To appear in Volume 3, Foundations of the Pure Sciences, in History and Philosophy of Science for Science Students Committed to Africa, Helen Lauer (Editor), Ghana University Press, Lagon, Ghana.

Chapter

49

An Introduction to Classical Propositional Logic: Syntax, Semantics, Sequents

by

RAJEEV GORÉ

Figure 1 from page 11.
Truth Tables for the Semantics of Classical Propositional Logic

Table 1(a)	
φ	$\neg\varphi$
t	**f**
f	**t**

Table 1(b)					
φ	ψ	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\varphi \rightarrow \psi$	$\varphi \leftrightarrow \psi$
t	**t**	**t**	**t**	**t**	**t**
t	**f**	**f**	**t**	**f**	**f**
f	**t**	**f**	**t**	**t**	**f**
f	**f**	**f**	**f**	**t**	**t**

Figure 2 from page 24.
Sequent Calculus SK for Classical Propositional Logic

$$(\mathrm{Id}) \quad \Gamma, \varphi \Longrightarrow \varphi, \Delta$$

$$(\neg l) \quad \frac{\Gamma \Longrightarrow \varphi, \Delta}{\Gamma, \neg\varphi \Longrightarrow \Delta} \qquad (\neg r) \quad \frac{\Gamma, \varphi \Longrightarrow \Delta}{\Gamma \Longrightarrow \neg\varphi, \Delta}$$

$$(\wedge l) \quad \frac{\Gamma, \varphi, \psi \Longrightarrow \Delta}{\Gamma, \varphi \wedge \psi \Longrightarrow \Delta} \qquad (\wedge r) \quad \frac{\Gamma \Longrightarrow \varphi, \Delta \quad \Gamma \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \varphi \wedge \psi, \Delta}$$

$$(\vee l) \quad \frac{\Gamma, \varphi \Longrightarrow \Delta \quad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \varphi \vee \psi \Longrightarrow \Delta} \qquad (\vee r) \quad \frac{\Gamma \Longrightarrow \varphi, \psi, \Delta}{\Gamma \Longrightarrow \varphi \vee \psi, \Delta}$$

$$(\rightarrow l) \quad \frac{\Gamma \Longrightarrow \varphi, \Delta \quad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \varphi \rightarrow \psi \Longrightarrow \Delta} \qquad (\rightarrow r) \quad \frac{\Gamma, \varphi \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \varphi \rightarrow \psi, \Delta}$$

$$(\leftrightarrow l) \quad \frac{\Gamma \Longrightarrow \varphi, \psi, \Delta \quad \Gamma, \varphi, \psi \Longrightarrow \Delta}{\Gamma, \varphi \leftrightarrow \psi \Longrightarrow \Delta}$$

$$(\leftrightarrow r) \quad \frac{\Gamma, \varphi \Longrightarrow \psi, \Delta \quad \Gamma, \psi \Longrightarrow \varphi, \Delta}{\Gamma \Longrightarrow \varphi \leftrightarrow \psi, \Delta}$$

1 Introduction

For over 2000 years we have tried to understand and model how we reason. An important aspect of human reasoning is our ability to distinguish good arguments from bad ones, and our ability to deduce new information from our current knowledge. One of the most significant theories of deduction is based upon the Western traditions initiated by Aristotle and is called Classical Propositional Logic. This chapter is a very elementary introduction to classical propositional logic assuming some very rudimentary knowledge of sets and high school algebra.

There are many schools of thought that decry formal logic as a model for human reasoning for it is not at all clear that we actually do use formal logic in our brains [JL83]. Indeed, we often make logical mistakes, but we see these mistakes most clearly when the explanation makes use of logic. Although classical propositional logic is not widely accepted as a theory of human intelligence, it has proved extremely useful in many areas where a formal description of a situation is required, particularly in areas like Hardware Verification and Software Verification where computers are used to verify that digital circuits or computer programs meet their intended purpose.

Classical propositional logic originated from attempts to distill the logical rigour inherent in a certain finite collection of natural language expressions like "and", "or", "not", "if _ then _", and "_ if and only if _" which we use to tie together simpler (English) expressions into larger arguments or narratives. Consider, for example, the following two arguments:

(1) Hindus are vegetarians and if Rajeev is a Hindu then he is a vegetarian.

(2) Hindus are vegetarians and if Rajeev is a vegetarian then he is a Hindu.

The first makes logical sense as it is not possible to find a way for Rajeev to be both a Hindu and a non-vegetarian. The second is suspect, for Rajeev might be a vegetarian because he does not like the way animals are farmed for meat, or even because he does not like the taste of meat. Thus there is a way for Rajeev to be a vegetarian non-Hindu. Classical propositional logic is an attempt to give a rigorous method for expressing such reasoning.

If we want to insist on rigour then most natural languages like English are not suitable as a formalism because they are too long-winded, imprecise and sometimes even ambiguous. Consider, for example, the following sentence "If statement one is true and if statement one is true then statement two is true is true then statement two is true.". Can you tell whether the quoted sentence is true or false? If you answered "true" then you are well on your way to understanding logic, but if you answered "false" or "I don't know" then it is probably because you became entangled in the words. As another example consider the two sentences "He is a kind man." and "He is not an unkind man.". Do the two sentences have the same meaning? It depends

upon whether you believe that kindness is a black and white quality or whether there are degrees of kindness.

After some two thousand years of study we have found three important aspects of any logic:

Syntax: the syntax of a logic is a formal unambiguous language, distinct from natural language, which specifies the legal expressions of the logic. By using a separate syntax we can intermingle logical expressions with (English) prose without causing confusion.

Semantics: the semantics of a logic is a method for assigning meanings to the expressions that can be built using the logical syntax. The most important role of such a semantics is to define the notion of **logical consequence**, telling us when a particular given expression follows logically from a collection of given expressions.

Calculi: a calculus for a logic is a *purely syntactic* method for defining the notion of a **deduction**, telling us when a given expression is deducible from a collection of given expressions. By "purely syntactic" we mean that the rules of the calculus must manipulate only the symbols from the syntax, and possibly some other special symbols, without explicitly using the meaning assigned to these symbols by the semantics.

If we have a syntax and a semantics then the logic is said to be defined semantically. If we have a syntax and a calculus then the logic is said to be defined syntactically. If we have all three, then it is imperative to check that an expression is a logical consequence of a collection of given expressions exactly when there is a deduction of it from the same expressions, thereby guaranteeing that these semantic and syntactic aspects of the logic agree. If, in addition, the semantic or the syntactic aspect can be implemented on a computer, then we can determine whether or not a given expression follows logically from a given collection of expressions *automatically*.

This chapter is an introduction to an existing logic called Classical Propositional Logic. In Section 2 we will see its syntax. In Section 3 we will see its semantics. In Section 4 we will see a (sequent) calculus for calculating whether or not there is a deduction of a given expression from a collection of logical expressions. In Section 5 we will see that a given expression is a logical consequence of a given set of expressions exactly when there is deduction of the given expression from the given set of expressions. In Section 5.4 we shall also see that the calculus does indeed give an algorithm for computing logical consequence in classical propositional logic on a computer automatically. In Section 6 we discuss sequent calculi and in Section 7 we show how they can be used to prove results about classical propositional logic itself.

Once you have grasped these various aspects of classical propositional logic, you might want to think about how well it captures human reasoning, but while learning it, you are asked to accept it as it is.

2 Syntax

In this section we shall give a recipe for constructing legal expressions in the formal language of classical propositional logic. As in English, the language is made up of words and these are formed into sentences. In English, we form sentences from words by writing them according to the grammar of English. We shall do the same using the grammar of our logic once it is defined, but instead of building words from the twenty-six letters of the English alphabet, we shall use a very specific (infinite) set of predefined words. Let us begin.

Each member of the infinite list $p_0, p_1, p_2, p_3, \cdots$ is an **atomic word**. The list is infinite because the dots are supposed to stand for all the other atomic words that can be formed by attaching any natural number as a subscript to the English letter p, since we cannot possibly list them all in this book. These atomic words are the words of our logical language and the adjective "atomic" means that we are not allowed to break down these words any further (into letters). The atomic word p_0 is pronounced as "p zero", the atomic word p_1 as "p one", and so on.

Example 2.1. So p_{44} is an atomic word since it would appear in the list if we were to expand the dots. But p (without a subscript) and q_5 and *kangaroo* and $elephant_5$ are not atomic words since they can never appear in this list, no matter how far we expand the dots.

In English we form words into sentences using connecting words like "and" and "or". In classical logic we use special symbols for a particular collection of such connecting words and use these special symbols to form sentences from our predefined collection of atomic words. Each member of the finite list $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$ is a logical connective, or **connective** for short.

A similar situation arises in simple arithmetic where we form expressions of arithmetic like $1 \times ((-3) + 2)$ from natural numbers (words) like 0, 1, 2, $\cdots$ using arithmetic connectives like $+$, $\times$ and $-$, where the dots in this case stand for the infinite sequence of natural numbers. In arithmetic we know that each of $\times$, $-$ and $+$ usually takes two arguments, one before the sign and one after. We also know that $-$ can be added just before an expression to make its argument negative as in (-3). But we know that both $(2\times)$ and $(\times 3)$ are not legal arithmetic expressions.

We now give the grammatical rules, or **formation rules**, for constructing logical sentences from our atomic words using our logical connectives. But instead of calling them sentences, we shall call them **formulae**. We also use the parentheses "(" and ")" to disambiguate the extent of smaller formulae just as we do in arithmetic.

Atomic Formulae: Every atomic word is an atomic formula. For example both p_{13} and p_{27} are atomic formulae since each of p_{13} and p_{27} are

atomic words: each would appear in our list of atomic words if we expanded the dots far enough.

Negated Formulae: A negated formula is formed by writing $\neg$ immediately to the left of a formula. The connective $\neg$ is an attempt to capture the logical content of "not" so $\neg p_0$ is pronounced "not p_0".

Conjunctive Formulae: A conjunctive formula is formed by writing $\wedge$ between two formulae. The connective $\wedge$ is an attempt to capture the logical content of "and" so $p_0 \wedge p_1$ is pronounced "p_0 and p_1".

Disjunctive Formulae: A disjunctive formula is formed by writing $\vee$ between two formulae. The connective $\vee$ is an attempt to capture the logical content of "or" so $p_0 \vee p_1$ is pronounced "p_0 or p_1".

Conditional Formulae: A conditional formula is formed by writing $\rightarrow$ between two formulae. The connective $\rightarrow$ is an attempt to capture the logical content of "if _ then _" so $p_0 \rightarrow p_1$ is pronounced "if p_0 then p_1". It is also referred to as implication with the reading that $p_0 \rightarrow p_1$ stands for "p_0 implies p_1".

Bi-Conditional Formulae: A bi-conditional formula is formed by writing $\leftrightarrow$ between two formulae. The connective $\leftrightarrow$ is an attempt to capture the logical content of "_ if and only if _" so $p_0 \leftrightarrow p_1$ is pronounced "p_0 if and only if p_1". It is also referred to as "logical equivalence" with $p_0 \leftrightarrow p_1$ read as "p_0 is logically equivalent to p_1".

Example 2.2. The expressions $\wedge p_{27}$ and $p_{13}\neg$ and $p_{13} \wedge \wedge p_{27}$ are not formulae because:

$\wedge p_{27}$ does not have a formula as a left component for the $\wedge$ connective;

$p_{13}\neg$ has the negation sign to the right of a formula whereas it must appear to the left of a formula;

$p_{13} \wedge \wedge p_{27}$ contains $\wedge$ connectives which are not sandwiched between two formulae since neither $p_{13}\wedge$ nor $\wedge p_{27}$ are formulae.

The examples given above like $p_0 \wedge p_1$ all have atomic formulae to the right and left of the connectives, but the rules also allow us to build arbitrary complex formulae like $(p_{13} \wedge (p_{13} \rightarrow p_{27})) \rightarrow \neg\neg p_{27}$. Notice that we use the parentheses "(" and ")" to disambiguate the extent of smaller formulae just as we do in arithmetic.

Exercise 2.1. Compare and contrast the following *different* formulae:

1. $(p_{13}\wedge(p_{13} \rightarrow p_{27})) \rightarrow \neg\neg p_{27}$ is a conjunctive formula where $p_{13}\wedge(p_{13} \rightarrow p_{27})$ is the left component and $\neg\neg p_{27}$ is the right component.

2. $p_{13} \wedge (p_{13} \rightarrow (p_{27} \rightarrow \neg\neg p_{27}))$ is a conjunctive formula where p_{13} is the left component and $p_{13} \rightarrow (p_{27} \rightarrow \neg\neg p_{27})$ is the right component.

3. $p_{13} \wedge ((p_{13} \rightarrow p_{27}) \rightarrow \neg\neg p_{27})$ is a conjunctive formula where p_{13} is the left component and $(p_{13} \rightarrow p_{27}) \rightarrow \neg\neg p_{27}$ is the right component.

4. $(p_{13} \wedge p_{13}) \rightarrow (p_{27} \rightarrow \neg\neg p_{27})$ is a conditional formula where $p_{13} \wedge p_{13}$ is the left component and $p_{27} \rightarrow \neg\neg p_{27}$ is the right component.

5. $((p_{13} \wedge p_{13}) \rightarrow p_{27}) \rightarrow \neg\neg p_{27}$ is a conditional formula where $(p_{13} \wedge p_{13}) \rightarrow p_{27}$ is the left component and $\neg\neg p_{27}$ is the right component.

Example 2.3. The formulae $(p_0 \wedge p_1) \wedge p_2$ and $p_0 \wedge (p_1 \wedge p_2)$ are different purely because of the placement of the parentheses.

Every non-atomic formula can be categorised into one and only one of the five non-atomic formula categories: negative formula, conjunctive formula, disjunctive formula, conditional formula, and bi-conditional formula. The connective that determines the category of a formula is called its **main** connective.

Exercise 2.2. Find the main connective of each formula from Exercise 2.1.

The same formula may be written in many different ways by using spurious parentheses.

Example 2.4. The formulae $(p_0 \wedge p_1) \wedge p_2$ and $((p_0) \wedge (p_1)) \wedge (p_2)$ are the same formula since their main connective is $\wedge$ and the two right components p_2 and (p_2) are clearly the same formula and the two left components $p_0 \wedge p_1$ and $(p_0) \wedge (p_1)$ are also clearly the same formula.

Example 2.5. The expression $p_{13} \wedge p_{13} \rightarrow p_{27} \rightarrow \neg\neg p_{27}$ is *ambiguous* since it cannot be categorised uniquely: there are five different ways to categorise it by utilising parentheses; see Exercise 2.1.

In the rest of this chapter, we will often wish to refer to a formula of a particular shape, for example one having the $\wedge$ connective as its main connective. We could use the term conjunctive formula in this particular case, but we may also want to refer to the formulae to the right and left of the $\wedge$ connective. To make this task easier we shall use the Greek letters φ and ψ, pronounced as "phi" and "psi" respectively, as **names** for arbitrary formulae. Then we can refer to an arbitrary conjunctive formula as the shape $\varphi \wedge \psi$ since this conveys precisely that the main connective of this formula is $\wedge$, that the (name of the) formula to the left is φ, and that the (name of the) formula to the right is ψ. We do not know, nor care, what φ

and ψ actually look like. Sometimes we need more than just the two names φ and ψ so we shall add subscripts and use names like φ_1 and ψ_1.

