University of Southern Denmark

Logic for Computer Science

Course notes v0.5

Luís Cruz-Filipe

January 2024

Contents

Introduction

The purpose of these notes is to support the course on Logic for Computer Science, currently taught at the University of Southern Denmark as an elective course for Master students in both Computer Science and Mathematics.

These notes are meant not only as support to the lectures and exercise classes planned throughout the semester, but also as material for self-study for those students who cannot attend those classes. They include formal definitions accompanied by informal explanations meant to help understanding the basic concepts and gain the necessary intuition to work with the different topics presented. The presentation also includes numerous examples, together with exercises that can be used by the student to train concepts and techniques as they are introduced. Additional exercises can be found at the end of each chapter.

I want to thank Marco Peressotti for suggestions for some exercises, as well as for helping with typesetting figures.

Odense, March 2021

Luís Cruz-Filipe

Chapter 1

General notions about logic

1.1 What is a logic?

The notion of "logic" (as "the" logic) has been around since Ancient Greece, as a symbolic way to represent the reasoning process. However, it was only in the 19th century that a precise notion of the abstract concept started to develop. This started mainly through the works of Tarski, who first realized the independence between the symbols used in formulas and the entities they refer to.

Nowadays, there are several logics in current usage, and it is widely understood that the logic suited to one particular application does not need to be the same as for another. Besides the conceptual differences between the dozens of different logics available, several of their features may be combined together in multiple ways, giving rise to hundreds of potentially useful logics. Regardless of their particular aspects, though, all of these logics share three characteristics:

- a *language*, typically inductively defined, consisting of the set of formulas of the logic;
- a *semantics*, explaining how to assign meaning to the symbols in the language so that every formula can be assigned a *truth value*;
- a *deductive system*, allowing the mechanical derivation of new formulas from a set of *hypotheses*.

1.2 The language

Logic languages consist of a set of *well-formed formulas* (wffs, or simply formulas), and are usually inductively generated from a small set of symbols. For example, modal propositional logic uses a countable set $\mathcal{P} = \{p_1, \ldots, p_n, \ldots\}$ of propositional symbols, the connectives $\neg$, $\vee$, $\wedge$, $\rightarrow$ and $\leftrightarrow$, and the modalities $\Box$ and $\Diamond$. Wffs are defined inductively by means of a grammar:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \rightarrow \varphi) \mid (\varphi \leftrightarrow \varphi) \mid (\Box\varphi) \mid (\Diamond\varphi)$$

It is implicitly assumed that the set of wffs is decidable, and usually with low complexity. Although this is not a requirement of a logic, it is hard to imagine a useful logic where we cannot even decide whether something is a formula in reasonable time!

Example. When a language is defined by means of a grammar[1], it is simple to check whether a sequence of symbols corresponds to a wff: we match the sequence with one of the possible rules for the grammar, possibly recursively. If this terminates successfully, the sequence is a wff, and if some step fails the sequence is not a wff.

As an example, consider the grammar given above and the candidate formula $((p_1 \vee p_2) \to \neg p_1)$, where $p_1, p_2 \in \mathcal{P}$. We can see that $((p_1 \vee p_2) \to \neg p_1)$ has the form $(\varphi_1 \to \varphi_2)$ if we take φ_1 to be $(p_1 \vee p_2)$ and φ_2 to be $\neg p_1$. In turn, φ_1 has the form $(\varphi_{1,1} \vee \varphi_{1,2})$ with $\varphi_{1,1}$ is p_1 and $\varphi_{1,2}$ is p_2. Both $\varphi_{1,1}$ and $\varphi_{1,2}$ are of the form p for some $p \in \mathcal{P}$, so this branch succeeds. Finally, φ_2 has the form $\neg\varphi_{2,1}$, where $\varphi_{2,1}$ is $p_1 \in \mathcal{P}$, so this branch also succeeds. Therefore $((p_1 \vee p_2) \to \neg p_1)$ is a wff in this language.

For an example of a candidate formula that is not well-formed, consider $p_1\vee$. This sequence of symbols does not match any of the possibilities in the grammar for the language, and therefore it is not a wff. ◁

The process exemplified here is known in Computer Science as *parsing*.

In many logics, the structure of formulas is defined by more complex grammars, requiring some auxiliary concepts. For example, first-order logic formulas are built from predicates and terms, and the notion of term is itself inductively defined.

1.3 Semantics

The purpose of logic is to model some domain of the (real) world, namely in order to allow reasoning over it. Therefore, there needs to be a connection between the abstract symbols in the language and the actual objects in the domain being modeled. This connection is formalized by the notion of *interpretation* or *model*. An interpretation is a function assigning to each formula a *truth value*, taken from a set that is assumed to contain a distinguished element $\top$ (truth) and at least one other element. In order to define semantics for a logic, we therefore need to say what its set of truth values is and what its possible models are. *Classical*, or Boolean, logics assume that the set of truth values is the smallest possible, namely $\{\top, \bot\}$, but many useful logics have larger (even infinite) sets of truth formulas.

Interpretations are typically defined by induction, following the structural definition of the language. Again, there are some implicit computability requirements: interpretations should be computable functions, and computing the truth value of a formula should be efficient.

Because of the existence of a distinguished truth value, it is usually not important to know exactly what value is assigned to a formula, but only whether it is the distinguished value or not. We write

$$\mathfrak{M} \models \varphi$$

(read "model $\mathfrak{M}$ satisfies formula φ", or "$\mathfrak{M}$ is a model of φ") to say that $\mathfrak{M}(\varphi) = \top$. Using this notation, formulas can be divided in four groups.

- A formula φ is *valid*, or a *tautology*, written $\models \varphi$, if $\mathfrak{M} \models \varphi$ for every model $\mathfrak{M}$.
- A formula φ is *satisfiable* if $\mathfrak{M} \models \varphi$ for some model $\mathfrak{M}$.
- A formula φ is *falsifiable*, written $\not\models \varphi$, if $\mathfrak{M} \not\models \varphi$ for some model $\mathfrak{M}$.
- A formula φ is *unsatisfiable* if $\mathfrak{M} \not\models \varphi$ for every model $\mathfrak{M}$.

[1] Technically, by a context-free grammar. These are the grammars typically used to define logic languages.

The most important semantic concept is that of entailment. Let Γ be a set of formulas; we write $\mathfrak{M} \models \Gamma$ if $\mathfrak{M} \models \psi$ for every $\psi \in \Gamma$ Then we say that Γ *entails* a formula φ, or that φ is a (semantic) consequence of Γ, written

$$\Gamma \models \varphi$$

if $\mathfrak{M} \models \varphi$ for every $\mathfrak{M}$ such that $\mathfrak{M} \models \Gamma$. In particular, φ is *valid* if $\emptyset \models \varphi$, which justifies the notation $\models \varphi$. It is common to call the formulas in Γ the *hypotheses* of the entailment, and φ its *conclusion*.

The *entailment problem* for a logic is: given a set Γ and a formula φ, does $\Gamma \models \varphi$?

It is possible to propose several different semantics for the same logic. However, it is required that the notions of valid and satisfiable formula, as well as that of entailment, coincide for all the semantics – otherwise we are talking about a different logic. For example, the standard semantics for classical propositional logic uses valuations, but it is possible to give an alternative semantics based on Boolean algebras. However, the standard Kripke semantics over the same language gives us intuitionistic propositional logic, which is a different logic.

1.4 Deductive systems

In order to address the entailment problem, we need mechanical ways to express reasoning. In very abstract terms, we can characterize these mechanisms as algorithmic, non-deterministic ways to produce new formulas from sets of formulas. Such a system is usually known as a *deductive system*, and we write

$$\Gamma \vdash_D \varphi$$

(read "Γ derives φ in D", or "φ is a syntactic consequence of Γ in D") to denote that φ can be produced from Γ. In particular, we write $\vdash_D \varphi$ to denote that we can derive φ from the empty set in D. If D is clear from the context, we omit it and write simply $\vdash$ instead of $\vdash_D$.

