. By repeatedly applying the Soundness Lemma in this way, we can find a whole branch, b, such that ν is faithful to every initial section of it. (An initial section is a path from the root down the branch, but not necessarily all the way to the tip.) It follows that ν is faithful to b itself, but we do not need this fact to make the proof work. Now, if b were closed, it would have to contain some formulas of the form B and $\neg B$, and these must occur in some initial section of b. But this is impossible since ν is faithful to this section, and so it would follow that $\nu(B) = \nu(\neg B) = 1$, which cannot be the case. Hence, the tableau is open, i.e., $\Sigma \nvdash A$. ■

1.11.4 Definition: Let b be an open branch of a tableau. The interpretation *induced* by b is any interpretation, ν, such that for every propositional parameter, p, if p is at a node on b, $\nu(p) = 1$, and if $\neg p$ is at a node on b, $\nu(p) = 0$. (And if neither, $\nu(p)$ can be anything one likes.) This is well defined, since b is open, and so we cannot have both p and $\neg p$ on b.

1.11.5 Completeness Lemma: Let b be an open complete branch of a tableau. Let ν be the interpretation induced by b. Then:

if A is on b, $\nu(A) = 1$
if $\neg A$ is on b, $\nu(A) = 0$

Proof:
The proof is by induction on the complexity of A. If A is a propositional parameter, the result is true by definition. If A is complex, it is of the form $B \wedge C$, $B \vee C$, $B \supset C$, $B \equiv C$, or $\neg B$. Consider the first case, and suppose that $B \wedge C$ is on b. Since b is complete, the rule for conjunction has been applied to it. Hence, both B and C are on the branch. By induction hypothesis, $\nu(B) = \nu(C) = 1$. Hence, $\nu(B \wedge C) = 1$, as required. Next, suppose that $\neg(B \wedge C)$ is on b. Since the rule for negated conjunction has been applied to it, either $\neg B$ or $\neg C$ is on the branch. By induction hypothesis, either $\nu(B) = 0$ or $\nu(C) = 0$. In either case, $\nu(B \wedge C) = 0$, as required. The cases for the other binary connectives are similar. For $\neg$: suppose that $\neg B$ is on b. Then, since the result holds for B, by the induction hypothesis, $\nu(B) = 0$. Hence, $\nu(\neg B) = 1$. If $\neg\neg B$ is on b, then so is B, by the rule for double negation. By induction hypothesis, $\nu(B) = 1$, so $\nu(\neg B) = 0$. ■

1.11.6 COMPLETENESS THEOREM: For finite Σ, if $\Sigma \models A$ then $\Sigma \vdash A$.

Proof:
We prove the contrapositive. Suppose that $\Sigma \nvdash A$. Consider a completed open tableau for the inference, and choose an open branch. The interpretation that the branch induces makes all the members of Σ true, and A false, by the Completeness Lemma. Hence, $\Sigma \not\models A$. ■

1.12 History

The propositional logic described in this chapter was first formulated by Frege in his *Begriffsschrift* (translated in Bynum, 1972) and Russell (1903). Semantic tableaux in the form described here were first given in Smullyan (1968). The issue of how to understand the conditional is an old one. Disputes about it can be found in the Stoics and in the Middle Ages. Some logicians at each of these times endorsed the material conditional. For an account of the history, see Sanford (1989). The defence of the material conditional in terms of conversational rules first seems to have been suggested by Ajdukiewicz (1956). The idea was brought to prominence by Grice (1989, chs. 1–4). The argument for distinguishing between the indicative and subjunctive conditionals was first given by Adams (1970). The examples of 1.9 are taken from a much longer list given by Cooper (1968). The argument of 1.10 was given by Faris (1968).

1.13 Further Reading

For an introduction to classical logic based on tableaux, see Jeffrey (1991), Howson (1997) or Restall (2006). For a number of good papers discussing the connection between material, indicative and subjunctive conditionals, see Jackson (1991). For further discussion of the examples of 1.9, see Routley, Plumwood, Meyer and Brady (1982, ch. 1).

1.14 Problems

1. Check the truth of each of the following, using tableaux. If the inference is invalid, read off a counter-model from the tree, and check directly that it makes the premises true and the conclusion false, as in 1.5.4.

(a) $p \supset q, r \supset q \vdash (p \vee r) \supset q$
(b) $p \supset (q \wedge r), \neg r \vdash \neg p$
(c) $\vdash ((p \supset q) \supset q) \supset q$
(d) $\vdash ((p \supset q) \wedge (\neg p \supset q)) \supset \neg p$
(e) $p \equiv (q \equiv r) \vdash (p \equiv q) \equiv r$
(f) $\neg(p \supset q) \wedge \neg(p \supset r) \vdash \neg q \vee \neg r$
(g) $p \wedge (\neg r \vee s), \neg(q \supset s) \vdash r$
(h) $\vdash (p \supset (q \supset r)) \supset (q \supset (p \supset r))$
(i) $\neg(p \wedge \neg q) \vee r, p \supset (r \equiv s) \vdash p \equiv q$
(j) $p \equiv \neg\neg q, \neg q \supset (r \wedge \neg s), s \supset (p \vee q) \vdash (s \wedge q) \supset p$

2. Give an argument to show that $A \models B$ iff $\models A \supset B$. (Hint: split the argument into two parts: left to right, and right to left. Then just apply the definition of $\models$. You may find it easier to prove the contrapositives. That is, assume that $\not\models A \supset B$, and deduce that $A \not\models B$; then vice versa.)
3. How, if at all, could one defend or attack the arguments of 1.7, 1.8 and 1.9?
4. *Check the details omitted in 1.11.2 and 1.11.5.
5. *Use the Soundness and Completeness Lemmas to show that if one completed tableau for an inference is open, they all are. Infer that the result of a tableau test is indifferent to the order in which one lists the premises of the argument and applies the tableau rules.