Sometimes we shall want to refer to an arbitrary atomic formula. We will simply use p or q or r to stand for (the names of) such atomic formulae.

In general, we say just "formula φ" instead of the more pedantic "formula with name φ" and say "atomic formula p" instead of the more pedantic "atomic formula with name p". It may seem as if we are now contradicting Example 2.1 where we said that p itself was *not* an atomic word and hence not a formula. But there is no contradiction, for p is the *name* of an unspecified atomic formula, it is not an atomic formula itself.

Example 2.6. The formula (with shape) $p \wedge q$ cannot contain further connectives since p and q must be (names for) atomic formulae. On the other hand, the formula (with shape) $\varphi \wedge \psi$ can contain further connectives since the formula (with name) φ could be $(p_1 \vee p_2) \rightarrow p_3$ and the formula (with name) ψ could be $\neg\neg p_{55}$ so that then the formula $\varphi \wedge \psi$ would be $((p_1 \vee p_2) \rightarrow p_3) \wedge \neg\neg p_{55}$.

Using this new notation, the previous formation rules can be replaced by one single **formation rule**: each atomic formula p is a formula, and if φ and ψ are formulae, then so are each of $\neg(\varphi)$, $(\varphi) \wedge (\psi)$, $(\varphi) \vee (\psi)$, $(\varphi) \rightarrow (\psi)$, and $(\varphi) \leftrightarrow (\psi)$.

The new notation allows us to add the parentheses in the formation rule itself rather than as an ancillary device. But too many parentheses tend to clutter the formula and make it unreadable: compare the two ways of writing the same formula below:

$$(p_{13} \wedge (p_{13} \rightarrow p_{27})) \rightarrow \neg\neg p_{27} \qquad ((p_{13}) \wedge ((p_{13}) \rightarrow (p_{27}))) \rightarrow (\neg(\neg(p_{27}))).$$

We therefore try to keep the use of parentheses to a minimum.

Exercise 2.3. Compare this succinct version with the formation rules we gave previously to ensure that they specify the same collection of formulae.

Subformulae: The **strict subformulae** of a formula φ are all the formulae from which φ is built up. The **subformulae** of φ are all its strict subformulae plus φ itself. We say that φ is an **immediate subformula** of $\neg\varphi$, and that φ and ψ are both **immediate subformulae** of each of $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$ and $\varphi \leftrightarrow \psi$.

Example 2.7. For the formula $(p_0 \wedge (p_0 \rightarrow p_1)) \rightarrow p_1$: the strict subformulae are $p_0 \wedge (p_0 \rightarrow p_1)$, $p_0 \rightarrow p_1$, p_0 and p_1; the subformulae are $(p_0 \wedge (p_0 \rightarrow p_1)) \rightarrow p_1$, $p_0 \wedge (p_0 \rightarrow p_1)$, $p_0 \rightarrow p_1$, p_0 and p_1; and the immediate subformulae are $p_0 \wedge (p_0 \rightarrow p_1)$ and p_1.

Formula Length: Every atomic formula has length 1. If φ has length n then $\neg\varphi$ has length $n+1$. If φ has length n and ψ has length m then each of $\varphi \wedge \psi$ and $\varphi \vee \psi$ and $\varphi \to \psi$ and $\varphi \leftrightarrow \psi$ has length $n+m+1$. Thus the length of a formula is simply the number of symbols from which it is composed, without counting parenthesis.

Example 2.8. The length of $(p_0 \wedge (p_0 \to p_1)) \to p_1$ is 7.

We have just defined the syntax (language) of classical propositional logic: it is simply the set of all formulae we can form using the formation rules. The syntax we have chosen is fairly standard but here are some alternative symbols that are sometimes used instead of the symbols we have used:

Our symbol	$\neg$		$\wedge$	$\vee$		$\to$		$\leftrightarrow$	
Alternatives	$\sim$	$-$	&	$\mid$	$+$	$\supset$	$\Rightarrow$	$\equiv$	$\Leftrightarrow$

Some authors use the letters A, B and C as names for formulae instead of our φ and ψ, and use P and Q as names for atomic formulae instead of our p and q. The formula $(\neg(\varphi \vee \psi)) \to p$, for example, might appear in some books as $(\sim (A + B)) \Rightarrow Q$.

Some authors also allow two special atomic symbols called the verum and falsum constants into the syntax, and typically use $\top$ or $\mathbf{tt}$ and $\bot$ or $\mathbf{ff}$ respectively as symbols for these constants. Then, expressions like $\bot \to \top$ are also formulae. We have not included this slight complication as it makes no difference at our level of study of classical propositional logic.

But we still do not have a way to give these formulae any meaning. For example, what exactly does the formula $(p_{13} \wedge (p_{13} \to p_{27})) \to \neg\neg p_{27}$ mean? In the next section, we shall see one way to give meanings to formulae.

3 Semantics of Classical Propositional Logic

In the previous section we saw how to invent a very particular formal language which defined the class of formulae of classical propositional logic. In this section we shall first see how we can give a meaning to the individual symbols that make up that language. Then we shall move on to deeper semantic notions like satisfiability, validity and logical consequence.

3.1 Truth Values, Interpretations, and Truth Tables

The meaning of the symbols is built up by giving meanings to the atomic formulae of our syntax and using our intuitions about the meaning of the linguistic connectives "not", "and", "or", "if _ then _", and "_ if and only if _" to build up the meanings of larger formulae. Thus there are two parts to the semantics of classical propositional logic: the first part is an assignment

of meanings to the atomic formulae, and the second part is a recipe for calculating the meaning of larger formulae from those of smaller formulae.

But the meaning of an atomic formula like p_1 is completely subjective. One person might interpret p_1 to mean "it is raining" while another person might interpret p_1 to mean "I have a sister". Indeed, every person on Earth could interpret p_1 in a completely different way! So how can we possibly judge whether a certain given formula follows logically from another given collection of formulae?

The trick is to notice that most subjective interpretations will allow a person to assign a truth value to p_1. For example, if I interpret p_1 to mean "I have a sister" then I will assign a truth value of "true" to p_1 since I do indeed have a sister. On the other hand, if I interpret p_1 to mean "it will rain tomorrow" then I will assign a truth value of "unknown" to p_1 since I do not currently know whether or not it will rain tomorrow. Of course, there are many other possible truth values we could think of: "possibly", "probably", "maybe", "indeterminate" to name a few. There are also interpretations where no truth value is given to p_1: for example if I interpret p_1 as the request "please close the door" then it has no obvious truth value. How can we possibly take all these alternatives into account?

The simple answer is that we cannot. So, in classical propositional logic, we make the following two simplifying assumptions:

Law of the Excluded Middle: each formula is "true" or "false".

Law of Non-Contradiction: no formula is both "true" and "false".

The law of the excluded middle means that there are only two legal truth values: "true" and "false". Other intermediate truth values like "unknown" or "maybe" or "neither" are simply forbidden. The law of non-contradiction means that the truth value of each atomic formula is either "true" or else it is "false", never both. The result of these two laws is that each atomic formula of classical propositional logic has one of only two truth values under any one (person's) interpretation. Interpretations like "please close the door" which do not impart one of the truth values "true" or "false" to an atomic formula are simply forbidden. Note however that the interpretation of p_1 as "it will rain tomorrow" is allowed, but its truth value is restricted to either "true" or "false" rather than some other third value like "unknown".

So even if every person on Earth interpreted p_1 differently from every other person, in classical logic, there would be only two distinct classes of people: those whose interpretation of p_1 made it take the value "true", and those whose interpretation of p_1 made it take the value "false". Thus the multitude of actual subjective interpretations of p_1 are collapsed into two: "true" or "false". We now make these intuitions precise.

We use the extra symbols **t** and **f** to stand for the truth values "true" and "false" respectively. These extra symbols are *not* part of our syntax, so

Table 1(a)	
φ	$\neg\varphi$
t	**f**
f	**t**

Table 1(b)					
φ	ψ	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\varphi \rightarrow \psi$	$\varphi \leftrightarrow \psi$
t	**t**	**t**	**t**	**t**	**t**
t	**f**	**f**	**t**	**f**	**f**
f	**t**	**f**	**t**	**t**	**f**
f	**f**	**f**	**f**	**t**	**t**

Figure 1: Truth Tables for the Semantics of Classical Propositional Logic.

expressions like $\mathbf{f} \vee \mathbf{t}$ and $\mathbf{t} \wedge p_0$ are *not* formulae.

An **interpretation** is a (meaning) function which assigns one and only one of **t** and **f** to each atomic formula. Since our list of atomic formulae is infinite, every interpretation can be viewed as an infinite list of truth values with the first truth value in the list being the value for p_0, the second value in the list being the value for p_1 and so on, for ever. In real life, and especially in this chapter, we usually only need to speak about a finite number of atomic formulae, we therefore list their truth values explicitly.

Example 3.1. The interpretation **t**, **t**, **f**, **f**, **f**, $\cdots$ where the dots represent an infinite list of **f** symbols assigns a truth value of **t** to p_0 and to p_1, and assigns a truth value of **f** to all other atomic formulae.
But if we were interested only in the truth values of p_0, p_1, p_2 and p_3 then we could write this interpretation simply as shown at right with the assumption that all atomic formulae not shown explicitly take on the truth value **f** (say).

p_0	p_1	p_2	p_3
t	**t**	**f**	**f**

Recall that we are trying to give a meaning to arbitrary formulae of classical propositional logic. So how can a simple interpretation function that tells us the truth value of each atomic formula possibly tell us the truth values of *all* arbitrary formulae? Figure 1 shows the relationship between the truth values of larger formulae and their smaller constituents in classical propositional logic. We explain this truth table on a case by case basis, try to justify these semantics in terms of their linguistic origins, and mention some of the shortcomings of the semantics.

Negation: The truth value of $\neg\varphi$ is **t** when the truth value of φ is **f**, and the truth value of $\neg\varphi$ is **f** when the truth value of φ is **t**.

Note that we can use Table 1(a) in two directions: one to calculate the truth value of $\neg\varphi$ from the truth value of φ, and the other to calculate the truth value of φ from the truth value of $\neg\varphi$.

The meaning of the negation symbol $\neg$ can be read off easily from its intended meaning as the linguistic connective "not": if φ is "true" then $\neg\varphi$ must be "false", and vice versa. So if we were to interpret p_0 as "He is a kind man." then classical propositional logic forces us to interpret $\neg p_0$ as

"He is an unkind man" showing that classical propositional logic forces us to interpret kindness as a purely black and white notion: it does not allows us to say that there are degrees of kindness.

Conjunction: The truth value of the conjunction $\varphi \wedge \psi$ is $\mathbf{t}$ when the truth values of φ and ψ are both $\mathbf{t}$, and the truth value of $\varphi \wedge \psi$ is $\mathbf{f}$ otherwise.

We can always compute the truth value of $\varphi \wedge \psi$ given the truth values of φ and ψ by using Table 1(b), but we cannot use Table 1(b) the other way in all cases. If we know, for example, that the truth value of $\varphi \wedge \psi$ is $\mathbf{f}$, then there are three possible ways to give truth values to φ and ψ according to the third column of Table 1(b): all we can say is that at least one of the truth values of φ and ψ must be $\mathbf{f}$.

The meaning of the conjunction symbol $\wedge$ can be read off easily from its intended meaning as the linguistic connective "and": that is, "φ is true" and "ψ is true" exactly when "φ and ψ is true".

If we interpret p_0 as "it is raining", for example, and interpret p_1 as "it is hot" then $p_0 \wedge p_1$ says "it is hot and raining".

Disjunction: The truth value of the disjunction $\varphi \vee \psi$ is $\mathbf{t}$ when the truth value of at least one of φ and ψ is $\mathbf{t}$, and the truth value of $\varphi \vee \psi$ is $\mathbf{f}$ otherwise.

Again, we can always compute the truth value of $\varphi \vee \psi$ given the truth values of φ and ψ using Table 1(b), but we cannot use this table the other way in all cases. If we know, for example, that the truth value of $\varphi \vee \psi$ is $\mathbf{t}$, then there are three possible ways to give truth values to φ and ψ according to the fourth column of Table 1(b): all we can say is that at least one of the truth values of φ and ψ must be $\mathbf{t}$.

The meaning of the disjunction symbol $\vee$ can be read off easily from its intended meaning as the linguistic connective "or": that is, "φ is true" or "ψ is true" exactly when "φ or ψ is true". But it is not quite as easy as it was for conjunction, for there are two ways to read "φ or ψ is true": namely as "inclusive or" or as "exclusive or".

For example, if I am asked "Would you like tea or coffee?" then I would choose between "tea" and "coffee" since this "or" is exclusive: I can have one or the other but not both. But if am asked "Would you like milk or sugar" then I would answer "both" as this "or" is inclusive: I can have one or the other or both.

In classical propositional logic we define "or" to be "inclusive or". The easiest way to say this is to define the truth value of a disjunction to be "false" exactly when both components of the disjunction are "false".

Conditional: The truth value of the conditional $\varphi \to \psi$ is $\mathbf{t}$ if the truth value of φ is $\mathbf{f}$ or the truth value of ψ is $\mathbf{t}$ or both, and the truth value of

$\varphi \to \psi$ is **f** otherwise.

Knowing the truth values of φ and ψ allows us to compute the truth value of $\varphi \to \psi$. But knowing the truth value of $\varphi \to \psi$ does not uniquely determine the truth values of φ and ψ.

The conditional connective $\to$ is one of the hardest to explain from an intuitive sense in a totally satisfactory manner given its intended meaning of "if φ is true then ψ is true" or as "φ implies ψ".

It is fairly easy to see that $\varphi \to \psi$ should be "false" when φ is "true" and ψ is "false". It is also easy to see that $\varphi \to \psi$ should be "true " when φ is "true" and ψ is "true". But what should we do when φ is "false"?

Some philosophers have argued that in this case, the formula $\varphi \to \psi$ should take an "unknown" or "indeterminate" value. Since these truth values are forbidden by the Law of Excluded Middle, we must assign either "true" or "false" to $\varphi \to \psi$ even when φ is "false". In classical propositional logic we put $\varphi \to \psi$ to "true" in the two cases where φ is "false".

If, for example, we interpret p_0 as "you will eat your dinner" and interpret p_1 as "you will get dessert" then $p_0 \to p_1$ says "if you will eat your dinner then you will get dessert": a conditional that most children easily bypass by making enough noise to get the dessert without eating dinner. On the other hand, if we interpret p_0 as "the moon is made of green cheese" and interpret p_1 as "pigs can fly", then $p_0 \to p_1$ says "if the moon is made of green cheese then pigs can fly", a statement that tells us nothing since we know the moon is not made of green cheese.

Bi-Conditional: The truth value of the bi-conditional $\varphi \leftrightarrow \psi$ is **t** when the truth values of the formulae φ and ψ are the same, and the truth value of $\varphi \leftrightarrow \psi$ is **f** when they are different.