It is quite common to present several deductive systems for the same logic. Usually, each of these will be more tailored for a particular application. The usefulness of these systems is measured by their computational complexity, on the one hand, and by their relationship to the semantics, on the other hand.

Definition. A deductive system D is said to be:

- *sound* if: whenever $\Gamma \vdash_D \varphi$, it is the case that $\Gamma \models \varphi$.
- *complete* if: whenever $\Gamma \models \varphi$, it is the case that $\Gamma \vdash_D \varphi$.

A deductive system is weakly sound/complete if the above implications hold when Γ is the empty set.

Ideally, we would like to find efficient, sound and complete deductive systems for a given logic. However, this is usually only possible for very simple logics: logics expressive enough to represent basic arithmetic are incomplete by Gödel's incompleteness theorem, and even in logics for which there are complete deductive systems the entailment decision problem can be undecidable or NP-complete. Typical approaches to minimize this problem include:

- consider only a decidable fragment of the language (changing the logic);
- allow systems with infinite derivations (corresponding to non-terminating instances of the decision procedure);

- consider sound, but incomplete decision procedures to retain decidability.

Most deductive systems are (rule) inference systems, consisting of a set of rules of the form

$$\frac{\alpha_1 \ldots \alpha_n}{\beta}$$

where $\alpha_1, \ldots, \alpha_n, \beta$ (schematic) formulas, possibly together with some side conditions. Rules are typically read as "if we can derive $\alpha_1, \ldots, \alpha_n$ from Γ, then we can also derive β from Γ". Rule inference systems are particularly suitable for mechanical implementations.

Chapter 2

Classical propositional logic

Classical propositional logic, also called Boolean logic, is one of the simplest logics in existence – but also one that is very widely used. While at first sight it is not expressive enough for modeling relations between data, decades of fruitful research in decision procedures for deciding its satisfiability problem have made it a very interesting logic in practice. In recent years, encoding complex problems in classical propositional logic has become a successful technique both in mathematics and in practical applications.

The simplicity of classical propositional logic also makes it a very natural starting point for the study of logic in general. In this chapter, we present this logic and use it to illustrate the abstract concepts that were discussed earlier in the general case. After defining its language and semantics, we introduce a number of deductive systems for classical propositional logic, illustrating the variety of methods that can be used to tackle the entailment problem for this logic. Several of these systems will be extended or adapted to more expressive logics in later chapters.

2.1 Language

The basic building blocks of propositional logic[1] are *propositional symbols*: abstract symbols, typically denoted p, q, r, p', p_1, etc., which correspond to properties in the real world that can be true or false. Propositional symbols abstract from the actual nature of these facts and how their truth or falsity is determined: propositional logic is concerned only about their interaction, and properties of that interaction.

Complex formulas are built from propositional symbols by using *connectives*: logical symbols that take one or two propositional formulas and produce more complex formulas. Typical connectives are negation (which returns a formula with meaning opposite to the original one's), conjunction (which takes two formulas and returns a new formula that expresses truth of both original formulas) or implication (which models a dependency between formulas).

Examples of properties that we could want to represent are "the sky is blue", or "the painting is nice". If we use propositional symbol p to represent that "the sky is blue" and propositional symbol q to represent that "the painting is nice", then formula $\neg p$ reads "the sky is not blue", while $p \to q$ reads "if the sky is blue, then the painting is nice".

Definition. A *propositional signature* is a countable set $\mathcal{P} = \{p_i \mid i \in \mathbb{N}\}$.

[1] Although this chapter is about classical propositional logic, the syntax is common to all variants of propositional logic.

We assume the propositional signature to be fixed from this point onwards. We reserve the notation $p_1, p_2, \ldots$ for the concrete symbols in $\mathcal{P}$, which we use when proving properties of propositional logic, and use p, q, r, p', etc., in concrete formulas.

Definition. The set of propositional formulas over $\mathcal{P}$ is defined inductively by the following grammar.

$$\varphi, \psi ::= p_i \mid (\neg\varphi) \mid (\varphi \vee \psi) \mid (\varphi \wedge \psi) \mid (\varphi \to \psi) \mid (\varphi \leftrightarrow \psi)$$

In other words: every propositional symbol is a propositional formula; and if φ and ψ are formulas, then so are $(\neg\varphi)$, $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, $(\varphi \to \psi)$ and $(\varphi \leftrightarrow \psi)$.

The connectives shown above are the most commonly used ones; as we will see later, they can be defined in terms of each other, and therefore there is some freedom among which ones we use.

The only unary connective is negation, which we already explained above. Besides the symbol $\neg$, which we use in these notes, negation is also often written as $\sim$ or as a vertical line above the negated formula (i.e., $\bar{\varphi}$ instead of $\neg\varphi$). This last alternative is especially common in SAT solving.

Conjunction ($\wedge$) and disjunction ($\vee$) correspond to the words "and" and "or" in English, respectively. Thus, $(\varphi \wedge \psi)$ expresses that φ and ψ are both true, while $(\varphi \vee \psi)$ states that at least one of φ and ψ is true. Exclusive disjunction, often read as "xor" and written $\dot{\vee}$, is another binary connective that is used often in Boolean circuits: $(\varphi \dot{\vee} \psi)$ states that *exactly one* of φ and ψ holds.

Implication ($\to$) corresponds to the English word "implies", or the construction "if ... then": formula $(\varphi \to \psi)$ is read as "φ implies ψ", or "if φ is true, then so is ψ". However, this should not be understood as stating a causal dependency, but only a logical dependency – see the discussion below on the paradoxes of implication. Equivalence or bi-implication ($\leftrightarrow$) is often written as "iff" in Mathematics textbooks, and $(\varphi \leftrightarrow \psi)$ states that the formulas φ and ψ are either both true or both false.

The formal syntax of propositional logic requires that every subformula be parenthesized. This can easily make formulas extremely hard to read. Therefore, there are some standard conventions on priority of operators:

- negation has the strongest priority;
- conjunction and disjunction have equal priority;
- implication and equivalence have the lowest priority.

Thus, we can write e.g. $\neg p \vee \neg q$ for $((\neg p) \vee (\neg q))$, or $p \vee q \to p \wedge q$ for $((p \vee q) \to (p \wedge q))$. However, we will also strive to avoid ambiguity, and keep parenthesis if removing them might make the formula confusing to read. Note that parentheses are *always* required in formulas mixing conjunction and disjunction, e.g. $(p \vee q) \wedge r$ and $p \vee (q \wedge r)$, but not in formulas that have several of one of these – we will show below that $(p \vee q) \vee r$ and $p \vee (q \vee r)$ are equivalent, which allows us simply to write $p \vee q \vee r$.

A good rule of thumb is: in case of doubt, keep the parentheses.

Exercise 1. Using the propositional symbols p and q with the intended meaning as in the text above, write logical formulas that correspond to the following statements.

(a) The sky is blue and the painting is nice.

(b) If the sky is not blue, then the painting is nice.

(c) It is not the case that the sky is blue and the painting is nice.

(d) If the sky is blue and the sky is not blue, then the painting is nice.

(e) The sky is blue, the painting is nice, the sky is blue, and the sky is not blue.

(f) If it is the case that the sky being blue implies that the sky is not blue, then the sky is not blue.

Exercise 2. Translate the following formulas into natural language, again using p and q with the intended meaning described in the text.