Knowing the truth values of φ and ψ allows us to compute the truth value of $\varphi \leftrightarrow \psi$. But knowing the truth value of $\varphi \leftrightarrow \psi$ does not uniquely determine the truth values of φ and ψ.

The bi-conditional $\varphi \leftrightarrow \psi$ simply says that the formulae φ and ψ must take the same truth value, whichever of **t** and **f** it is.

If, for example, we interpret p_0 as "you will eat your dinner" and interpret p_1 as "you will get dessert" then $p_0 \leftrightarrow p_1$ says "you will eat your dinner if and only if you will get dessert". Unlike $p_0 \to p_1$, no amount of noise will allow a child to bypass $p_0 \leftrightarrow p_1$ since it states that the truth values of "you will eat your dinner" and "you will get dessert" must be the same.

Example 3.2. Let us compute the possible truth values for the formula $(\neg p_0) \vee p_1$ under an interpretation where p_0 and p_1 are both given a truth value of **t**. We write **t** under each of them as shown at left to indicate that p_0 has truth value **t** and p_1 has truth value **t**. The formula $\neg p_0$ has the shape $\neg\varphi$ if we let φ be the (name for the) formula p_0. Table 1(a)

$(\neg$	$p_0)$	$\vee$	p_1
	t		**t**

of Figure 1 tells us that the truth value of $\neg\varphi$ is **f** when the truth value of φ is **t**. That is, the truth value of $\neg p_0$ is **f** when the truth value of p_0 is **t**, so we extend the picture by writing **f** under the $\neg$ connective as shown at left to indicate that $\neg p_0$ is **f**. The formula $(\neg p_0) \vee p_1$ has the same shape as the formula $\varphi \vee \psi$ if we let φ be $\neg p_0$ and ψ be p_1. Table 1(b) tells us that the formula $\varphi \vee \psi$ has truth value **t** if one or both of φ and ψ has truth value **t**. That is, the formula $(\neg p_0) \vee p_1$ has truth value **t** if one or both of $\neg p_0$ and p_1 has truth value **t**. As p_1 has truth value **t**, we therefore know that $(\neg p_0) \vee p_1$ has truth value **t**, which we write by putting a **t** under the $\vee$ connective as shown at left. Thus when both p_0 and p_1 have truth value **t**, we know that $(\neg p_0) \vee p_1$ has truth value **t**.

$(\neg$	$p_0)$	$\vee$	p_1
f	**t**		**t**

$(\neg$	$p_0)$	$\vee$	p_1
f	**t**	**t**	**t**

Exercise 3.1. Repeat this exercise for the other three cases (rows) below:

$(\neg$	$p_0)$	$\vee$	p_1
f	**t**	**t**	**t**
?	**t**	?	**f**
?	**f**	?	**t**
?	**f**	?	**f**

In this way, any given interpretation can be extended to give a truth value to any formula of classical propositional logic.

We now need to ensure that these semantics do indeed obey the Law of the Excluded Middle and the Law of Non-Contradiction. To indicate that this property is of fundamental importance we call it a theorem and give a proof for it in English prose.

Theorem 3.1. Under any given interpretation,

Excluded Middle: every formula takes the value **t** or the value **f**

Non-Contradiction: no formula takes the value **t** and **f**.

Proof: Consider any interpretation and consider any formula. Since we know nothing about the formula, we proceed by considering all the various shapes it could take.

atomic: If the formula is atomic, then the given interpretation must assign it a truth value of either **t** or **f**, but not both since this is precisely what interpretations do by their very definition. Thus, under this given interpretation, both the Law of the Excluded Middle and the Law of Non-Contradiction are satisfied for all atomic formulae.

non-atomic: If the formula is non-atomic then it must have one of the shapes $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, $\varphi \leftrightarrow \psi$, for strict subformulae φ and ψ, since these are the only legal shapes for formulae. We next show that, under this given interpretation, the theorem holds for each formula shape if it holds for its immediate subformulae. We therefore have to consider each of these cases in turn. We consider the first two cases and leave the others as exercises.

$\neg\varphi$: Suppose the given formula has shape $\neg\varphi$. Further suppose that, under the given interpretation, the immediate subformula φ obeys Theorem 3.1, regardless of what shape φ itself has. The truth value of φ is therefore either **t** or **f**, but not both. Table 1(a) from Figure 1 then tells us that, under the given interpretation, the truth value of $\neg\varphi$ itself is either **f** or **t**, respectively, but not both. That is, both the Law of the Excluded Middle and the Law of Non-contradiction hold for $\neg\varphi$ if they hold for φ.

$\varphi \wedge \psi$: Suppose the given formula has shape $\varphi \wedge \psi$. Suppose further that, under the given interpretation, the immediate subformulae φ and ψ obey Theorem 3.1, regardless of their own shapes. That is, suppose that both the Law of the Excluded Middle and the Law of Non-contradiction hold for φ and for ψ. Thus the truth value of φ is either **t** or **f**, but not both, and the truth value of ψ is either **t** or **f**, but not both. There are four possible pairings of the possibilities, but Table 1(b) from Figure 1 tells us that, in each of these four pairings, the truth value of $\varphi \wedge \psi$ itself is either **t** or **f**, but not both. That is, both the Law of the Excluded Middle and the Law of Non-contradiction hold for $\varphi \wedge \psi$ if they hold for φ and ψ individually.

In general, we have shown that the theorem will hold for a formula if it holds for all smaller formulae, and also shown that the theorem holds for all the smallest formulae: namely all atomic formulae. Under this given interpretation, the theorem therefore must hold for all formulae.

Our proof does not assume nor demand any particular properties of the given interpretation: it can be any arbitrary interpretation. This means that the theorem must hold for any and all interpretations. **Q.E.D.**

We mark the end of a proof with the letters **Q.E.D.** which stand for the Latin words *quod erat demonstrandum* which roughly translate to "which was to be demonstrated".

Intuition:[1] The technical argument in the proof above can be summarised as follows: each interpretation by its definition respects the two laws for

[1] [Not used here in the technical sense of "intuitionism" discussed in Chapter 51.—Ed.]

atomic formulae, and the truth tables in Figure 1 then enforce these two laws in defining how to calculate the truth value of larger formulae from smaller formulae.

Exercise 3.2. Complete the arguments for the cases from the proof of Theorem 3.1 where the given formula has the shape $\varphi \vee \psi$, $\varphi \rightarrow \psi$, and $\varphi \leftrightarrow \psi$.

Exercise 3.3. Work out the truth values for the formulae $p_0 \rightarrow p_1$ and $(\neg p_0) \vee p_1$ and $((\neg p_0) \vee p_1) \leftrightarrow (p_0 \rightarrow p_1)$ given the four possible pairings of the truth values of p_0 and p_1. Below we have given the case where p_0 and p_1 both have truth value **t**:

p_0	p_1	$p_0 \rightarrow p_1$	$(\neg p_0) \vee p_1$	$((\neg p_0) \vee p_1) \leftrightarrow (p_0 \rightarrow p_1)$
t	**t**	**t**	**t**	**t**
t	**f**	?	?	?
f	**t**	?	?	?
f	**f**	?	?	?

Notice that the truth values of $p_0 \rightarrow p_1$ and $(\neg p_0) \vee p_1$ are always equal: such formulae are said to be **logically equivalent** to each other. Notice also that the truth value of $((\neg p_0) \vee p_1) \leftrightarrow (p_0 \rightarrow p_1)$ is always **t** since the meaning of the connective $\leftrightarrow$ is supposed to be "_ is logically equivalent to _".

3.2 Satisfiability, Validity and Logical Consequence

Having seen how to give meaning to the syntax of classical propositional logic via interpretations, we now move on to more complicated semantic notions. We say that a formula is:

satisfiable: if there is at least *one* interpretation which gives it a value **t**

falsifiable: if there is at least *one* interpretation which gives it a value **f**

unsatisfiable: if *every* interpretation gives it a value **f**

valid: if *every* interpretation gives it a value **t**.

Example 3.3. The formula $p_0 \rightarrow p_1$ is satisfiable since any interpretation where p_0 is **f** gives $p_0 \rightarrow p_1$ the truth value **t**. Check this using the method outlined in Example 3.2.

Example 3.4. The formula $p_0 \rightarrow p_1$ is also falsifiable since any interpretation where p_0 is **t** and p_1 is **f** gives $p_0 \rightarrow p_1$ the truth value **f**. Check this using the method outlined in Example 3.2.

Thus there exist formulae which are both satisfiable and falsifiable and these terms are not opposites.

To show that a formula is unsatisfiable or to show that it is valid apparently requires us to inspect every possible interpretation. Since there are an infinite number of interpretations, how can we possibly achieve this task?

Example 3.5. Consider the formula $(\neg p_0) \vee p_0$ and any interpretation at all: this interpretation must give a truth value of either **t** or **f** to p_0 which we write below p_0 in the first column in two separate rows. Table 1(a) then tells us that $\neg p_0$ has truth value **f** when p_0 has truth value **t**, and $\neg p_0$ has truth value **t** when p_0 has truth value **f**, as captured by the two rows of the second column. Under each of these possibilities, $(\neg p_0) \vee p_0$ has value **t** according to Table(1b). Since our initial interpretation was totally arbitrary, this must hold for every interpretation, and therefore, $(\neg p_0) \vee p_0$ is valid.

p_0	$\neg p_0$	$(\neg p_0) \vee p_0$
t	**f**	**t**
f	**t**	**t**

As shown by Example 3.5, the truth value of a formula is determined only by the atomic formulae that appear in it. It is easy to see, for example, that the truth value of the atomic formula p_{57} has no bearing on the truth value of the formula $(\neg p_0) \vee p_0$. But we had to consider two cases in order to determine whether or not $(\neg p_0) \vee p_0$ was valid. How many different cases do we need to consider to determine the validity of a formula φ built up from n atomic formulae, where n is a positive natural number? Let us consider the first two values of n, and generalise:

n = 1: If $n = 1$ then φ must be built up from one atomic formula only, say p_0. This does not mean that φ is itself an atomic formula, for both $p_0 \wedge \neg p_0$ and $p_0 \leftrightarrow (p_0 \vee \neg p_0)$ are built up from only one atomic formula p_0. But it should be clear from the truth tables from Figure 1 that the truth value of this larger formula is completely determined by the truth value of the atomic formula from which it is built. Thus there are only two cases for such formulae: the first where the single atomic formula p_0 has truth value **t** and the second where it has truth value **f**, which we can write in tabular form as shown at left. The truth tables will now completely determine the truth value of φ in these two cases.

p_0
t
f

n = 2: If $n = 2$ then φ is built from only two atomic formulae, say p_0 and p_1. Again, φ may be arbitrarily complex, for example both the formulae $p_0 \rightarrow p_1$ and $(\neg(p_0 \leftrightarrow p_1)) \rightarrow (p_1 \vee p_0)$ are built from only two atomic formulae. Once again, the cases we need to consider are the truth values of only these two atomic formulae since the truth value of atomic formulae like p_{16} or p_{1267} clearly do not affect the truth value of φ if they do not appear in φ. There are now four cases: two where p_0 is **t** and p_1 is **t** or **f**, and another two where

p_0	p_1
t	**t**
t	**f**
f	**t**
f	**f**

p_0 is **f** and p_1 is **t** or **f**, as shown at left.

General Case: In general, adding another atomic formula to the table doubles the number of cases. That is, if φ is built from n different atomic formulae, there are 2^n cases where $2^0 = 1$ and $2^n = 2 \times 2^{n-1}$ for $n = 1, 2, 3, \cdots$ Of course, under each one of these cases, φ itself evaluates to either **t** or **f**.

Exercise 3.4. Extend the table at left to $n = 10$ by doubling the value of 2^n for the previous entry at each stage. This shows the difficulty of using truth tables: if the formula φ is built from 20 different atomic formulae, then we have to examine $2^{10} = 1024$ different interpretations to tell if it is valid. Moreover, for a formula with just one extra atomic formula, the number of cases doubles to $2^{11} = 2048$. If n is 100 then there are $2^{100} = 1267650600228229401496703205376$ cases!

n	2^n
1	2
2	4
3	8
$\cdots$	$\cdots$
10	1024

Example 3.6. Show that an arbitrary formula φ cannot be both satisfiable and unsatisfiable. If φ is satisfiable then there is some interpretation that makes it **t**. If φ is unsatisfiable then every interpretation must make it **f**. The only way for φ to be both satisfiable and unsatisfiable is for some interpretation to make φ both **t** and **f**. But this interpretation would then break the Law of Non-Contradiction, which we showed was obeyed by all interpretations. It is thus impossible for φ to be both satisfiable and unsatisfiable.

Exercise 3.5. Show that a formula cannot be both falsifiable and valid.

Exercise 3.6. Show that a formula cannot be both unsatisfiable and valid.

Logical Consequence: We now move on to the crucial semantic notion of logical consequence. We use the capital Greek letter Γ, pronounced "Gamma", as a name for an arbitrary collection of formulae. We say that an interpretation is a **model** for a formula φ if the interpretation gives the truth value **t** to φ. We say that an interpretation is a **model** for a set of formulae Γ if the interpretation gives the truth value **t** to every formula in Γ.[2] We say that an arbitrary formula φ is a **logical consequence** of Γ if every model for Γ is also a model for φ. That is, if every interpretation that gives a truth value **t** to every formula in Γ also gives a truth value **t** to φ.

Note that a formula is valid exactly when it is a logical consequence of the empty set since every interpretation is a model for the empty set.

Exercise 3.7. Show that the formula $p_2 \wedge p_1$ is a logical consequence of the set $\{p_2 \wedge p_1\}$.

[2] [See Chapter 44 for more about model theory.—Ed.]

Example 3.7. Let Γ be the set of formulae $\{(\neg p_0) \vee p_1, (\neg p_1) \vee p_2\}$ and let φ be $p_0 \to p_2$. The table below shows the $2^3 = 8$ possible different interpretations we can construct for the atomic formulae p_0, p_1 and p_2 which appear in Γ and φ. For each case, the table shows the truth values of the larger formulae that appear in Γ and the truth value of φ. The interpretations marked with $\bullet$ are all models for Γ since these interpretations assign **t** to every member of Γ: if any of these marked interpretations assign **f** to φ, then φ is not a logical consequence of Γ. Since all these marked interpretations assign **t** to $p_0 \to p_2$, the formula $p_0 \to p_2$ is a logical consequence of the set $\{(\neg p_0) \vee p_1, (\neg p_1) \vee p_2\}$.

Atomic			Γ		φ	
p_0	p_1	p_2	$(\neg p_0) \vee p_1$	$(\neg p_1) \vee p_2$	$p_0 \to p_2$	
t	**t**	**t**	**t**	**t**	**t**	$\bullet$
t	**t**	**f**	**t**	**f**	**f**	
t	**f**	**t**	**f**	**t**	**t**	
t	**f**	**f**	**f**	**t**	**f**	
f	**t**	**t**	**t**	**t**	**t**	$\bullet$
f	**t**	**f**	**t**	**f**	**t**	
f	**f**	**t**	**t**	**t**	**t**	$\bullet$
f	**f**	**f**	**t**	**t**	**t**	$\bullet$

Example 3.8. Let Γ_1 be the set of formulae $\{(\neg p_0) \vee p_1, (\neg p_1) \vee p_2, \neg(p_0 \to p_2)\}$ and let φ be an arbitrary formula. The set Γ_1 from this example contains the set Γ from Example 3.7 and contains the extra formula $\neg(p_0 \to p_2)$. Suppose Γ_1 has a model. This model must assign **t** to $\neg(p_0 \to p_2)$ since it is a member of Γ_1. This model for Γ_1 must also be a model for Γ since Γ is contained in Γ_1. Since our previous example told us that $p_0 \to p_2$ was a logical consequence of Γ, any interpretation that is a model for Γ_1, must be a model for $p_0 \to p_2$, and hence must assign **f** to $\neg(p_0 \to p_2)$. This model therefore assigns both **t** and **f** to $\neg(p_0 \to p_2)$. The Law of Non-Contradiction tells us that it is impossible for an interpretation to assign both **t** and **f** to any formula. Such an interpretation cannot exist: thus no interpretation can be a model for Γ_1.

How are we to evaluate "every model for Γ_1 is also a model for φ" when Γ_1 has no models? In classical propositional logic, we deem the quoted expression to be true in this case. That is, any formula φ whatsoever is a logical consequence of a set Γ_1 that has no models.

The argument in Example 3.8 illustrates a method of reasoning called *reductio ad absurdum* in Latin, which roughly translates to "reduction to absurdity". We assume that some given statement is true and show that this assumption leads to a clear contradiction. The given statement therefore cannot be true. By the Law of the Exclude Middle, it must therefore be false. Such a proof is commonly knows as a **proof by contradiction**. We shall use such reasoning repeatedly in the rest of this chapter.

We now illustrate some general principles of reasoning which are valid in classical propositional logic. We deliberately use formula shapes in terms of φ and ψ to show that they apply for all formulae with these shapes.

Exercise 3.8. Show that the following formulae (shapes) are valid:

1. $(\varphi \wedge \psi) \leftrightarrow (\psi \wedge \varphi)$ illustrates the *commutativity* of conjunction since the order of the components of the conjunction is immaterial.

2. $(\varphi \vee \psi) \leftrightarrow (\psi \vee \varphi)$ illustrates the *commutativity* of disjunction since the order of the components of the disjunction is immaterial.

3. $(\varphi \leftrightarrow \psi) \leftrightarrow (\psi \leftrightarrow \varphi)$ illustrates the *commutativity* of logical equivalence since the order of the components of the inner bi-conditionals is immaterial.

4. $(\varphi_1 \wedge (\varphi_2 \wedge \varphi_3)) \leftrightarrow ((\varphi_1 \wedge \varphi_2) \wedge \varphi_3)$ illustrates the *associativity* of conjunction since the order of the parentheses between the components of the conjunctions is immaterial.

5. $(\varphi_1 \vee (\varphi_2 \vee \varphi_3)) \leftrightarrow ((\varphi_1 \vee \varphi_2) \vee \varphi_3)$ illustrates the *associativity* of disjunction since the order of the parentheses between the components of the disjunctions is immaterial.

6. $(\varphi_1 \leftrightarrow (\varphi_2 \leftrightarrow \varphi_3)) \leftrightarrow ((\varphi_1 \leftrightarrow \varphi_2) \leftrightarrow \varphi_3)$ illustrates the *associativity* of logical equivalence since the order of the parentheses between the components of the inner bi-conditionals is immaterial.

7. $(\varphi \rightarrow \psi) \leftrightarrow ((\neg\psi) \rightarrow (\neg\varphi))$ is called the principle of *contraposition*: "if φ then ψ" is logically equivalent to "if $\neg\psi$ then $\neg\varphi$".

8. $(\neg\neg\varphi) \leftrightarrow \varphi$ is called the principle of *double negation*: two adjacent negation signs cancel out.

9. $(\neg(\varphi \wedge \psi)) \leftrightarrow ((\neg\varphi) \vee (\neg\psi))$ is called De Morgan's Law I and demonstrates how negation can be pushed through conjunction.

10. $(\neg(\varphi \vee \psi)) \leftrightarrow ((\neg\varphi) \wedge (\neg\psi))$ is called De Morgan's Law II and demonstrates how negation can be pushed through disjunction.

11. $(\neg(\varphi \rightarrow \psi)) \leftrightarrow (\varphi \wedge \neg\psi)$ illustrates how negation can be pushed through implication.

12. $(\varphi \rightarrow \psi) \leftrightarrow (\neg(\varphi \wedge \neg\psi))$ illustrates how a conditional is logically equivalent to the negation of a conjunction.

13. $(\varphi \wedge (\varphi \rightarrow \psi)) \rightarrow \psi$ illustrates the principle that if φ is true, and φ implies ψ is true, then ψ is true.

14. $(\varphi \leftrightarrow \psi) \leftrightarrow ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$ illustrates that an "_ if and only if _" statement can be written as the conjunction of two "if _ then _" statements.

Example 3.9. Suppose we read φ as "statement one is true" and read ψ as "statement two is true". Then we can read $\varphi \rightarrow \psi$ as "if statement one is true then statement two is true". To say that $\varphi \rightarrow \psi$ is true is to say that "if statement one is true then statement two is true is true". In English, the principle from Exercise 3.8(13) is precisely the statement "If statement one is true and if statement one is true then statement two is true is true then statement two is true." which we mentioned in the introduction.

Exercise 3.9. Show that:

1. If the formula $\neg\varphi$ is unsatisfiable then the formula φ is valid. Assume that $\neg\varphi$ is unsatisfiable and show that φ must then be valid.

2. If the formula φ is valid then the formula $\neg\varphi$ is unsatisfiable. Assume that φ is valid and show that $\neg\varphi$ must then be unsatisfiable.

Exercise 3.10. The negation of the valid formula $(\neg p_0) \vee p_0$ from Example 3.5 is $\neg((\neg p_0) \vee p_0)$. Confirm that $\neg((\neg p_0) \vee p_0)$ is unsatisfiable by showing that it takes the value **f** under all interpretations.

Exercise 3.9 tells us that a formula $\neg\varphi$ is unsatisfiable if and only if φ is valid. It also illustrates that to prove that a bi-conditional statement is true, it suffices to prove that the two conditional statements to which it is logically equivalent by Exercise 3.8(14) are individually true. We shall use this idea over and over again in the rest of this chapter.

We next prove a fundamental property about the notion of logical consequence in classical propositional logic. To indicate that the property is fundamental, we again call it a theorem, and prove it true by using valid principles and properties of classical propositional logic itself.

Theorem 3.2. A formula ψ is a logical consequence of the set $\Gamma \cup \{\varphi\}$ if and only if $\varphi \rightarrow \psi$ is a logical consequence of the set Γ.

Proof: The theorem is stated as an "_ if and only if _" statement and hence is a bi-conditional where the left component is "a formula ψ is a logical consequence of the set $\Gamma \cup \{\varphi\}$" and the right component is "$\varphi \rightarrow \psi$ is a logical consequence of the set Γ". By the valid principle from Exercise 3.8(14) we know that a bi-conditional is logically equivalent to a conjunction of two "if _ then _" statements. The theorem can therefore be stated as the conjunction of the statements shown below:

(1) If a formula ψ is a logical consequence of the set $\Gamma \cup \{\varphi\}$ then $\varphi \rightarrow \psi$ is a logical consequence of the set Γ.

(2) If $\varphi \to \psi$ is a logical consequence of the set Γ then ψ is a logical consequence of the set $\Gamma \cup \{\varphi\}$.

To prove the theorem true, it suffices to prove each of these conditional statements true individually.

Consider statement (1): it is an "if _ then _" statement and we want to show that it is true. Table 1(b) from Figure 1 tells us that there are three ways to make a conditional statement true but only one way to make it false. It is therefore more efficient to use a proof by contradiction to show that the conditional *cannot* be false, and to use the Law of the Excluded Middle to then conclude that the conditional is true.

Table 1(b) from Figure 1 tells us that the way to make an "if _ then _" statement false is to make the "if" part true and the "then" part false. In the proof of (1) below we assume that the "if" part of (1) is true, and that the "then" part of (1) false: that is we assume that the whole "if _ then _" statement (1) is false. We then show that these assumptions lead to a contradiction, thereby proving that our assumption is flawed: the "if _ then _" statement (1) *cannot* be false. But the Law of the Excluded Middle tells us that each statement is either true or it is false. Since the "if _ then _" statement cannot be false, it must be true.

The proof of (2) is similar except that the "if" part and the "then" part are interchanged.

Proof of (1): Suppose ψ is a logical consequence of the set $\Gamma \cup \{\varphi\}$ and suppose that $\varphi \to \psi$ is *not* a logical consequence of the set Γ. The latter means that there is some interpretation which assigns **t** to each member of Γ, but which assigns **f** to $\varphi \to \psi$. This interpretation must assign **t** to φ and assign **f** to ψ since this is the only way to assign **f** to $\varphi \to \psi$. This interpretation is therefore a model for the set $\Gamma \cup \{\varphi\}$ since it assigns **t** to each member of $\Gamma \cup \{\varphi\}$. Our first assumption was that ψ is a logical consequence of the set $\Gamma \cup \{\varphi\}$. That is, every model for $\Gamma \cup \{\varphi\}$ must assign **t** to ψ. Our particular interpretation is a model for $\Gamma \cup \{\varphi\}$ so it must assign **t** to ψ. But it already assigns **f** to ψ, so it must assign both **t** and **f** to ψ. Theorem 3.1 tells us that this is impossible. Hence it is impossible for ψ to be a logical consequence of the set $\Gamma \cup \{\varphi\}$ and $\varphi \to \psi$ *not* to be a logical consequence of the set Γ. That is, if ψ is a logical consequence of the set $\Gamma \cup \{\varphi\}$ then $\varphi \to \psi$ is a logical consequence of the set Γ.

Proof of (2): Suppose $\varphi \to \psi$ is a logical consequence of the set Γ and suppose that ψ is *not* a logical consequence of the set $\Gamma \cup \{\varphi\}$. Show that these two assumptions lead to a contradiction by using reasoning similar to the proof of (1).

Q.E.D.

Intuition: Theorem 3.2 shows that the conditional $\rightarrow$ captures logical consequence between single formulae: that is, ψ is a *logical consequence* of the set $\{\varphi\}$ if and only if the formula $\varphi \rightarrow \psi$ is *valid*. Notice that it is not sufficient for the conditional $\varphi \rightarrow \psi$ to be merely satisfiable (**t** under *some* interpretation), it must be valid (**t** under *all* interpretations).

We now have a syntax and a semantics for classical propositional logic. In the next section we shall see a calculus for classical logic.

4 The Sequent Calculus SK

We now move onto the final part of classical propositional logic, a calculus. Recall that a calculus must be purely syntactic, hence the semantics must play no role in it. Calculi therefore usually consist of a collection of rules for manipulating expressions built out of the syntax (of formulae), possibly using some extra symbols. Different extra symbols, and different ways of building expressions from formulae give different styles of calculi. Of course, each calculus must rigorously define when there is a deduction of the formula φ from a given collection of formulae Γ.

There are three main types of syntactic calculi: Hilbert-style calculi, natural deduction style calculi and sequent style calculi. Each use different extra symbols to build expressions from formulae, and each define deductions in different ways.

Hilbert-style calculi are the most traditional since their origins go back to Aristotle and other ancient Greek philosophers. They are named after the German mathematician David Hilbert although Gottlob Frege did much of the work in formalising them. Hilbert-style calculi are good for reasoning *about* deductions, but are not very good for *finding* deductions, so we do not go into further details here.

Natural deduction style calculi were invented independently by the Polish mathematician Stanisław Jászkowski [Ják34] and the German mathematician Gerhard Gentzen [Gen35]. Most authors choose natural deduction calculi for teaching purposes since deductions in them follow the usual structure of mathematical arguments, whence the name "natural deduction".

Sequent calculi were invented by Gerhard Gentzen [Gen35]. We concentrate on sequent style calculi because they provide a basis for an algorithm for automatically computing whether or not φ is logical consequence of Γ.

Sequent calculi use the following extra symbols:

$$\Longrightarrow \qquad , \qquad \frac{\qquad}{\qquad} \qquad \Gamma \quad \Delta$$

respectively called the sequent arrow, the comma, the horizontal line, and the capital Greek letters pronounced "Gamma" and "Delta". Once again, these symbols are **not** part of the syntax of classical propositional logic but are ancillary symbols which we use to define the sequent calculus. Thus

$$\text{(Id)}\quad \Gamma, \varphi \Longrightarrow \varphi, \Delta$$

$$(\neg l)\quad \frac{\Gamma \Longrightarrow \varphi, \Delta}{\Gamma, \neg\varphi \Longrightarrow \Delta} \qquad (\neg r)\quad \frac{\Gamma, \varphi \Longrightarrow \Delta}{\Gamma \Longrightarrow \neg\varphi, \Delta}$$

$$(\wedge l)\quad \frac{\Gamma, \varphi, \psi \Longrightarrow \Delta}{\Gamma, \varphi \wedge \psi \Longrightarrow \Delta} \qquad (\wedge r)\quad \frac{\Gamma \Longrightarrow \varphi, \Delta \quad \Gamma \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \varphi \wedge \psi, \Delta}$$

$$(\vee l)\quad \frac{\Gamma, \varphi \Longrightarrow \Delta \quad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \varphi \vee \psi \Longrightarrow \Delta} \qquad (\vee r)\quad \frac{\Gamma \Longrightarrow \varphi, \psi, \Delta}{\Gamma \Longrightarrow \varphi \vee \psi, \Delta}$$

$$(\rightarrow l)\quad \frac{\Gamma \Longrightarrow \varphi, \Delta \quad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \varphi \rightarrow \psi \Longrightarrow \Delta} \qquad (\rightarrow r)\quad \frac{\Gamma, \varphi \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \varphi \rightarrow \psi, \Delta}$$

$$(\leftrightarrow l)\quad \frac{\Gamma \Longrightarrow \varphi, \psi, \Delta \quad \Gamma, \varphi, \psi \Longrightarrow \Delta}{\Gamma, \varphi \leftrightarrow \psi \Longrightarrow \Delta}$$

$$(\leftrightarrow r)\quad \frac{\Gamma, \varphi \Longrightarrow \psi, \Delta \quad \Gamma, \psi \Longrightarrow \varphi, \Delta}{\Gamma \Longrightarrow \varphi \leftrightarrow \psi, \Delta}$$

Figure 2: Sequent Calculus SK for Classical Propositional Logic.

expressions like $\Gamma \rightarrow \Delta$ and $\Gamma \vee \Delta$ and $\neg\Gamma$ are neither formulae nor formulae shapes. Using these symbols we now define the following important notions.

Sequent: A sequent is an expression of the form $\Gamma \Longrightarrow \Delta$ where Γ and Δ are **finite, possibly empty, sets** of formulae. We use Γ and Δ as set names and write the set $\{p_0, p_1, p_2\}$ as just p_0, p_1, p_2 by omitting the braces. Since the order of the elements in a set is immaterial, this set can also be written as p_1, p_0, p_2 or as any of the other five permutations we can obtain from these three formulae. If Γ is a set of formulae and φ is a formula, then Γ, φ and φ, Γ both stand for the union of the sets Γ and $\{\varphi\}$.