(a) $\neg p \vee \neg q$

(b) $\neg(p \vee q)$

(c) $(p \to q) \wedge (q \to p)$

(d) $p \vee \neg q \to (q \to p)$

(e) $p \to (q \to p)$

(f) $((p \to q) \wedge (\neg p \to q)) \to q$

These exercises illustrate why using a symbolic language is an advantage: it is easier to reason about statements expressed abstractly than using natural language. One should also be aware that, in natural language, the distinction between "and", "or" and "either... or" (corresponding to xor) is often blurred. For example, when we say "we can go to the movies, or we can go for a walk", "or" actually stands for $\wedge$: one would only use this construction if both sentences are true (compare with "we can go to the movies, or we can go to the Moon").

Proposition 1. The set of propositional formulas over $\mathcal{P}$ is decidable.

We omit a formal proof of this result, as it would require a background in recursion theory that is beyond the scope of these notes. Informally, given a formula we need to match it with one of the formation rules in the grammar for this language, and recursively apply the same procedure to any subformulas.

If we are tolerant with parentheses (meaning that we allow extra parentheses, and disambiguate any unparenthesized formulas by giving highest priority to the leftmost operator – so that the inambiguous $p \vee q \vee r$ stands for $(p \vee q) \vee r$, but also the ambiguous $p \vee q \wedge r$ stands for $(p \vee q) \wedge r$), then we can actually check that a formula is valid by checking that:

- the total number of opening and closing parentheses is equal;
- at no point in the formula is the number of closing parentheses higher than the number of opening parentheses (that is, this property holds for every string that is a prefix of the formula);
- the symbol immediately after an opening parenthesis must always be another opening parenthesis, a propositional symbol or a negation;
- the symbol immediately after a closing parenthesis or a propositional symbol must always be a closing parenthesis or a binary connective;

- the symbol immediately after a negation must be an opening parenthesis, a propositional symbol or another negation;

- the symbol after a binary connective must be an opening parenthesis or a propositional symbol.

Alternatively, the fact that we can write a parser for this language also shows its decidability.

Exercise 3. Check that the five properties given above characterize the set of well-formed formulas in propositional logic, provided that we are tolerant with parentheses in the sense described in the text.

2.2 Semantics

The classical semantics of propositional logic is a truth-functional semantics with two truth values. This means that there are exactly two truth values (only one of which is distinguished), and that the truth value of a formula is determined by the truth value of its subformulas. There are other possible semantics of propositional logic: intuitionistic semantics (which is not truth functional) and fuzzy semantics (where there are more than two truth values) are two well-known examples.

The two truth values of classical propositional logic are called *true* and *false*, with true naturally being the distinguished truth value. In logic literature, it is common to represent them as $\top$ and $\bot$, respectively; in programming languages, these are often implemented as 1 and 0, as the semantics can then be specified as simple functions on natural numbers. We will follow the practice in logic, in order to avoid the common misconception that 1 and 0 *are* the truth values of propositional logic.

The semantics of classical propositional logic is based on the notion of *valuation*.

Definition. A valuation, or *assignment*, is a function $V : \mathcal{P} \to \{\top, \bot\}$.

Definition. Satisfaction of a formula φ by a valuation V, denoted $V \models \varphi$, is defined inductively as follows.

$$
\begin{array}{ll}
V \models p_i \text{ iff } V(p_i) = \top & V \models \neg\varphi \text{ iff } V \not\models \varphi \\
V \models \varphi \vee \psi \text{ iff } V \models \varphi \text{ or } V \models \psi & V \models \varphi \wedge \psi \text{ iff } V \models \varphi \text{ and } V \models \psi \\
V \models \varphi \to \psi \text{ iff } V \not\models \varphi \text{ or } V \models \psi & V \models \varphi \leftrightarrow \psi \text{ iff } (V \models \varphi \text{ and } V \models \psi) \\
 & \qquad \text{or } (V \not\models \varphi \text{ and } V \not\models \psi)
\end{array}
$$

In the terminology of the previous section, from each valuation V we can define a model V^* as a function from formulas to truth values, defined as $V^*(\varphi) = \top$ iff $V \models \varphi$. In particular, $V^*(p) = V(p)$ for each propositional symbol p, and it is common practice to abuse notation and use V in both cases.

Example. We show directly from the semantics that any valuation V such that $V(p) = \bot$ and $V(q) = \top$ satisfies the formula $(p \wedge q) \to (p \wedge \neg q)$.

One possibility is to decompose the formula using the recursive definition of satisfaction.

$$\begin{aligned} V \models (p \wedge q) \to (p \wedge \neg q) \text{ iff } & V \not\models p \wedge q \text{ or } V \models p \wedge \neg q \\ \text{iff } & V \not\models p \text{ or } V \not\models q \text{ or } (V \models p \text{ and } V \models \neg q) \\ \text{iff } & V \not\models p \text{ or } V \not\models q \text{ or } (V \models p \text{ and } V \not\models q) \\ \text{iff } & V(p) = \bot \text{ or } V(q) = \bot \text{ or } (V(p) = \top \text{ and } V(q) = \bot) \end{aligned}$$

and since $V(p) = \bot$ the first of these conditions is met. Therefore $V \models (p \wedge q) \to (p \wedge \neg q)$.

Alternatively, we can intuitively "guess" the relevant part of the formula, and build the proof upwards.

$$\begin{aligned} V(p) = \bot \text{ implies } & V \not\models p \\ \text{implies } & V \not\models p \wedge q \\ \text{implies } & V \models (p \wedge q) \to (p \wedge \neg q) \end{aligned}$$

This proof is shorter, but it requires analyzing what the formula is stating in order to understand how to build the proof. The previous proof is longer, but more systematic. ◁

Observe that, in the previous example, we distinguish between the logical connectives as syntactic elements of the language and the English words corresponding to them, which we use for expressing properties of the semantics. It is extremely important to keep this distinction: although we use the word "or" when reading both "$\varphi \vee \psi$" and "$V \models \varphi$ or $V \models \psi$", its meaning is different: in the first case, it is the name of a symbol used in the formula, while in the second case it is a word in the English language with a well-defined meaning.[2]

Truth-functionality is also reflected in the fact that we can write *truth tables* for any connective $*$: tables that, given the different possibilities for whether $V \models \varphi$ and $V \models \psi$ hold, return the corresponding value for $V \models \varphi * \psi$. For example, the truth table for $\vee$ can be written in one of the following ways.

	$V \models \varphi$	$V \not\models \varphi$
$V \models \psi$	$V \models \varphi \vee \psi$	$V \models \varphi \vee \psi$
$V \not\models \psi$	$V \models \varphi \vee \psi$	$V \not\models \varphi \vee \psi$

φ	ψ	$\varphi \vee \psi$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$

Recall that a formula φ is satisfiable if $V \models \varphi$ for some V, and that it is valid ($\models \varphi$) if $V \models \varphi$ for every V.

Example. The previous example shows that $(p \wedge q) \to (p \wedge \neg q)$ is satisfiable. ◁

In general, showing satisfiability of a formula requires "guessing" a valuation that makes that formula true, while validity requires proving that *all* valuations behave in that way.

[2]This distinction can also be described by distinguishing between the *logic level*, at which formulas are written, and the *meta-logical level*, at which properties of the formulas are discussed. When studying logic, it is important to keep the distinction between these two levels clear at all times.