Antecedent: The part to the left of the sequent arrow is the antecedent.

Succedent: The part to the right of the sequent arrow is the succedent.

Example 4.1. All of the expressions shown below are sequents:
$\neg\neg p_0 \Longrightarrow p_0 \qquad p_0, p_1, p_3 \Longrightarrow p_4, p_1, p_5 \qquad p_0, (p_0 \rightarrow p_1) \Longrightarrow p_1.$

Moreover, if Γ is the (name of the) set $\{p_0, p_1\}$ and Δ is the (name of the) set $\{p_4\}$ and φ is the (name of the) formula p_4 then $\Gamma, \varphi \Longrightarrow \varphi, \Delta$ is the sequent $p_0, p_1, p_4 \Longrightarrow p_4$ instead of the sequent $p_0, p_1, p_4 \Longrightarrow p_4, p_4$ since the expression φ, Δ stands for the union of the two sets $\{\varphi\}$ and Δ, and the union of $\{p_4\}$ and $\{p_4\}$ is the set $\{p_4\}$. Multiple occurrences of a formula, all in the antecedent or all in the succedent, are thus automatically collapsed into one occurrence.

Intuition: Recall that our calculus is supposed to consist of pure symbol manipulations, so a semantic intuition for a sequent is not appropriate. There is, however, a purely syntactic intuition which can be given without recourse to the semantics. Every sequent of the form

$$\varphi_1, \varphi_2, \varphi_3, \cdots, \varphi_n \Longrightarrow \psi_1, \psi_2, \psi_3, \cdots, \psi_m$$

can be read as the formula

$$(\varphi_1 \wedge (\varphi_2 \wedge (\varphi_3 \wedge \cdots \wedge (\varphi_{n-1} \wedge \varphi_n) \cdots))) \rightarrow (\psi_1 \vee (\psi_2 \vee (\psi_3 \vee \cdots \vee (\varphi_{m-1} \vee \psi_m) \cdots)))$$

where $n \geq 0$ and $m \geq 0$ are natural numbers. Under this formula reading of a sequent, Gentzen's comma should be read as an $\wedge$ connective when it appears in the antecedent, but should be read as an $\vee$ connective when it appears in the succedent. Unlike the reading of formulae, the reading of the comma *changes* from antecedent to succedent. The constants $\top$ and $\bot$ mentioned in the last paragraph of Section 2 are needed for this reading where an empty left hand side of a sequent is read as the formula $\top$ and an empty right hand side of a sequent is read as the formula $\bot$. Since we do not make use of this reading, we do not need these constants.

Sequent Rule: A sequent rule is an expression consisting of 0, 1 or 2 sequents, called the **premisses**, above a horizontal line, and a single sequent, called the **conclusion**, below the horizontal line. The name of the sequent rule is attached to the left of the horizontal line in parentheses. If the rule has zero premisses then the horizontal line is omitted. See Figure 2 for a listing of the rules that we shall use.

I have deliberately not given any intuitions about how to read sequent rules since they are supposed to constitute pure symbol manipulations: some formulae get joined together and moved about in going from the premisses to the conclusion while the formulae in Γ and Δ stay constant.

Principal and Side Formulae: A formula that is shown explicitly in the premisses is called a **sideformula** of that rule and a formula that is shown explicitly in the conclusion is called the **principal** formula of that rule. We say that a rule **introduces** its principal formula from its side formula(e).

Example 4.2. In Figure 2, the sequent rule with name $(\rightarrow l)$ has two premisses: a left premiss $\Gamma \Longrightarrow \varphi, \Delta$ and a right premiss $\Gamma, \psi \Longrightarrow \Delta$. The conclusion of this rule is the sequent $\Gamma, \varphi \rightarrow \psi \Longrightarrow \Delta$. The side formulae of the rule $(\rightarrow l)$ are φ and ψ while its principal formula is $\varphi \rightarrow \psi$.

Sequent Calculus SK: The sequent calculus SK is the collection of sequent rules shown in Figure 2. Many authors use the name LK for this calculus, since it is a simplified version of Gentzen's original sequent calculus, which he called LK. For every connective, there are two rules in SK: one with a name containing l (for "left") that introduces a principal formula with that connective into the antecedent of the conclusion, and another with a name containing r (for "right") that introduces a principal formula with that connective into the succedent of the conclusion. The rule (Id) is the only rule with no premisses, and it introduces an arbitrary formula into both its antecedent and succedent *simultaneously*.

The rules in Figure 2 are really built from sequent shapes, rather than actual sequents. In the rule (Id), for example, the symbols Γ and Δ are names for arbitrary unknown sets of formulae, and φ is the name of some arbitrary unknown formula which must occur in both the antecedent and the succedent of the sequent.

Exercise 4.1. Work out the formula corresponding to the rule (Id) under the formula reading mentioned above. You might care to work out the formulae corresponding to the premisses and conclusion of each rule, but at this stage, it is unlikely to shed any light on how the rules capture deduction.

Sequent Instances: We form an **instance** of a sequent rule (shape) by uniformly replacing the set names like Γ and Δ that appear in it by sets of formulae, and uniformly replacing formulae names like φ and ψ that appear in it by particular formulae. By "uniformly" we mean that all occurrences of the name φ (say) must be replaced by the same formula. We cannot replace one occurrence of φ with one formula, and another occurrence of φ with a different formula. Although the instances of Γ and Δ may be empty sets of formulae, the instances of side-formulae like φ and ψ must be present.

Example 4.3. Below are two instances of the (Id) rule $\Gamma, \varphi \Longrightarrow \varphi, \Delta$:

$$p_5, p_6, p_7 \wedge p_8 \Longrightarrow p_{17}, p_7 \wedge p_8 \qquad\qquad p_1 \wedge (p_1 \rightarrow p_2) \Longrightarrow p_1 \wedge (p_1 \rightarrow p_2)$$

Γ is $\{p_5, p_6\}$ $\quad$ Δ is $\{p_{17}\}$ $\quad$ φ is $p_7 \wedge p_8$

Γ is empty $\quad$ Δ is empty $\quad$ φ is $p_1 \wedge (p_1 \rightarrow p_2)$

In the left hand sequent, the "uniformly" restriction forbids us from instantiating the occurrence of φ in the antecedent of (Id) with p_6 or p_5 and instantiating the occurrence of φ in the succedent of (Id) with p_{17}.

Example 4.4. Neither of the sequents below is an instance of (Id):

$$p_5, p_6, p_8 \wedge p_7 \Longrightarrow p_{17}, p_7 \wedge p_8 \qquad\qquad p_1, (p_1 \to p_2) \Longrightarrow p_1 \wedge (p_1 \to p_2)$$

since neither sequent contains a formula which is common to its antecedent and succedent. In the left hand sequent, the formulae $p_8 \wedge p_7$ and $p_7 \wedge p_8$ are *different*. In the right hand sequent, the formula $p_1 \to p_2$ appears in the antecedent as a whole formula, but $p_1 \to p_2$ appears in the succedent as a *strict subformula* of $p_1 \wedge (p_1 \to p_2)$.

Example 4.5. Below are two instances of the rule $(\to l)$ from Figure 2:

$$\frac{\Longrightarrow p_1 \quad p_2 \Longrightarrow}{p_1 \to p_2 \Longrightarrow} \qquad \frac{p_1 \Longrightarrow p_4 \to p_5, p_2, p_3 \vee p_4 \quad p_1, p_2 \Longrightarrow p_2, p_3 \vee p_4}{p_1, (p_4 \to p_5) \to p_2 \Longrightarrow p_2, p_3 \vee p_4}$$

Γ and Δ are empty
φ is p_1 and ψ is p_2

Γ is $\{p_1\}$ and Δ is $\{p_2, p_3 \vee p_4\}$
φ is $p_4 \to p_5$ and ψ is p_2.

As the example on the right shows, it is often easier to read the rules backwards (upwards), from the conclusion to the premisses.

Derivation: A derivation is a tree of sequents where every leaf sequent of the tree is an instance of (Id), and where the child sequents in the tree are obtained from their parent sequent by instantiating a rule from SK.

The bottom-most sequent in a derivation is called the **end-sequent** and the derivation is said to be a derivation of that sequent.

Example 4.6. This example illustrates that each rule instantiation is independent of the others. In the instance of the $(\wedge l)$ rule, Γ is $\{p_2\}$ and Δ is $\{p_1\}$ and φ is p_0 and ψ is $\neg\neg p_1$. In the instance of the $(\neg l)$ rule, Γ is $\{p_2, p_0\}$ and Δ is $\{p_1\}$ and φ is $\neg p_1$. In the instance of the $(\neg r)$ rule, Γ is $\{p_2, p_0\}$ and Δ is $\{p_1\}$ and φ is p_1. Thus the instantiations are not constant throughout a derivation, but must be constant throughout one rule instance.

$$\dfrac{\dfrac{\dfrac{p_2, p_0, p_1 \Longrightarrow p_1}{p_2, p_0 \Longrightarrow \neg p_1, p_1}\,(\neg r)}{p_2, p_0, \neg\neg p_1 \Longrightarrow p_1}\,(\neg l)}{p_2, p_0 \wedge \neg\neg p_1 \Longrightarrow p_1}\,(\wedge l)$$

Notice that a derivation (tree) is built from smaller derivations (trees). The leaf sequents are instances of (Id) and are derivations of themselves. Given a derivation of some end-sequent, applying a single premiss rule *downward* transforms the derivation into a derivation of the conclusion of the rule. Given derivations of two end-sequents, applying a two premiss rule *downward* transforms the *two* derivations into a *single* derivation of the conclusion of the rule. We use this insight to define the notion of a deduction in SK.

Deduction in SK: A derivation *is* a **deduction** of its end-sequent from the collection of its leaf sequents. Each rule of SK, when read downwards, shows how to turn the deductions of its premisses (from their leaf sequents) into a deduction of its conclusion (from these same leaf sequents). The leaves of a derivation are deductions of themselves.

We still have to convince ourselves that instances of (Id) are acceptable starting points for deductions but we leave this issue till later since it is best done by using semantic notions.

Deduction of φ from Γ: Reading the sequent arrow $\Longrightarrow$ in a slightly different way, we say that there is a deduction of a formula φ from the set of formulae Γ exactly when the sequent $\Gamma \Longrightarrow \varphi$ is derivable in SK. This is a purely syntactic definition that involves no semantic notions.

Example 4.7. The derivation at right has leaf sequents $p_1, p_2 \Longrightarrow p_2$ and $p_1, p_2 \Longrightarrow p_1$ and each is an instance of (Id) since the first contains the formula p_2 in its antecedent and succedent while the second contains the formula p_1 in its antecedent and succedent. The top-most rule instance is of the rule $(\wedge r)$ and introduces the formula $p_2 \wedge p_1$ into the succedent of its conclusion sequent $p_1, p_2 \Longrightarrow p_2 \wedge p_1$. The bottom-most rule instance is of the rule $(\wedge l)$ with a premiss sequent $p_1, p_2 \Longrightarrow p_2 \wedge p_1$ and a conclusion sequent $p_1 \wedge p_2 \Longrightarrow p_2 \wedge p_1$.

$$\dfrac{\dfrac{p_1, p_2 \Longrightarrow p_2 \quad p_1, p_2 \Longrightarrow p_1}{p_1, p_2 \Longrightarrow p_2 \wedge p_1}\,(\wedge r)}{p_1 \wedge p_2 \Longrightarrow p_2 \wedge p_1}\,(\wedge l)$$

Example 4.8. The derivation at right is also of the same end-sequent as Example 4.7, but now we consider the rules bottom-up. The end-sequent $p_1 \wedge p_2 \Longrightarrow p_2 \wedge p_1$ is an instance of the conclusion $\Gamma \Longrightarrow \varphi \wedge \psi, \Delta$ of the rule $(\wedge r)$ where Γ is $\{p_1 \wedge p_2\}$ and Δ is empty and φ is p_2 and ψ is p_1. Notice that the "uniformly" restriction is not broken by this instantiation since we are allowed to instantiate $\varphi \wedge \psi$ in the succedent with $p_2 \wedge p_1$ while also instantiating Γ in the antecedent with $p_2 \wedge p_1$. The corresponding instances of the premisses $\Gamma \Longrightarrow \varphi, \Delta$ and $\Gamma \Longrightarrow \psi, \Delta$ of the $(\wedge r)$ rule are the sequents $p_1 \wedge p_2 \Longrightarrow p_2$ and $p_1 \wedge p_2 \Longrightarrow p_1$ respectively. The left sequent $p_1 \wedge p_2 \Longrightarrow p_2$ is an instance of the conclusion $\Gamma, \varphi \wedge \psi \Longrightarrow \Delta$ of the $(\wedge l)$ rule, where Γ is empty and Δ is $\{p_2\}$ and φ is p_1 and ψ is p_2, with corresponding premiss instance $p_1, p_2 \Longrightarrow p_2$. The right sequent $p_1 \wedge p_2 \Longrightarrow p_1$ is an instance of the conclusion $\Gamma, \varphi \wedge \psi \Longrightarrow \Delta$ of the $(\wedge l)$ rule, where Γ is empty and Δ is $\{p_1\}$ and φ is p_1 and ψ is p_2, with corresponding premiss instance $p_1, p_2 \Longrightarrow p_1$. Since these premisses are instances of (Id), we have another derivation of $p_1 \wedge p_2 \Longrightarrow p_2 \wedge p_1$.

$$\dfrac{\dfrac{p_1, p_2 \Longrightarrow p_2}{p_1 \wedge p_2 \Longrightarrow p_2}\,(\wedge l) \quad \dfrac{p_1, p_2 \Longrightarrow p_1}{p_1 \wedge p_2 \Longrightarrow p_1}\,(\wedge l)}{p_1 \wedge p_2 \Longrightarrow p_2 \wedge p_1}\,(\wedge r)$$

Examples 4.7 and 4.8 show that a sequent may have multiple derivations.

We now have a method for recognising derivations, but how might we find them? Suppose we were searching for a derivation of the end-sequent $p_1 \wedge p_2 \Longrightarrow p_2 \wedge p_1$. Since this sequent is not an instance of (Id), the only way to find a derivation for it is to find a rule of SK whose conclusion can be instantiated to $p_1 \wedge p_2 \Longrightarrow p_2 \wedge p_1$. This sequent, for example, cannot be an instance of the conclusion $\Gamma, \neg\varphi \Longrightarrow \Delta$ of the rule $(\neg l)$ since there is no negated formula in $p_1 \wedge p_2 \Longrightarrow p_2 \wedge p_1$. By such a process of elimination we can conclude that the only rules whose conclusions could be instantiated to give $p_1 \wedge p_2 \Longrightarrow p_2 \wedge p_1$ are the $(\wedge l)$ and $(\wedge r)$ rules. Using $(\wedge l)$ as the bottom-most rule instance gives the derivation in Example 4.7, while using $(\wedge r)$ as the bottom-most rule instance gives the derivation in Example 4.8. These examples indicate how to find a derivation for a given sequent.

Systematic Backward Search Procedure for Finding Derivations:

Initialisation: Write the given sequent at the bottom of a page so that it forms the sole leaf of a tree with only one node.

Choose Leaf: Choose a leaf sequent which is not an instance of (Id).