Example. The formula $(p \wedge q) \to (p \wedge \neg q)$ is not valid. We show this by constructing a valuation that falsifies it. Arguing as above, for any valuation V we have:

$$\begin{aligned} V \models (p \wedge q) \to (p \wedge \neg q) \text{ iff } & V \not\models p \wedge q \text{ or } V \models p \wedge \neg q \\ \text{iff } & V \not\models p \text{ or } V \not\models q \text{ or } (V \models p \text{ and } V \models \neg q) \\ \text{iff } & V \not\models p \text{ or } V \not\models q \text{ or } (V \models p \text{ and } V \not\models q) \\ \text{iff } & V(p) = \bot \text{ or } V(q) = \bot \text{ or } (V(p) = \top \text{ and } V(q) = \bot) \end{aligned}$$

and if we choose V_0 such that $V_0(p) = \top$ and $V_0(q) = \top$ then none of the conditions in the last line hold for V_0. Therefore $V_0 \not\models p \wedge q \to p \wedge \neg q$, and therefore $p \wedge q \to p \wedge \neg q$ is not valid. ◁

Example. The formula $p \to (q \to p)$ is valid. We show this by contradiction. Given a valuation V, we have that

$$\begin{aligned} V \not\models p \to (q \to p) \text{ iff } & V \models p \text{ and } V \not\models q \to p \\ \text{iff } & V \models p \text{ and } V \models q \text{ and } V \not\models p \\ \text{iff } & V(p) = \top \text{ and } V(q) = \top \text{ and } V(p) = \bot \end{aligned}$$

and the condition in the last line cannot be satisfied by any V, since $\top \neq \bot$. ◁

Exercise 4. For each of the following formulas, decide whether they are satisfiable by either exhibiting a valuation that satisfies them or proving that no such valuation exists.

(a) $p \wedge q$

(b) $p \to p \vee q$

(c) $\neg p \vee (p \vee q \to \neg q)$

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

(e) $p \to \neg p$

(f) $p \wedge q \wedge p \to \neg q$

Exercise 5. For each of the formulas in the previous exercise, decide whether they are valid. In the negative case, exhibit a valuation that falsifies them.

Entailments can be analyzed in a similar way, provided that the set of hypotheses is finite.

Example. We show that $\{(p \vee q) \to \neg p\} \models \neg p$. Again we proceed by contradiction: we suppose that there is a valuation V that makes all hypotheses of this entailment true and the conclusion false, that is, $V \models (p \vee q) \to \neg p$ but $V \not\models \neg p$. We get:

$$\begin{aligned} V \models (p \vee q) \to \neg p \text{ and } V \not\models \neg p \text{ iff } & (V \not\models p \vee q \text{ or } V \models \neg p) \text{ and } V \models p \\ \text{iff } & ((V \not\models p \text{ and } V \not\models q) \text{ or } V \not\models p) \text{ and } V \models p \\ \text{iff } & ((V(p) = \bot \text{ and } V(q) = \bot) \text{ or } V(p) = \bot) \text{ and } V(p) = \top \end{aligned}$$

The first two conditions imply that $V(p) = \bot$, which contradicts the requirement $V(p) = \top$. Therefore we conclude that this entailment holds. ◁

Example. Consider now the entailment $\{p \vee q, p \to r, q \to s\} \models r \vee s$. In order to show that it holds, we again proceed by contradiction, assuming that we can find V such that $V \models p \vee q$, $V \models p \to r$, $V \models q \to s$ and $V \not\models r \vee s$.

$V \models p \vee q$ and $V \models p \to r$ and $V \models q \to s$ and $V \not\models r \vee s$
iff $(V \models p$ or $V \models q)$ and $(V \not\models p$ or $V \models r)$ and $(V \not\models q$ or $V \models s)$ and $V \not\models r$ and $V \not\models s$
iff $\underbrace{(V(p) = \top \text{ or } V(q) = \top)}_{(1)}$ and $\underbrace{(V(p) = \bot \text{ or } V(r) = \top)}_{(2)}$
and $\underbrace{(V(q) = \bot \text{ or } V(s) = \top)}_{(3)}$ and $\underbrace{V(r) = \bot}_{(4)}$ and $\underbrace{V(s) = \bot}_{(5)}$

Since $V(r)$ cannot be simultaneously equal to $\bot$ and $\top$, from (4) and (2) we conclude that $V(p) = \bot$. Likewise, from (5) and (3) we conclude that $V(q) = \bot$. But then none of the conditions in (1) hold, and we get a contradiction.

Therefore the given entailment holds. ◁

Exercise 6. Decide whether the following entailments hold.

(a) $\{p \vee q, p \wedge q\} \models p$
(b) $\{\neg p \to p\} \models p$
(c) $\{p \wedge q, \neg p\} \models r$
(d) $\{p\} \models q \to p$
(e) $\{p \vee q, p \to r, q \to r\} \models r$
(f) $\{p \to (q \wedge r), \neg p\} \models q \vee r$

In the general case, we rely on the fact that entailment is compact: if $\Gamma \models \varphi$, then there is a finite set $\Gamma_{\mathsf{fin}} \subseteq \Gamma$ such that $\Gamma_{\mathsf{fin}} \models \varphi$. A direct proof of this result is not simple to provide, and we will instead show later that it is a consequence of the completeness of a particular deductive system for propositional logic (Theorem 14). If we want to prove an entailment $\Gamma \models \varphi$ with Γ infinite, we first need to select a finite subset of Γ that suffices to establish the entailment.

Example. Let $\Gamma = \{p_{2i} \vee p_{2i+1} \mid i \in \mathbb{N}\}$. Then $\Gamma \models (\neg p_0 \wedge \neg p_2) \to (p_1 \wedge p_3)$.

In order to establish this entailment, we observe that the relevant formulas in Γ are $p_0 \vee p_1$ and $p_2 \vee p_3$. Indeed, if V is such that $V \models \Gamma$ and $V \not\models (\neg p_0 \wedge \neg p_2) \to (p_1 \wedge p_3)$, then in particular we have:

$V \models p_0 \vee p_1$ and $V \models p_2 \vee p_3$ and $V \not\models (\neg p_0 \wedge \neg p_2) \to (p_1 \wedge p_3)$
iff $(V \models p_0$ or $V \models p_1)$ and $(V \models p_2$ or $V \models p_3)$
and $V \models \neg p_0 \wedge \neg p_2$ and $V \not\models p_1 \wedge p_3$
iff $(V \models p_0$ or $V \models p_1)$ and $(V \models p_2$ or $V \models p_3)$
and $V \models \neg p_0$ and $V \models \neg p_2$ and $(V \not\models p_1$ or $V \not\models p_3)$
iff $\underbrace{(V(p_0) = \top \text{ or } V(p_1) = \top)}_{(1)}$ and $\underbrace{(V(p_2) = \top \text{ or } V(p_3) = \top)}_{(2)}$
and $\underbrace{V(p_0) = \bot}_{(3)}$ and $\underbrace{V(p_2) = \bot}_{(4)}$ and $\underbrace{(V(p_1) = \bot \text{ or } V(p_3) = \bot)}_{(5)}$

Since $V(p_0)$ cannot simultaneously be $\top$ and $\bot$, from (3) and (1) we conclude that $V(p_1) = \top$. Likewise, from (4) and (2) we conclude that $V(p_3) = \top$. But this contradicts (5), therefore no such V exists. ◁

From the semantics of propositional logic, we can derive relationships between the different connectives.

Example. For every pair of formulas φ and ψ, the formulas $\varphi \vee \psi$ and $\neg(\neg\varphi \wedge \neg\psi)$ are equivalent, in the sense that $V \models \varphi \vee \psi$ iff $V \models \neg(\neg\varphi \wedge \neg\psi)$. Indeed,

$$\begin{aligned} V \models \neg(\neg\varphi \wedge \neg\psi) &\text{ iff } V \not\models \neg\varphi \wedge \neg\psi \text{ iff } V \not\models \neg\varphi \text{ or } V \not\models \neg\psi \\ &\text{iff } V \models \varphi \text{ or } V \models \psi \text{ iff } V \models \varphi \vee \psi \end{aligned}$$

and therefore the two formulas are equivalent. ◁

This allows us to define some connectives in terms of others, which is useful when we want to prove results about the semantics of propositional formulas by structural induction: by assuming that some connectives are defined from others, the number of cases in the inductive step of the proof becomes smaller.

Definition. Let $\mathcal{C}$ be a set of connectives. A binary connective $\star$ is *representable* in $\mathcal{C}$ if the formula $\varphi \star \psi$ is equivalent to a formula that only uses φ, ψ and connectives from $\mathcal{C}$.

$\mathcal{C}$ is called *functionally complete* if every binary connective is representable in $\mathcal{C}$.

Since the semantics of propositional logic is truth-functional, a functionally complete set of connectives can express any property that can be written in propositional logic using any connectives.

Example. Consider the set of connectives $\{\neg, \vee\}$. Both $\wedge$ and $\rightarrow$ are representable in this set. Indeed, $\varphi \wedge \psi$ and $\varphi \rightarrow \psi$ are equivalent to $\neg(\neg\varphi \vee \neg\psi)$ and $\neg\varphi \vee \psi$, respectively.

$$\begin{aligned} V \models \neg(\neg\varphi \vee \neg\psi) &\text{ iff } V \not\models \neg\varphi \vee \neg\psi \\ &\text{iff } V \not\models \neg\varphi \text{ and } V \not\models \neg\psi \\ &\text{iff } V \models \varphi \text{ and } V \models \psi \\ &\text{iff } V \models \varphi \wedge \psi \end{aligned} \qquad \begin{aligned} V \models \neg\varphi \vee \psi &\text{ iff } V \models \neg\varphi \text{ or } V \models \psi \\ &\text{iff } V \not\models \varphi \text{ or } V \models \psi \\ &\text{iff } V \models \varphi \rightarrow \psi \end{aligned}$$

In general, given a formula that uses conjunctions and implications, we can obtain a formula equivalent to it that only uses negations and disjunctions by recursively transforming it using these properties:

$$p_i^\circ = p \qquad (\neg\varphi)^\circ = \neg\varphi^\circ \qquad (\varphi \vee \psi)^\circ = \varphi^\circ \vee \psi^\circ$$
$$(\varphi \wedge \psi)^\circ = \neg(\neg\varphi^\circ \vee \neg\psi^\circ) \qquad (\varphi \rightarrow \psi)^\circ = \neg\varphi^\circ \vee \psi^\circ$$

For example,

$$\begin{aligned} (((p \wedge q) \vee (p \wedge \neg q)) \rightarrow p)^\circ &= \neg((p \wedge q) \vee (p \wedge \neg q))^\circ \vee p^\circ \\ &= \neg\left((p \wedge q)^\circ \vee (p \wedge \neg q)^\circ\right) \vee p \\ &= \neg\left(\neg\left(\neg p^\circ \vee \neg q^\circ\right) \vee \neg\left(\neg p^\circ \vee \neg\neg q^\circ\right)\right) \vee p \\ &= \neg(\neg(\neg p \vee \neg q) \vee \neg(\neg p \vee \neg\neg q)) \vee p \end{aligned}$$

which is equivalent to the original formula. ◁

Exercise 7. Show that all connectives in $\{\neg, \vee, \wedge, \to, \leftrightarrow\}$ are representable in the following sets of connectives.

(a) $\{\neg, \to\}$ (b) $\{\neg, \wedge\}$ (c) $\{\bot, \to\}$ (d) $\{\bar{\wedge}\}$

In the last two sets, $\bot$ is a 0-ary connective (read "false") such that $V \not\models \bot$ for any V, and $\bar{\wedge}$ is a binary connective (read "nand", shorthand for "not and") such that $V \models \varphi \mathbin{\bar{\wedge}} \psi$ iff $V \not\models \varphi$ or $V \not\models \psi$.

In order to prove that $\{\neg, \vee\}$ is functionally complete, however, we need to show that *all* binary connectives are representable in $\{\neg, \vee\}$. This is more complicated than the example below. We start with a simple observation.

Proposition 2. There are exactly 16 possible binary connectives in classical propositional logic.

Proof. Let $*$ be a binary connective. Since the semantics of classical propositional logic is truth-functional, the property $V \models \varphi * \psi$ can be represented by a truth table whose entries are the properties $V \models \varphi$ and $V \models \psi$. Since each of the latter either holds or does not hold, there are four possible inputs to the truth table; since the result also either holds or does not hold, there are two possible outputs. Therefore the total number of possible truth tables is $2^4 = 16$. □

Lemma 1. The set $\{\neg, \wedge, \vee\}$ is functionally complete.

Proof. Let $*$ be an arbitrary binary connective. In order to construct a formula equivalent to $\varphi * \psi$, we look at the truth table for $*$. For each entry of the table where $\varphi * \psi$ holds, we write the formula $\gamma_\varphi \wedge \gamma_\psi$, where γ_θ is θ if θ holds in the corresponding entry, and $\neg\theta$ otherwise. Then $\varphi * \psi$ is equivalent to the disjunction of all these formulas.

In the particular case where there are no formulas (because $\varphi * \psi$ never holds), we write $\varphi \wedge \neg\varphi$. □

Example. Consider the connective $\to$ and its truth table, given below.

φ	ψ	$\varphi \to \psi$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$

Applying the construction described in the previous proof, we see that there are three entries where $\varphi \to \psi$ holds. These entries generate the formulas $\varphi \wedge \psi$, $\neg\varphi \wedge \psi$ and $\neg\varphi \wedge \neg\psi$. Therefore, $\varphi \to \psi$ is equivalent to $(\varphi \wedge \psi) \vee (\neg\varphi \wedge \psi) \vee (\neg\varphi \wedge \neg\psi)$. ◁

As the previous example shows, the formulas obtained by this construction are not necessarily the shortest possible.

Lemma 1 simplifies proofs of functional completeness: if we can represent any connective in $\{\neg, \wedge, \vee\}$ by a set $\mathcal{C}$, then $\mathcal{C}$ is functionally complete. More generally, if $\mathcal{C}$ can represent any connective in $\mathcal{C}'$ and $\mathcal{C}'$ is functionally complete, then $\mathcal{C}$ is also functionally complete.

Theorem 1. The sets of connectives $\{\neg, \vee\}$, $\{\neg, \rightarrow\}$, $\{\neg, \wedge\}$, $\{\bot, \rightarrow\}$ and $\{\bar{\wedge}\}$ are all functionally complete.

Proof. Using Lemma 1, it suffices to show that we can represent any connective in $\{\neg, \wedge, \vee\}$ by each of these sets.

For $\{\neg, \vee\}$, this amounts to showing that conjunction is representable, which we did in a previous example.

For $\{\neg, \rightarrow\}$, we observe that $\varphi \vee \psi$ is equivalent to $\neg\varphi \rightarrow \psi$. Functional completeness now follows from functional completeness of $\{\neg, \vee\}$.

For $\{\neg, \wedge\}$ the conclusion follows similarly from the observation that $\varphi \vee \psi$ is equivalent to $\neg(\neg\varphi \wedge \neg\psi)$.

For $\{\bot, \rightarrow\}$ we observe that $\neg\varphi$ is equivalent to $\varphi \rightarrow \bot$ (since in this implication the consequent can never be true). The thesis then follows from the functional completeness of $\{\neg, \rightarrow\}$.

Finally, for $\{\bar{\wedge}\}$ we observe that $\neg\varphi$ is equivalent to $\varphi \mathbin{\bar{\wedge}} \varphi$, and $\varphi \wedge \psi$ is equivalent to $\neg(\varphi \mathbin{\bar{\wedge}} \psi)$. The thesis then follows from the functional completeness of $\{\neg, \wedge\}$. □

Exercise 8. Consider the binary connective $\bar{\vee}$, read "nor", whose semantics are given by $V \models \varphi \mathbin{\bar{\vee}} \psi$ iff $V \not\models \varphi$ and $V \not\models \psi$. Show that $\{\bar{\vee}\}$ is functionally complete.

One of the consequences of the semantics of classical propositional logic is that the truth value of a formula φ is determined by the propositional symbols that appear in φ. We phrase this simple observation as a useful lemma.