Choose Rule: Choose a rule of SK whose conclusion can be instantiated to the chosen leaf sequent.

Apply Rule Backwards: If both the above steps succeed then

1. Draw a horizontal line over the chosen leaf so it is no longer a leaf.
2. Instantiate the chosen rule's premiss(es) and write them above the horizontal line to obtain up to two new leaf sequents.
3. Write the name of the chosen rule to the right of the horizontal line.
4. Return to step Choose Leaf.

End: If every leaf sequent is an instance of (Id) then report success.

Different choices of rules will give different search trees and potentially different derivations. To conclusively report that a sequent has no derivation we must try *all* such search trees (although later on we will show that only one will actually suffice).

Exercise 4.2. Find derivations for the following sequents, and compare the formulae involved with those in Exercise 3.3 and Example 3.7:

1. $p_0 \to p_1 \Longrightarrow (\neg p_0) \vee p_1$
2. $(\neg p_0) \vee p_1 \Longrightarrow p_0 \to p_1$
3. $\Longrightarrow ((\neg p_0) \vee p_1) \leftrightarrow (p_0 \to p_1)$
4. $(\neg p_0) \vee p_1, (\neg p_1) \vee p_2 \Longrightarrow (\neg p_0) \vee p_2$.

Exercise 4.3. Try to find derivations for the following sequents:
$p_0 \to p_1 \Longrightarrow p_1 \to p_0$ $\quad p_1 \wedge p_2 \Longrightarrow (p_1 \wedge p_2) \wedge p_3$ $\quad p_1 \vee p_2 \Longrightarrow p_1$.
If you do find derivations then you have done something wrong as these sequents should not be derivable in SK.

Subformula Property for Rules: For every rule of SK, the sideformulae that appear in the premisses are strict subformulae of the principal formula in the conclusion.

Example 4.9. The side formula of the $(\neg l)$ rule is φ and it is a strict subformula of the principal formula $\neg\varphi$ of $(\neg l)$.

Intuition: If we read each rule of SK upwards, the constituents of its premisses already appear in its conclusion.

Subformula Property for Derivations: The formulae that appear in any sequent in any backward search tree of the root sequent $\Gamma \Longrightarrow \Delta$ are all subformulae of the formulae in the set $\Gamma \cup \Delta$ formed by taking the union of the sets Γ and Δ.

Given that the root sequent $\Gamma \Longrightarrow \Delta$ is built from finite sets, the set $\Gamma \cup \Delta$ must be finite. Since multiple occurrences of a formula are forbidden in the antecedents and succedents of sequents, the number of different sequents we can build from the set of formulae in $\Gamma \cup \Delta$ is finite. From a finite set of sequents, we can build only a finite number of different search trees or derivations. This means that any particular backward search tree for a given sequent $\Gamma \Longrightarrow \Delta$ must be finite, and even more importantly, that $\Gamma \Longrightarrow \Delta$ has a finite number of backward search trees. So backward search must end. Below we give a more direct argument for this fact.

Termination of backward search for a derivation: In any search tree, the number of logical connectives that appear in any one premiss is always one less than the number of logical connectives that appear in its conclusion since every backward rule application removes the connective of its principal formula. Every branch of a backward search tree for a finite root sequent will therefore eventually contain a (leaf) premiss sequent which is an instance of (Id) or which is not an instance of (Id) but which has no connectives in it at all. In the first case, the systematic procedure will never choose this leaf again. In the second case, the leaf will consist of atomic formulae only so no further rule of SK is backward applicable to it. In either case, backward search along this branch terminates. Since each backward rule application causes the current branch to split into at most two, and the root sequent has a *finite* antecedent and conclusion, there can be only a finite number of such branches, and the backward search procedure as a whole must eventually terminate and reach the step marked End.

We have now seen how to find and recognise derivations for a given sequent in the calculus SK and used this notion to define how to find and recognise a deduction of a formula φ from a set of formulae Γ.

5 Soundness and Completeness of SK

In this section we make connections between the syntactic notion of deduction and the semantic notion of logical consequence and show that they are two faces of the same coin. Since our notion of deduction is couched in terms of derivations in SK, it is easier to first make the connection between derivations in SK and logical consequence. We shall therefore show that SK captures logical consequence in the following two senses:

Soundness of SK: If the sequent $\Gamma \Longrightarrow \varphi$ is derivable in SK then φ is a logical consequence of the set of formulae Γ.

Completeness of SK: If φ is a logical consequence of the set of formulae Γ then the sequent $\Gamma \Longrightarrow \varphi$ is derivable in SK.

These facts together show that the purely syntactic notion of derivability in SK captures the purely semantic notion of logical consequence *precisely*.

To prove these facts we prove certain properties of the sequent calculus SK as stepping stones. To do so we use principles of classical propositional logic which we have already shown to be valid. We are therefore using classical propositional logic to reason about SK. Some of these proofs are quite complicated, so don't worry if you have trouble understanding them on a first reading.

We first extend some of the semantic notions from Section 3.2 from formulae to sequents. We say that a sequent is:

falsifiable: If some interpretation assigns $\mathbf{t}$ to every member of the antecedent and assigns $\mathbf{f}$ to *every* member of the succedent.

valid: If every interpretation which assigns $\mathbf{t}$ to every member of the antecedent assigns $\mathbf{t}$ to *some* member of the succedent.

Example 5.1.

1. The sequent $p_0, p_0 \wedge p_1 \Longrightarrow p_2$ is falsifiable under the interpretation $\mathbf{t}, \mathbf{t}, \mathbf{f}, \mathbf{f}, \mathbf{f}, \mathbf{f}, \cdots$.

2. The sequent $p_0, p_1 \Longrightarrow p_0 \wedge p_1$ is valid since any interpretation that assigns $\mathbf{t}$ to p_0 and to p_1, must assign $\mathbf{t}$ to $p_0 \wedge p_1$.

3. The sequent $p_0, \neg p_0 \Longrightarrow p_1$ is valid since no interpretation can assign $\mathbf{t}$ to every member of its antecedent. This sequent is an example where the conditional statement of validity is true vacuously because no interpretation makes the "if" part true.

4. The sequent $p_0, p_1 \Longrightarrow$ is falsifiable vacuously since the interpretation $\mathbf{t}, \mathbf{t}, \cdots$ assigns $\mathbf{t}$ to every member of the antecedent and assigns $\mathbf{f}$ to every member of the empty succedent.

5. The sequent $p_1, p_2 \Longrightarrow p_3, \neg p_3, p_4$ is valid trivially since every interpretation must assign $\mathbf{t}$ to either p_3 or to $\neg p_3$ by the Law of the Excluded Middle and Table 1(a).

Exercise 5.1. Show that if a sequent is not valid then it is falsifiable.

Exercise 5.2. Show that if a sequent is falsifiable then it is not valid.

Exercise 5.3. Together, Exercises 5.1 and 5.2 show that a sequent is falsifiable if and only if it is not valid. By using the principle of contraposition on each of the statements of Exercises 5.1 and 5.2, show that a sequent is valid if and only if it is not falsifiable. Thus the notions of falsifiable and valid are opposites.

5.1 Soundness of SK

When a result is not fundamental in itself, but is used at a later stage as a step in proving something fundamental, it is called a lemma. We now prove some lemmata which are eventually used to prove the soundness of SK.

Lemma 5.1. For every instance of every rule of SK except (Id): if the premisses are valid, then the conclusion is valid.

Proof: We must consider each rule in turn. We give the case for the $(\rightarrow l)$ rule and leave the others as exercises.

$(\rightarrow l)$: Consider any arbitrary instance of the rule $(\rightarrow l)$, and suppose that the instances of the premisses $\Gamma \Longrightarrow \varphi, \Delta$ and $\Gamma, \psi \Longrightarrow \Delta$ of this rule are each valid sequents. We must show that the corresponding instance of the conclusion $\Gamma, \varphi \rightarrow \psi \Longrightarrow \Delta$ is also valid.

If no interpretation assigns $\mathbf{t}$ to every member of the antecedent $\Gamma, \varphi \rightarrow \psi$ then we are done since the sequent $\Gamma, \varphi \rightarrow \psi \Longrightarrow \Delta$ is then valid vacuously as in Example 5.1(3).

Otherwise, consider any interpretation that assigns $\mathbf{t}$ to every member of the instance of the antecedent $\Gamma, \varphi \rightarrow \psi$. We must show that this interpretation assigns $\mathbf{t}$ to some member of the instance of the succedent Δ. There are now two cases depending upon how this interpretation assigns $\mathbf{t}$ to $\varphi \rightarrow \psi$:

φ is $\mathbf{f}$: One way for the interpretation to assign $\mathbf{t}$ to $\varphi \rightarrow \psi$ is by assigning $\mathbf{f}$ to φ. We can now use the assumption that the instance

of the premiss $\Gamma \Longrightarrow \varphi, \Delta$ is valid. That is, since this interpretation assigns $\mathbf{t}$ to each member of Γ, it must assign $\mathbf{t}$ to some member of φ, Δ. We know it cannot assign $\mathbf{t}$ to φ since φ is already assigned $\mathbf{f}$. Thus some member of Δ must be assigned $\mathbf{t}$ by this interpretation. This was precisely what we had to show.

φ is $\mathbf{t}$: The only other way for the interpretation to assign $\mathbf{t}$ to $\varphi \rightarrow \psi$ is by assigning $\mathbf{t}$ to both φ and ψ. We can now use the assumption that the instance of the premiss $\Gamma, \psi \Longrightarrow \Delta$ is valid. That is, since this interpretation assigns $\mathbf{t}$ to each member of Γ and assigns $\mathbf{t}$ to ψ, it must assign $\mathbf{t}$ to some member of Δ. This was precisely what we had to show.

Q.E.D.

Exercise 5.4. Complete this proof for each rule of SK except (Id).

Intuitions: The intuition behind Lemma 5.1 is that each rule of SK except (Id) preserves validity *downwards* from all of its premisses to its conclusion. The (Id) rule itself has no premisses but it has a much stronger semantic property as shown next.

Lemma 5.2. Every instance of (Id) is valid.

Proof: Suppose a sequent is an instance of (Id): hence it has the shape $\Gamma, \varphi \Longrightarrow \varphi, \Delta$. If no interpretation assigns $\mathbf{t}$ to every member of the antecedent Γ, φ then the sequent is vacuously valid. Otherwise, consider any interpretation that assigns $\mathbf{t}$ to each member of the antecedent Γ, φ. Clearly, this interpretation assigns $\mathbf{t}$ to some member of the succedent φ, Δ namely φ. The instance of $\Gamma, \varphi \Longrightarrow \varphi, \Delta$ then fulfils the definition of a valid sequent. **Q.E.D.**

Exercise 5.5. Go back to the definition of "Deduction in SK" just above Example 4.7 and convince yourself that it is acceptable to begin all deductions with instances of (Id).

Theorem 5.3. [Soundness] If a sequent is derivable then it is valid.

Proof: Suppose the sequent $\Gamma \Longrightarrow \Delta$ is derivable. We have to show that it must be valid. The derivation of the sequent $\Gamma \Longrightarrow \Delta$ could be totally arbitrary, so we consider its shape.

If the sequent $\Gamma \Longrightarrow \Delta$ is itself an instance of (Id) then the derivation consists of just this one sequent and we know it must be valid by Lemma 5.2.

Otherwise, the derivation is a tree of arbitrary shape and we begin at the leaves, which are all instances of (Id). Each of these leaf sequents (there

may be only one) is the premiss of a rule of SK and all of these leaf sequents are valid by Lemma 5.2. Lemma 5.1 then tells us that the conclusions of all these rule applications must be valid. Repeating this argument, we can use Lemma 5.1 to percolate the validity of the leaf sequents all the way down to the end-sequent $\Gamma \Longrightarrow \Delta$, which must then be valid. **Q.E.D.**

A fundamental result that follows easily from previous theorems or definitions is called a corollary.

Corollary 5.4. If a sequent is not valid then it is not derivable.

Proof: The statement of the corollary is logically equivalent to the statement of Theorem 5.3 by the valid principle of contraposition. **Q.E.D.**

Theorem 5.3 tells us that the calculus SK allows us to derive only valid sequents. Clearly, if the first backward search tree we attempt turns out to be a derivation, then we can declare the end-sequent to be derivable. By restricting the succedent to a single formula, we obtain the version of soundness we seek.

Theorem 5.5. [Soundness of SK] If the sequent $\Gamma \Longrightarrow \varphi$ is derivable then φ is a logical consequence of Γ.

Proof: If the sequent $\Gamma \Longrightarrow \varphi$ is derivable then the sequent $\Gamma \Longrightarrow \varphi$ is valid by Theorem 5.3. By the definition of valid sequent, every interpretation that assigns $\mathbf{t}$ to every member of Γ then assigns $\mathbf{t}$ to φ. By the definition of a model, this is the same as saying that every model for Γ is a model for φ. By the definition of logical consequence, the formula φ is then a logical consequence of Γ. **Q.E.D.**

5.2 Completeness of SK

We now turn to completeness and once again start with some lemmata which are used as stepping stones in the completeness proof.

Lemma 5.6. For every instance of every rule of SK except (Id): if at least one of the premisses is falsifiable, then the conclusion is falsifiable.

Proof: We must consider each rule in turn. We give half of the case for the $(\rightarrow l)$ rule and leave the others as exercises.

$(\rightarrow l)$**:** Consider an arbitrary instance of the $(\rightarrow l)$ rule and suppose that the instance of its right premiss $\Gamma, \psi \Longrightarrow \Delta$ is falsifiable. We must show that the corresponding instance of its conclusion $\Gamma, \varphi \rightarrow \psi \Longrightarrow \Delta$ is also falsifiable.

Since $\Gamma, \psi \Longrightarrow \Delta$ is falsifiable, there exists some interpretation that assigns $\mathbf{t}$ to each member of Γ and assigns $\mathbf{t}$ to ψ, but which assigns

f to each member of Δ. There are now two cases depending upon the assignment for φ under this interpretation:

φ is **f**: If the interpretation assigns **f** to φ then it assigns **t** to $\varphi \to \psi$.

φ is **t**: If the interpretation assigns **t** to φ then it assigns **t** to $\varphi \to \psi$ since it already assigns **t** to ψ.

In both cases, this interpretation assigns **t** to each member of the antecedent of $\Gamma, \varphi \to \psi \Longrightarrow \Delta$ and assigns **f** to each member of the succedent. This is precisely what we have to show to conclude that $\Gamma, \varphi \to \psi \Longrightarrow \Delta$ is falsifiable.

Q.E.D.

Exercise 5.6. Complete this proof for the case where the instance of the left premiss $\Gamma \Longrightarrow \varphi, \Delta$ of the rule $(\to l)$ is falsifiable.

Exercise 5.7. Complete this proof for all other rules of SK except (Id).

Intuition: The intuition of Lemma 5.6 is that each rule of SK except (Id) preserves falsifiability downwards from any one of its premisses to its conclusion. More importantly, Lemma 5.6 also tells us that the interpretation that falsifies the premiss, also falsifies the conclusion. That is, the rules of SK also preserve the falsifying interpretation itself downwards from any one falsifiable premiss to its conclusion.

Completed Sequent: We say that a sequent is **completed** if it cannot be the instance of the conclusion of any rule of SK except possibly (Id). That is, the only rule of SK whose conclusion might possibly be instantiated to this sequent is the (Id) rule.

Lemma 5.1 tells us that every instance of (Id) is valid, so no instance of (Id) is falsifiable. We can, however, say something about completed sequents which are *not* instances of (Id).

Lemma 5.7. Every completed sequent that is not an instance of (Id) is falsifiable.

Proof: Suppose we are given a completed sequent which is not an instance of (Id). If the antecedent of this sequent contained a (non-atomic) formula $\neg\varphi$, then we could make this sequent into an instance of the conclusion $\Gamma, \neg\varphi \Longrightarrow \Delta$ of the $(\neg l)$ rule. But then, our given sequent would not be completed, hence the antecedent of our sequent cannot contain a negated formula. This same argument can be repeated for every non-atomic formula shape $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \to \psi$, $\varphi \leftrightarrow \psi$, by replacing the rule $(\neg l)$ by the appropriate rule, hence the antecedent cannot contain any non-atomic formula. A

symmetric argument using the right version of each of the rules tells us that the succedent of our sequent also cannot contain any non-atomic formula. That is, the given sequent must consist of atomic formulae only.

Let the interpretation be as follows: from our given sequent, assign **t** to every atomic formula in the antecedent, and assign **f** to every atomic formula in the succedent. All other atomic formulae can be assigned **f**.

By definition, this interpretation falsifies the given sequent. The only complication would be if the antecedent and succedent both contained some common atomic formula p, since then our interpretation would assign both **t** and **f** to this atomic formula. But there can be no such common p since our given sequent is not an instance of (Id). **Q.E.D.**

Theorem 5.8. [Completeness] If a sequent is valid then it is derivable.

Proof: To prove this theorem we first use valid principles of classical logic to transform it into an equivalent form.

Step 1: The theorem is a statement of the form "if _ then _" and is therefore a conditional.

Step 2: We can use the valid principle of contraposition from Exercise 3.8(7) to restate the conditional as another conditional by negating each component and exchanging their places, since the latter is logically equivalent to the former. Thus the theorem can be equivalently restated as: If a sequent is not derivable then it is not valid.

Step 3: Exercise 5.3 told us that a sequent is falsifiable if and only if it is not valid. Thus we can equivalently restate the second version of the theorem as: If a sequent is not derivable then it is falsifiable.

To prove this third version, suppose the sequent $\Gamma \Longrightarrow \Delta$ is not derivable. This means that *every* attempt to find a derivation for $\Gamma \Longrightarrow \Delta$ using our systematic backward search procedure terminates with a finite tree of sequents where at least one leaf is completed and not an instance of (Id).

Consider any such failed backward search tree and choose one of its completed leaves which is not an instance of (Id). Lemma 5.7 tells us that this leaf sequent must be falsifiable.

If this leaf is the sequent $\Gamma \Longrightarrow \Delta$ because no rules were backwards applicable to $\Gamma \Longrightarrow \Delta$, then we are done.

Otherwise, this leaf must be the premiss of an instance of some rule of SK. Lemma 5.6 then tells us that the corresponding conclusion in the chosen tree must also be falsifiable, regardless of how many premisses this rule has.

By repeating this argument, we can use Lemma 5.6 to percolate the falsifiability of the original (falsifiable) leaf all the way down the branch to the sequent $\Gamma \Longrightarrow \Delta$ at the root of the chosen tree, which must then be falsifiable. **Q.E.D.**

Theorem 5.8 tells us that every valid sequent must have a derivation using the rules of SK.

Example 5.2. The sequent $p_0, \neg p_0 \Longrightarrow p_1$ is valid vacuously as shown in Example 5.1(3). Theorem 5.8 tells us that it must be derivable. The derivation at right can be found using our backward search procedure.

$$\frac{p_0 \Longrightarrow p_0, p_1}{p_0, \neg p_0 \Longrightarrow p_1}\,(\neg l)$$

We now make the promised connection between logical consequence and derivability by restricting the succedent to a single formula.

Theorem 5.9. [Completeness of SK] If φ is a logical consequence of Γ then the sequent $\Gamma \Longrightarrow \varphi$ is derivable.

Proof: If φ is a logical consequence of Γ then every model for Γ is a model for φ by the definition of model. Thus every interpretation that assigns $\mathbf{t}$ to every member of Γ assigns $\mathbf{t}$ to φ: that is, the sequent $\Gamma \Longrightarrow \varphi$ is valid. By Theorem 5.8, the sequent $\Gamma \Longrightarrow \varphi$ is then derivable. **Q.E.D.**

5.3 Deductions and Logical Consequence

We now make the promised connections between deduction and logical consequence.

Corollary 5.10. The formula φ is a logical consequence of Γ if and only if the sequent $\Gamma \Longrightarrow \varphi$ is derivable.

Proof: This is just the conjunction of Theorem 5.9 and Theorem 5.5 written as an "_ if and only if _" statement. **Q.E.D.**

Corollary 5.11. The formula φ is valid if and only if the sequent $\Longrightarrow \varphi$ is derivable.

Proof: The formula φ is valid exactly when it is a logical consequence of the empty set. So just make Γ empty in Corollary 5.10. **Q.E.D.**

Exercise 5.8. For each of the valid shapes of Exercise 3.8, find a derivation for the sequent with an empty antecedent and a succedent consisting wholly of that valid shape. The derivation of the shape $(\neg\neg\varphi) \leftrightarrow \varphi$ is shown at right. Strictly, each of these derivations is really a template since it is not a tree of sequents *per se* but a tree of sequent shapes built from formulae names like φ and ψ. Uniformly replacing

$$\dfrac{\dfrac{\dfrac{\varphi \Longrightarrow \varphi}{\Longrightarrow \neg\varphi, \varphi}\,(\neg r)}{\neg\neg\varphi \Longrightarrow \varphi}\,(\neg l) \qquad \dfrac{\dfrac{\varphi \Longrightarrow \varphi}{\neg\varphi, \varphi \Longrightarrow}\,(\neg l)}{\varphi \Longrightarrow \neg\neg\varphi}\,(\neg r)}{\Longrightarrow (\neg\neg\varphi) \leftrightarrow \varphi}\,(\leftrightarrow r)$$

the names like φ and ψ with particular formulae will give actual derivations. The distinction is the same as between a formula *per se* and a formula with name φ. By now, you should be comfortable with this abuse of notation.

Corollary 5.12. There is a deduction of the formula φ from a set of formulae Γ if and only if φ is a logical consequence of Γ.

Proof: There is a deduction of the formula φ from a set of formulae Γ if and only if there is a derivation of the sequent $\Gamma \Longrightarrow \varphi$ in SK by the definition of deduction, if and only if φ is a logical consequence of Γ by Corollary 5.10. **Q.E.D.**

5.4 Algorithmic Aspects of SK

We now move on to the algorithmic aspects of SK. We know that the systematic backward search procedure will terminate for every (finite) root sequent: either because it finds a derivation or because it finds a completed leaf sequent that is not an instance of (Id). But in this latter case, the systematic procedure will backtrack over previous rule choices and seek other rules to apply backwards in the hope of finding a derivation. Do we really have to search through all other possible search trees before deciding whether the given sequent is valid or falsifiable? We already have enough information to answer this question in the negative.

Suppose the root sequent in this latter case is $\Gamma \Longrightarrow \Delta$. The completed leaf which is not an instance of (Id) is falsifiable by the interpretation chosen by Lemma 5.7. The proof of (the Completeness) Theorem 5.8 tells us that the falsifiability of this leaf can be percolated down to the root sequent $\Gamma \Longrightarrow \Delta$. Since $\Gamma \Longrightarrow \Delta$ is falsifiable, it is not valid. Corollary 5.4 (of Soundness) tells us that $\Gamma \Longrightarrow \Delta$ is not derivable: that is, there is no point in trying other backward rule applications since there is *no* derivation for this sequent in SK. We can therefore conclude that the root sequent is not derivable as soon as we find the *first* completed leaf sequent which is not an instance of (Id) in the *first* backward search tree we try. The order in which we apply the rules of SK backwards is *irrelevant*. Moreover, by the intuitions of Lemma 5.6 that follow Exercise 5.7, we know that the interpretation that falsifies this first completed leaf, also falsifies the root sequent $\Gamma \Longrightarrow \Delta$.

Backward search in SK not only provides a *purely syntactic* **algorithm** for deciding whether φ is a logical consequence of Γ, but it also gives a concrete *semantic* interpretation that proves that φ is not a logical consequence of Γ when this is the case.

Example 5.3. The formula p_2 is not a logical consequence of the set

$\{p_0, p_0 \to p_1\}$ as shown by the failed search tree at right. The completed leaf $p_0, p_1 \Longrightarrow p_2$ is not an instance of (Id) and it can be falsified by the interpretation $\mathbf{t}, \mathbf{t}, \mathbf{f}, \mathbf{f}, \mathbf{f} \cdots$. This interpretation is a model for the set $\{p_0, p_0 \to p_1\}$ but makes p_2 false giving a concrete semantic interpretation proving why the formula p_2 is not a logical consequence $\{p_0, p_0 \to p_1\}$.

$$\frac{p_0 \Longrightarrow p_0, p_2 \quad p_0, p_1 \Longrightarrow p_2}{p_0, p_0 \to p_1 \Longrightarrow p_2}\,(\to l)$$

Exercise 5.9. For every sequent from Exercise 4.3, find a failed search tree and extract from it an interpretation that falsifies that sequent.

As we have seen, computing whether or not φ is a logical consequence of some given Γ using truth tables is a rather laborious task when Γ and φ consist of many different atomic formulae; see Exercise 3.4. The question of whether we can compute logical consequence automatically on a computer is therefore very important. The algorithm based on SK is usually, but not always, more efficient than truth tables [DM94].

6 Gentzen's Original Sequent Calculus LK

Gentzen's original sequent calculus was called LK and differed from our SK in the following ways.

Sequences instead of sets: The antecedents and succedents of sequents in LK were built from *sequences* (lists) of formulae rather than *sets* of formulae. Not only were the order of the formulae significant, but so were the number of occurrences of each formula in an antecedent and succedent. The following sequents are different in LK but collapse to either of the two left hand sequents in SK:

$$p_0, p_1 \Longrightarrow p_3 \qquad p_1, p_0 \Longrightarrow p_3 \qquad p_0, p_1 \Longrightarrow p_3, p_3.$$

Gentzen's LK therefore contained two extra rules, called the structural rules of exchange and contraction which allowed him to permute the order of formulae and reduce multiple consecutive occurrences of a formula into one single occurrence respectively. These rules have no analogues in SK since sequents in SK are built from sets, so exchange and contraction are already built into SK.

Different symbols: Gentzen used $\to$ as the sequent arrow and used $\supset$ as the conditional connective whereas we have used $\Longrightarrow$ and $\to$ respectively, so we must be careful when comparing LK and SK.

Different Id Rule: The (Id) rule in Gentzen's LK was of the form $\varphi \Longrightarrow \varphi$. Gentzen's LK therefore contained an extra structural rule of thinning to

allow the addition of arbitrary formulae to the antecedent and succedent of sequents. We could have formulated SK in this way but using our (Id) rule simplifies derivations and makes them easier to find using backward search.

Cut Rule: Gentzen's LK contained a rule whose analogue for SK is:

$$(\text{cut}) \quad \frac{\Gamma \Longrightarrow \varphi, \Delta \qquad \Gamma, \varphi \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta}$$

The cut rule allows us to compose the derivations of its left and right premisses (downward) into a derivation of its conclusion sequent $\Gamma \Longrightarrow \Delta$. It is the only rule we have seen which can break the subformula property: there is a formula φ in each of the premisses which may not be in the conclusion. The cut rule is backwards applicable to *any* sequent, and demands that we choose some arbitrary formula φ to add to the left and right premisses. Since there is no relationship between the formulae in $\Gamma \cup \Delta$ and φ, backward search cannot be applied when the cut rule is present in a sequent calculus.

SKCut: We use SKCut to mean the sequent calculus obtained by adding the cut rule to SK.

Exercise 6.1. Show that the cut rule is sound with respect to the semantics of classical propositional logic. That is, show that for every instance of the cut rule: if the premisses are valid then the conclusion is valid.

Gentzen proved a very famous theorem which he called the *Hauptsatz*, but which is usually called the cut-elimination theorem, showing that the cut rule was redundant in LK. In SKCut the theorem has the form below.

Theorem 6.1. [Cut-elimination for SKCut] If a sequent is derivable in SKCut then the same sequent is derivable in SK.

Since SKCut contains the cut rule, the derivation of the sequent in SKCut can contain applications of the cut rule. Since SK does not contain the cut rule, the derivation of this same sequent in SK cannot contain any applications of the cut rule: so the theorem states that the cut rule is redundant in SKCut.

Proof: Suppose a sequent is derivable in SKCut. Since the extra cut rule is sound by Exercise 6.1, (the Soundness) Theorem 5.3 also applies to SKCut, and tells us that the sequent is valid. But SK itself is sound and complete: a sequent is derivable in SK if and only if it is valid. That is, all valid sequents are already derivable in SK, including this sequent. **Q.E.D.**

This proof relies on the semantics of classical propositional logic since a valid sequent is defined in terms of interpretations. Gentzen's proof of this

theorem was purely syntactic: he showed how to transform any LK derivation containing applications of the cut rule into a cut-free LK derivation of the same sequent. Gentzen's proof is long and difficult, and is the single most important result in the study of sequent calculi.

In Section 4 we mentioned that a sequent could be read as a formula by changing the sequent arrow into $\rightarrow$, replacing all commas in the antecedent by $\wedge$ and replacing all commas in the succedent by $\vee$ (with appropriate parentheses). This *overloading* of the comma to mean different connectives when in the antecedent and succedent is at the heart of sequent calculi. The termination argument at the end of Section 4, for example, relies on the fact that commas do *not* count as connectives.

Although Gentzen explicitly mentions this formula reading of a sequent, it is not known if he deliberately meant his comma to be overloaded (ambiguous). For Gentzen explicitly lists the sequent arrow as an auxiliary (extra) symbol, but he does not list his comma as one. So it is possible that Gentzen was using the comma as a mere punctuation mark to separate the members of his sequents, which are formed from lists of formulae rather than sets. At the bottom of page 71 of Szabo's English translation [Gen35] we find the following parenthetical remark: *"(The $\rightarrow$, like commas, is an auxiliary symbol and not a logical symbol.)"*. Remember that Gentzen used $\rightarrow$ as his sequent arrow, so he is saying that the sequent arrow is auxiliary as defined, but so is the comma! It therefore appears as if Gentzen meant for his comma to be an explicit auxiliary symbol, and not just a punctuation mark, but that he *forgot* to list it in the list of auxiliary symbols.

The deliberate overloading of extra auxiliary symbols can be generalised to give powerful variants of sequent calculi called display calculi [Bel82, Gor98]. The structure of sequents can also be extended by compartmentalising the antecedents and succedents in various ways [Doš88, Doš89, Avr96].

7 Applications of Sequent Calculi

We now show that sequent calculi are useful for proving results *about* logics.