Lemma 2. Let φ be a propositional formula and V_1 and V_2 be valuations such that $V_1(p) = V_2(p)$ for every propositional symbol p occurring in φ. Then $V_1 \models \varphi$ iff $V_2 \models \varphi$.

Proof. By structural induction on φ. If φ is a propositional symbol p_i, then $V_1(p_i) = V_2(p_i)$ by hypothesis, and the thesis follows immediately.

Suppose now that φ is $\neg\psi$. Since every propositional symbol occurring in ψ also occurs in φ, the induction hypothesis guarantees that $V_1 \models \psi$ iff $V_2 \models \psi$. Then $V_1 \models \neg\psi$ iff $V_1 \not\models \psi$ iff $V_2 \not\models \psi$ iff $V_2 \models \varphi$.

Suppose that φ is built from ψ and θ by applying one of the connectives $\vee$, $\wedge$, $\rightarrow$ or $\leftrightarrow$. Again we observe that every propositional symbol occurring in ψ or θ also occurs in φ, so the induction hypothesis applies to both these formulas. Then:

- $V_1 \models \psi \vee \theta$ iff ($V_1 \models \psi$ or $V_1 \models \theta$) iff ($V_2 \models \psi$ or $V_2 \models \theta$) iff $V_2 \models \psi \vee \theta$;
- $V_1 \models \psi \wedge \theta$ iff ($V_1 \models \psi$ and $V_1 \models \theta$) iff ($V_2 \models \psi$ and $V_2 \models \theta$) iff $V_2 \models \psi \wedge \theta$;
- $V_1 \models \psi \rightarrow \theta$ iff ($V_1 \not\models \psi$ or $V_1 \models \theta$) iff ($V_2 \not\models \psi$ or $V_2 \models \theta$) iff $V_2 \models \psi \rightarrow \theta$;
- $V_1 \models \psi \leftrightarrow \theta$ iff ($V_1 \models \psi$ iff $V_1 \models \theta$) iff ($V_2 \models \psi$ iff $V_2 \models \theta$) iff $V_2 \models \psi \leftrightarrow \theta$.

In all cases the thesis holds, hence the result follows by structural induction. □

This lemma yields a straightforward algorithm to determine whether $\models \varphi$.

Theorem 2 (Decidability of validity)**.** The validity problem for classical propositional logic is decidable.

Proof. Let φ be a formula and $\{q_1, \ldots, q_n\}$ be the propositional symbols that appear in φ. For $i = 0, \ldots, 2^n - 1$, let V_i be the valuation such that $V(q_j)$ is $\top$ if the j-th digit of the binary representation of i is 1, and $\bot$ otherwise. If $V_i \models \varphi$ for all i, then $\models \varphi$, as every valuation coincides with some V_i on $\{q_1, \ldots, q_n\}$; otherwise $\not\models \varphi$.

Since all V_i can be constructed in finite time, and evaluating whether $V_i \models \varphi$ can also be done in finite time, by enumerating and testing the finite set of all V_i it is possible to decide whether $\models \varphi$. □

Corollary 1 (Decidability of satisfiability)**.** The satisfiability problem for classical propositional logic is decidable.

Proof. Immediate consequence of the previous theorem and the fact that φ is satisfiable iff $\neg\varphi$ is not valid. □

The algorithm embodied in the proof above is called the *method of truth tables*, and it is very inefficient – in fact, it is the most *inefficient* decision procedure in use for propositional logic.[3]

Example. Consider the formula $p \to (q \to p)$. This formula has two propositional symbols, so we need to consider four valuations. To decide its validity by the method of truth tables, we make a table as follows.

p	q	$q \to p$	$p \to (q \to p)$
$\top$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$	$\top$
$\bot$	$\top$	$\bot$	$\top$
$\bot$	$\bot$	$\top$	$\top$

Since in all cases the formula has truth value $\top$, we conclude that it is valid. ◁

Compare this argument with the semantic reasoning discussed in p. 10. Although for this formula the method of truth tables might not seem like too much additional work, the number of valuations increases exponentially with the number of propositional symbols; as such, blindly going through all of them quickly becomes infeasible.

Corollary 2. If Γ is a finite set of formulas and φ is a formula, then $\Gamma \models \varphi$ is decidable.

Proof. We repeat the construction in the proof of Theorem 2, but now considering also the symbols that occur in formulas in Γ. Since Γ is finite, the number of such symbols is finite, and we can check for each valuation that makes φ false whether it also makes some formula in Γ false; if this is not the case, then the entailment does not hold. □

[3]This should be understood as: of all decision procedures *actually* used, this is the most inefficient one.

In case Γ is infinite, though, this construction fails because it may be necessary to consider an infinite number of valuations.

Example. Consider the entailment $\{p_i \to p_0, \neg p_i \mid i > 0\} \models p_0$. This entailment does not hold, but determining it requires processing an infinite amount of information. ◁

In the next sections, we focus on more efficient deductive systems for classical propositional logic. All the systems we will study are sound and complete, and the proofs of these properties typically rely on the following results.

Definition. A set of formulas Γ is *inconsistent* if there is no valuation V such that $V \models \gamma$ for all $\gamma \in \Gamma$.

We typically write $V \models \Gamma$ with intended meaning that $V \models \gamma$ for all $\gamma \in \Gamma$. So Γ is inconsistent if $V \not\models \Gamma$ for every valuation V.

Inconsistency deals with the relationship among the formulas in Γ rather than with the formulas themselves: individually, the formulas of an inconsistent may all be satisfiable, just not by the same valuation.

Example. Consider the set of formulas $\Gamma = \{p \vee q, p \to \neg q, p \leftrightarrow q\}$. This set is inconsistent: if $V \models \Gamma$, then the last formula implies that $V(p) = V(q)$. But if $V(p) = V(q) = \bot$, then $V \not\models p \vee q$; and if $V(p) = V(q) = \top$, then $V \not\models p \to \neg q$. Therefore, $V \not\models \Gamma$ for any V.

However, all formulas in Γ are individually satisfiable. ◁

Exercise 9. Which of the following sets of formulas are inconsistent?

(a) $\{p, \neg(p \to q)\}$

(b) $\{\neg p, \neg(p \to q)\}$

(c) $\{\neg(p \vee q), \neg p \to q\}$

(d) $\{(p \wedge q) \to \neg r, \neg r \vee p, \neg r \vee q\}$

(e) $\{p \vee q \vee r, \neg(p \wedge q), \neg r \wedge q, \neg q \to \neg p\}$

Lemma 3. Let Γ be a set of propositional formulas and φ be a formula. Then $\Gamma \models \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.

Proof. Suppose there exists V such that $V \models \Gamma \cup \{\neg\varphi\}$. Then $V \models \Gamma$ but $V \not\models \varphi$, so $\Gamma \not\models \varphi$.

Suppose now that $\Gamma \cup \{\neg\varphi\}$ is inconsistent and let V be a valuation such that $V \models \Gamma$. Then necessarily $V \not\models \neg\varphi$, whence $V \models \varphi$. Therefore $\Gamma \models \varphi$. □

We conclude this section with a discussion on implication. While disjunction and conjunction correspond to concepts in natural language, this is not the case for implication: typically, the construction "if φ then ψ" assumes a causal relationship between φ and ψ that is not captured by the semantics of propositional logic.

This issue has been discussed extensively over the centuries, with several authors pointing out the unexpected consequences of the definition we gave of logical implication. These include a number of unintuitive valid logical formulas, often known as the *paradoxes of implication*. The most well-known ones are *ex falso quodlibet*,[4] stating that anything follows from a contradiction ($(p \wedge \neg p) \to q$ or $p \to (\neg p \to q)$), and the irrelevance of premise, stating that anything implies a true proposition ($p \to (q \to p)$). Other paradoxes often cited include $(\neg(p \to q)) \to (p \wedge \neg q)$ or $p \to (q \vee \neg q)$.