Lemma 7.1. [Invertibility] For every instance of every rule of SK except (Id): if the conclusion is derivable, then the premisses are derivable.

Proof: Consider any instance of any rule of SK except (Id). Suppose the conclusion is derivable and some premiss is not derivable. The conclusion must be valid by Theorem 5.3. By the statement of Theorem 5.8 in Step 2 of the proof of Theorem 5.8, this premiss that is not derivable is not valid, and then by Step 3, this premiss is falsifiable. By Lemma 5.6, the conclusion is then falsifiable. The conclusion is thus both valid and falsifiable. But no sequent can be both falsifiable and valid by Exercise 5.3. So it is impossible to have a derivable conclusion and an underivable premiss. **Q.E.D.**

Corollary 7.2. For every instance of every rule of SK except (Id): the conclusion is derivable if and only if the premisses are derivable.

Proof: If the premisses are derivable then the conclusion is derivable by the definition of a derivation. If the conclusion is derivable then the premisses are derivable by Lemma 7.1. **Q.E.D.**

Intuitions: The invertibility lemma and its corollary tell us that the rules of SK preserve derivability (and validity) both upwards and downwards, once again, confirming that the order of rule applications is irrelevant.

Suppose we are told that some particular sequent is derivable, but we are not shown its derivation. Suppose that there is also at least one non-atomic formula in the sequent. Corollary 7.2 tells us that we can pick *any* non-atomic formula in the sequent and pretend that the bottom-most rule application of the unknown derivation introduces that formula into the sequent. The corresponding premiss is guaranteed to be derivable.

Example 7.1. Example 4.7 tells us that $p_1 \wedge p_2 \Longrightarrow p_2 \wedge p_1$ has a derivation where the bottom-most rule application is of the $(\wedge l)$ rule. But suppose we had only been told the information that $p_1 \wedge p_2 \Longrightarrow p_2 \wedge p_1$ is derivable. This *derivable* sequent $p_1 \wedge p_2 \Longrightarrow p_2 \wedge p_1$ however can also be made an instance of the conclusion $\Gamma \Longrightarrow \varphi \wedge \psi, \Delta$ of the $(\wedge r)$ rule with corresponding premisses $p_1 \wedge p_2 \Longrightarrow p_2$ and $p_1 \wedge p_2 \Longrightarrow p_1$. By Lemma 7.1, these premisses *must* have derivations. Example 4.8 confirms that indeed they do.

A syntactic calculus is **consistent** if it is impossible for both φ and $\neg\varphi$ to have deductions from the empty set in that calculus, for any formula φ.

Theorem 7.3. [Consistency of SK] The sequent $\Longrightarrow \varphi$ and the sequent $\Longrightarrow \neg\varphi$ cannot both be derivable in SK, for any formula φ.

Proof: (Semantic) The definition of valid formula immediately tells us that the formulae φ and $\neg\varphi$ cannot both be valid. So we could use these facts together with Corollary 5.11 to give a semantic proof. **Q.E.D.**

Proof: (Syntactic) Instead we give a purely syntactic proof. Suppose the sequents $\Longrightarrow \varphi$ and $\Longrightarrow \neg\varphi$ are derivable in SK: hence there are (unknown) derivations in SK whose end-sequents are $\Longrightarrow \varphi$ and $\Longrightarrow \neg\varphi$ respectively.

The conclusion $\Gamma \Longrightarrow \neg\varphi, \Delta$ of the $(\neg r)$ rule can be instantiated to the sequent $\Longrightarrow \neg\varphi$ by taking Γ and Δ to be empty, giving a corresponding premiss instance $\varphi \Longrightarrow$. Since $\Longrightarrow \neg\varphi$ is derivable in SK, and is an instance of the conclusion of $(\neg r)$, Lemma 7.1 tells us that its corresponding premiss $\varphi \Longrightarrow$ also has some (unknown) derivation in SK.

Since every rule of SK is also a rule in SKCut, the derivations of $\Longrightarrow \varphi$ and $\varphi \Longrightarrow$ are also derivations in SKCut. Since SKCut contains the cut rule, we can extend these derivations by the following application of cut:

$$\frac{\Longrightarrow \varphi \qquad \varphi \Longrightarrow}{\Longrightarrow} \text{ (cut)}$$

giving a derivation of the empty sequent $\Longrightarrow$ in SKCut.

By (the cut-elimination) Theorem 6.1, the cut rule is redundant in SKCut, so there must be a (cut-free) derivation of the empty sequent in SK. But this is *impossible* since the empty sequent is not an instance of (Id), and no rule of SK is backwards applicable to it! Thus the sequents $\Longrightarrow \varphi$ and $\Longrightarrow \neg\varphi$ cannot both be derivable in SK, and SK is consistent. **Q.E.D.**

We now give a syntactic proof of the syntactic analogue of Theorem 3.2.

Theorem 7.4. [Deduction Theorem] There is a deduction of the formula ψ from the set $\Gamma \cup \{\varphi\}$ if and only if there is a deduction of the formula $\varphi \to \psi$ from the set Γ.

Proof: We split the theorem into two parts and prove each independently.

1. Suppose there is a deduction of ψ from the set $\Gamma \cup \{\varphi\}$: that is, there is a derivation of the end-sequent $\Gamma, \varphi \Longrightarrow \psi$. If we apply the $(\to r)$ rule downwards to this end-sequent we obtain a new conclusion end-sequent $\Gamma \Longrightarrow \varphi \to \psi$ and hence obtain a derivation of $\Gamma \Longrightarrow \varphi \to \psi$. This means that there is a deduction of $\varphi \to \psi$ from the set Γ.

2. Suppose there is a deduction of the formula $\varphi \to \psi$ from the set Γ. That is, there is a derivation of the end-sequent $\Gamma \Longrightarrow \varphi \to \psi$. Independently, we can make $\Gamma \Longrightarrow \varphi \to \psi$ the conclusion of an instance of the $(\to r)$ rule, with empty Δ, giving a premiss instance $\Gamma, \varphi \Longrightarrow \psi$. Since the conclusion $\Gamma \Longrightarrow \varphi \to \psi$ is known to be derivable, (the Invertibility) Lemma 7.1 tells us that the premiss $\Gamma, \varphi \Longrightarrow \psi$ of this instance of $(\to r)$ must also be derivable. This means that there is a deduction of ψ from the set $\Gamma \cup \{\varphi\}$.

Q.E.D.

8 History

The history of classical propositional logic is over 2000 years old, so I cannot possibly do it justice in such a short chapter.

Modern Western logic originates from the work of the Greek philosopher Aristotle (384-322 BC) who postulated the Law of Non-Contradiction and defined a notion which is almost identical to our modern definition of logical consequence: there are some differences like the fact that Aristotle would not have allowed φ to be a logical consequence of itself. Although Aristotle saw the need for two truth values "true" and "false", he also saw the problems that this created with truth values of statements about the future like "It

will rain tomorrow". Of course, Aristotle did much more than this, but the main distinctions into syntax, semantics and calculi are very recent.

Augustus De Morgan (1806-1871) invented the modern connectives we have used while George Boole (1815-1864) invented the truth tables we saw in Figure 1. The basic principle inherent in these truth tables, that the truth value of a formula is determined by the truth values of its immediate subformulae, was known from Greek times.

The German mathematician Gottlob Frege (1848-1925) is considered to be the founder of modern symbolic logic because he invented a calculus for classical (propositional and first-order) logic and formalised the notion of a deduction as a sequence of purely syntactic manipulations. The style of calculus that he used is now known as a Hilbert-calculus after the German mathematician David Hilbert (1862-1943), although the origins of such calculi go back to the ancient Greeks. Frege attempted to show that all of mathematics could be based upon symbolic logic, but Frege's original calculus was shown to be inconsistent by Bertrand Russell (1872-1970): Russell showed that there was a formula φ such that both φ and $\neg\varphi$ had deductions from the empty set in Frege's calculus. The incompleteness results of Kurt Gödel (1906-1978) eventually showed that it is impossible to base all of mathematics upon a single consistent logical calculus.[3]

Charles Sanders Peirce (1839-1914), an American, also made many important and independent discoveries, including the invention of a purely syntactic calculus for classical logic.

Emile Post (1897-1954) formally stated the theorems we have called soundness and completeness theorems, although he couched them in terms of a Hilbert-style calculus, since this was the only syntactic calculus style known at the time. All of that was changed by essentially one man.

In 1935, Gerhard Gentzen (1909-1945) gave both a natural deduction style calculus and a sequent style calculus for classical logic [Gen35]. Stanisław Jászkowski (1906-1965) had given a natural deduction calculus independently in 1934 [Ják34], but sequent calculi are more general than natural deduction calculi in many ways. Gentzen proved the consistency of arithmetic using his sequent calculus. Gentzen had withdrawn an earlier paper in 1932 because he feared that Gödel's incompleteness results contradicted his own results. But there is no contradiction: to prove that arithmetic was consistent, Gentzen used mathematical principles which are outside of arithmetic itself. Gödel's results show that it is *essential* to use such higher level principles.

After completing his doctorate in 1933, Gentzen worked as an assistant for Hilbert, and after obtaining a German certificate that allowed him to teach in 1942, Gentzen took up a job teaching logic and mathematics at the German University in Prague, the capital of Czechoslovakia (now the

[3] [See Chapter 48 concerning Gödel's results.—Ed.]

Czech Republic). A common misconception is that Gentzen died in a Nazi concentration camp, but the following extract about Gentzen from [OR01] clarifies this aspect:

> *As part of the German war effort, he took up a teaching post as a Dozent in the Mathematical Institute of the German University of Prague and he taught there until arrested and taken into custody. The citizens of Prague rose in revolt against the occupying German forces on 5 May 1945, the day all the staff of the German University were arrested, and held the city until the Russian Army arrived four days later. One would have to mention the facts concerning Gentzen's political and military life that Vihan relates in [Vih95], namely his association with the SA, NSDAP and NSD Dozentenbund. Gentzen was interned by the Russian forces and held in poor conditions. He died of malnutrition after 3 months in internment.*

For more details see [Vih95].

In the 1950s, E W Beth independently invented semantic tableau calculi: purely syntactic calculi whose rules Beth derived from the semantics of classical propositional logic [Bet53, Bet55]. Tableau calculi are now known to be syntactic variants of cut-free sequent calculi.

9 Concluding Remarks

The systematic backward search method in the (cut-free) sequent calculus SK can be implemented on modern computers to give automatic programs that decide whether or not a sequent is derivable in SK. The field of research is known as automated deduction [Gal87] and the variant of sequent calculi called tableaux calculi are often used in this area [DGHP99].

Over the centuries, philosophers, mathematicians, and more recently computer scientists, have investigated numerous alternatives to classical propositional logic. Multi-valued logics reject the Law of the Excluded Middle and allow truth values other than just "true" and "false". Paraconsistent logics reject the Law of Non-Contradiction and allow a formula to be both "true" and "false". Modal and Temporal logics allow the truth value of a formula to change over place and time. Intuitionistic logic rejects $(\neg\neg\varphi) \rightarrow \varphi$, one half of the principle of double negation; see Exercise 3.8(8).[4] Relevant logics demand that φ and Γ share at least one common atomic formula for φ to be a logical consequence of Γ. Linear logic demands that every formula in Γ is used exactly once in a deduction of φ from Γ. For an introduction to some of these logics see [Pri01].

[4] [See Chapter 51 in this volume for more about intuitionistic logic.—Ed.]

Many of the proofs in this chapter only work for finite sets: for example, our systematic backward search procedure terminates only when the original sequent has a finite antecedent and succedent, and our proof of Theorem 5.8 depends upon the fact that backward search terminates. Nevertheless, many of the results hold even for infinite sets: for example, (the Deduction) Theorem 7.4. More complicated mathematical proofs are required to handle infinite sets, which are beyond the scope of this chapter. For more on sequent calculi and proof theory see [TS96].

I would appreciate any comments or criticisms that you might have about this chapter. Please do not hesitate to contact me either by post or email.

Acknowledgements: First I want to thank Helen Lauer, the editor of this volume, for her patience in waiting for over four months for me to complete this article. I hope the wait was worth it. Second I would like to thank Agnes Boskovitz, Jeremy Dawson and Paul Wong for their many constructive comments on an earlier version. All mistakes in this chapter were obviously caused by them and they should be ashamed of themselves.

References

[Avr96] A Avron. The method of hypersequents in proof theory of propositional non-classical logics. In C Steinhorn J Truss W Hodges, M Hyland, editor, *Logic: Foundations to Applications*, pages 1–32. Oxford Science Publications, 1996.

[Bel82] N D Belnap. Display logic. *Journal of Philosophical Logic*, 11:375–417, 1982.

[Bet53] E W Beth. On Padoa's method in the theory of definition. *Indag. Math.*, 15:330–339, 1953.

[Bet55] E W Beth. Semantic entailment and formal derivability. *Mededelingen der Koninklijke Nederlandse Akademie van Wetenschappen, Afd. Letterkunde*, 18:309–342, 1955.

[DGHP99] M D'Agostino, D Gabbay, R Hähnle, and J Posegga, editors. *Handbook of Tableaux Methods*. Kluwer, 1999.

[DM94] M D'Agostino and M Mondadori. The taming of the cut. Classical refutations with analytic cut. *Journal of Logic and Computation*, 4:285–319, 1994.

[Došš88] K Došen. Sequent systems and groupoid models, I. *Studia Logica*, 47:353–389, 1988.

[Došen89] K Došen. Sequent systems and groupoid models, II. *Studia Logica*, 48:41–65, 1989.

[Gal87] J H Gallier. *Logic for Computer Science: Foundations of Automatic Theorem Proving*. John Wiley and Sons, 1987.

[Gen35] G Gentzen. Untersuchungen über das logische schließen I and II. *Mathematische Zeitschrift*, 39:176–210 and 405–431, 1935. English translation: Investigations into logical deduction, in *The Collected Papers of Gerhard Gentzen*, M. E. Szabo (Ed), pp 68-131, North-Holland, 1969.

[Gor98] R Goré. Substructural logics on display. *Logic Journal of the Interest Group in Pure and Applied Logic*, 6(3):451–504, 1998.

[Ják34] S Jászkowski. On the rules of supposition in formal logic (in Polish). *Studia Logica (old series)*, 1:5–32, 1934. English Translation in *Polish Logic* 1920-39, S McCall, (Ed) Clarendon Press, Oxford, 1967, pp 232-258.

[JL83] P N Johnson-Laird. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. Cambridge University Press, 1983.

[OR01] J J O'Connor and E F Robertson. Gerhard Gentzen. `http://www-groups.dcs.st-andrews.ac.uk/~history/Mathematicians/Gentzen.html`, September 2001.

[Pri01] G Priest. *An introduction to non-classical logic*. Cambridge University Press, 2001.

[TS96] A Troelstra and H Schwichtenberg. *Basic Proof Theory*. Number 43 in Cambridge Tracts In Theoretical Computer Science. Cambridge University Press, 1996.

[Vih95] P Vihan. *Collegium Logicum*, volume 1 of *Annals of the Kurt Gödel Society*, chapter The Last Month of Gerhard Gentzen in Prague, pages 1–7. Springer-Verlag, 1995.