The answer to this discussion may not be completely satisfactory, but it is the pragmatic one: since the semantics of propositional logic is truth-functional, there are only 16 possible semantics for binary connectives. The intuitive semantics of implication tells us that, if $\varphi \to \psi$ holds:

- if φ is true, then ψ should be true (in other words, if $V \models \varphi$ and $V \not\models \psi$, then $V \not\models \varphi \to \psi$; and if $V \models \varphi$ and $V \models \psi$, then $V \models \varphi \to \psi$);
- if φ is false, then we know nothing about ψ's truth value (in other words, if $V \not\models \varphi$, then it is irrelevant whether $V \models \psi$ in order to decide whether $V \models \varphi \to \psi$).

The second observation gives us two options: either $V \not\models \varphi$ implies that $V \models \varphi \to \psi$ (which is our definition) or $V \not\models \varphi$ implies that $V \not\models \varphi \to \psi$. But in this second option $V \models \varphi \to \psi$ only holds if both $V \models \varphi$ and $V \models \psi$, which is the semantics of conjunction.

Therefore, the semantics given is the one that most closely captures the intuitive meaning of implication. It is common to call it *material implication* to distinguish it from the causal implication typically used in natural language.

Exercise 10. Show that all paradoxes of implication listed in the text are valid formulas.

2.3 Tableaux calculus

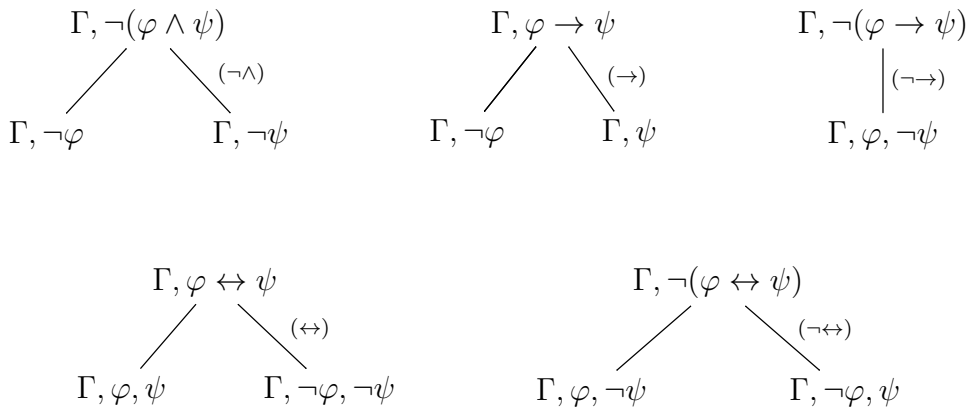
One of the simplest, and yet most powerful, deductive systems for classical logics is the method of semantic tableaux.[5] Tableaux calculi operate by constructing a tree where each node is labeled with a set of formulas. Each node is expanded by selecting one of its formulas and applying a rule to it, based on its main connective, yielding typically one or two descendants. The process stops when no formula in any node can be simplified further, or when every node contains a semantic contradiction.

Definition. A *propositional tableau* for a set Γ of propositional formulas is a labeled tree whose root contains Γ, and where the descendants of each node are obtained by applying one of the following rules to it.

$$\frac{\Gamma, \neg\neg\varphi}{\Gamma, \varphi}\,(\neg\neg) \qquad \frac{\Gamma, \varphi \vee \psi}{\Gamma, \varphi \;\big|\; \Gamma, \psi}\,(\vee) \qquad \frac{\Gamma, \neg(\varphi \vee \psi)}{\Gamma, \neg\varphi, \neg\psi}\,(\neg\vee) \qquad \frac{\Gamma, \varphi \wedge \psi}{\Gamma, \varphi, \psi}\,(\wedge)$$

[4] Latin for "Anything follows from false"

[5] *Tableau*, plural *tableaux*, is the French word for painting.



Semantic tableaux derive their name from the fact that the rules are very direct translations of the semantics of the connectives. The rule for conjunction ($\wedge$) illustrates that all formulas in one node are meant to be true at the same time, while the rule for disjunction ($\vee$) shows that descendants to the same node are seen as alternatives. Negation is also used to represent that a formula is false, capitalizing on the fact that we are working with a classical logic.

Using these intuitions, each pair of rules in the tableaux calculus corresponds to the semantics of a connective (in the case where it is used to build a true formula, and in the case where it is used to build a false formula). For example, the two rules for implication express the fact that $\varphi \to \psi$ is true if φ is false or ψ is true; and $\varphi \to \psi$ is false if φ is true and ψ is false. The rules select a formula in a node and simplify it, in a way that is very similar to the semantic analysis that we showed previously.

A leaf in a tableau is said to be *contradictory* or an *absurd* if it contains both a formula φ and its negation $\neg\varphi$. A tableau is *closed* if all its leaves are either contradictory or no more rules can be applied; otherwise it is called *open*.

We write $\Gamma \vdash_S \varphi$ if there is a closed tableau for $\Gamma \cup \{\neg\varphi\}$ whose leaves are all contradictory. In particular, $\vdash_S \varphi$ if $\emptyset \vdash_S \varphi$. The letter "S" is a reference both to the fact that these are semantic tableaux and to the logician Raymond Smullyan, who contributed to their development.

We now show some examples of tableaux. It is instructive to compare these examples with the semantic analysis of the same formulas and entailments in the previous section.

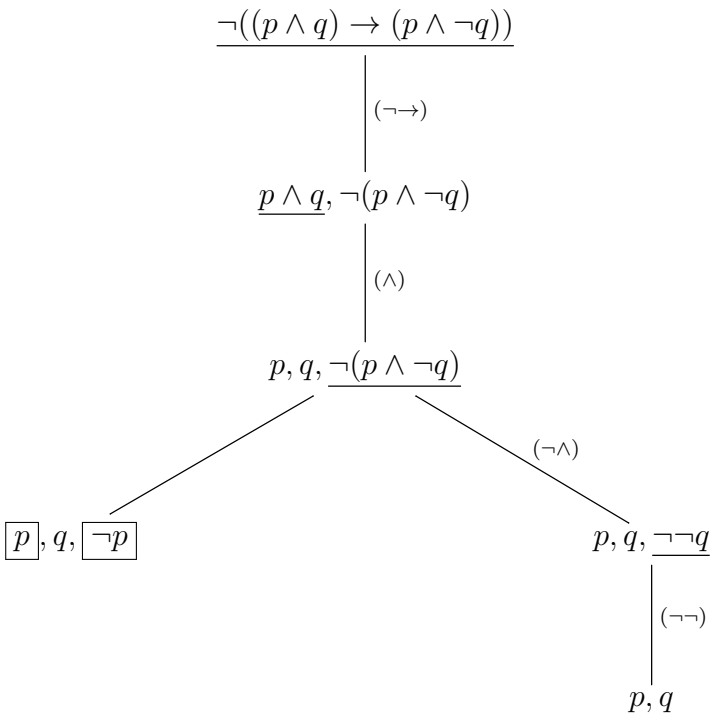
Example. We begin by showing that $\vdash_S p \to (q \to p)$. We start by negating this formula, and then systematically apply rules until we cannot proceed any further.

The formula to which a rule is applied at each node is underlined, and contradictory formulas in leaves are boxed.

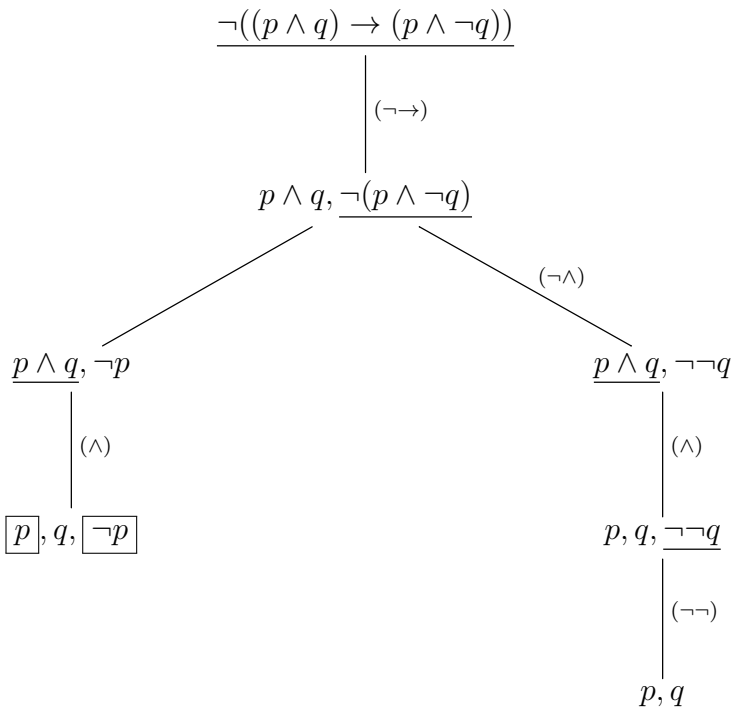
$$\underline{\neg(p \to (q \to p))}$$
$$\Big|\ (\neg\to)$$
$$p, \underline{\neg(q \to p)}$$
$$\Big|\ (\neg\to)$$
$$\boxed{p}, q, \boxed{\neg p}$$

The only leaf in this tableau contains a contradiction, so we conclude that $\vdash_S p \to (q \to p)$. $\triangleleft$

Example. It is not the case that $\vdash_S (p \wedge q) \to (p \wedge \neg q)$. Indeed, if we start a tableau with $\neg((p \wedge q) \to (p \wedge \neg q))$, we obtain the following tree.

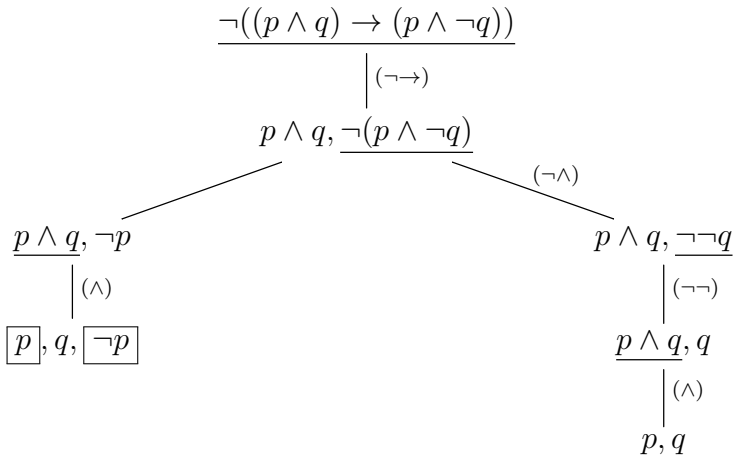


This tableau is closed, but it contains a leaf with no contradictions, and therefore we cannot conclude anything from it. However, in the second step we had a choice of which formula to expand; one might wonder where choosing the other formula would lead to a different result. The result is shown below.



Again we obtain a closed tableau with a non-contradictory leaf.

Finally, we could have switched the order of application of the last two rules on the right branch, obtaining yet another open tableau for the same formula.



Since these are all the closed tableaux with root $\neg((p \wedge q) \to (p \wedge \neg q))$, we can conclude that $\nvdash_S (p \wedge q) \to (p \wedge \neg q)$. ◁

The three tableaux in the last example are very similar to each other, as they only differ in the order in which the different formulas are expanded. In particular, they all contain the same leaves. We will see shortly that, in general, we do not need to consider all possible ways in which to build a tableau: as long as we can build *one* closed tableau with root $\neg\varphi$ and a non-contradictory leaf, we can guarantee that $\nvdash_S \varphi$. Therefore, it is often preferable to apply unary rules for as long as possible, in order to keep the tableau small.

The leaf that does not contain a contradiction suggests the valuation V_0 presented earlier to show that this formula is falsifiable – $V_0(p) = V_0(q) = \top$.

Exercise 11. For each of the following formulas, build a tableau with their negation as root to decide whether they are derivable in the tableaux calculus for propositional logic. In the negative case, use a non-contradictory leaf to construct a valuation that satisfies their negation.

(a) $p \wedge q$

(b) $p \to p \vee q$

(c) $\neg p \vee (p \vee q \to \neg q)$

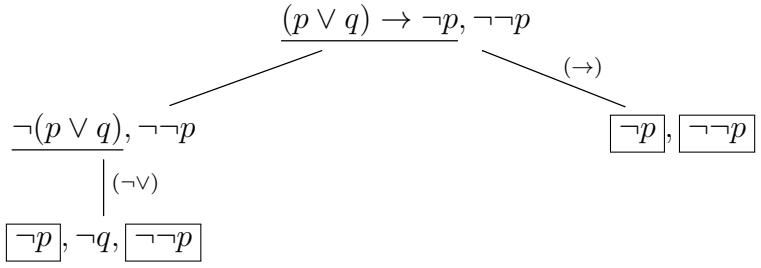
(d) $(p \to q) \wedge (r \to q) \wedge \neg q$

(e) $p \to \neg p$

(f) $p \wedge q \wedge p \to \neg q$

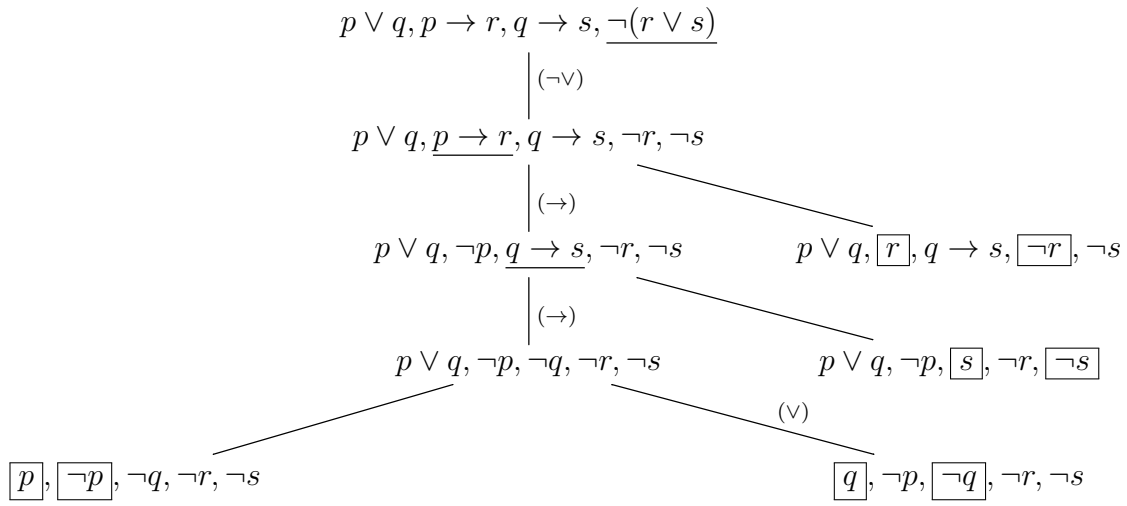
If we want to reason with hypotheses, the construction is similar.

Example. We show that $\{(p \vee q) \to \neg p\} \vdash_S \neg p$. We start constructing a tableau by labeling its root with $(p \vee q) \to \neg p, \neg\neg p$.



Since all leaves of this tableau are contradictory, we conclude that $\{(p \vee q) \to \neg p\} \vdash_S \neg p$. ◁

Example. As a next example, we show that $\{p \vee q, p \to r, q \to s\} \vdash_S r \vee s$.

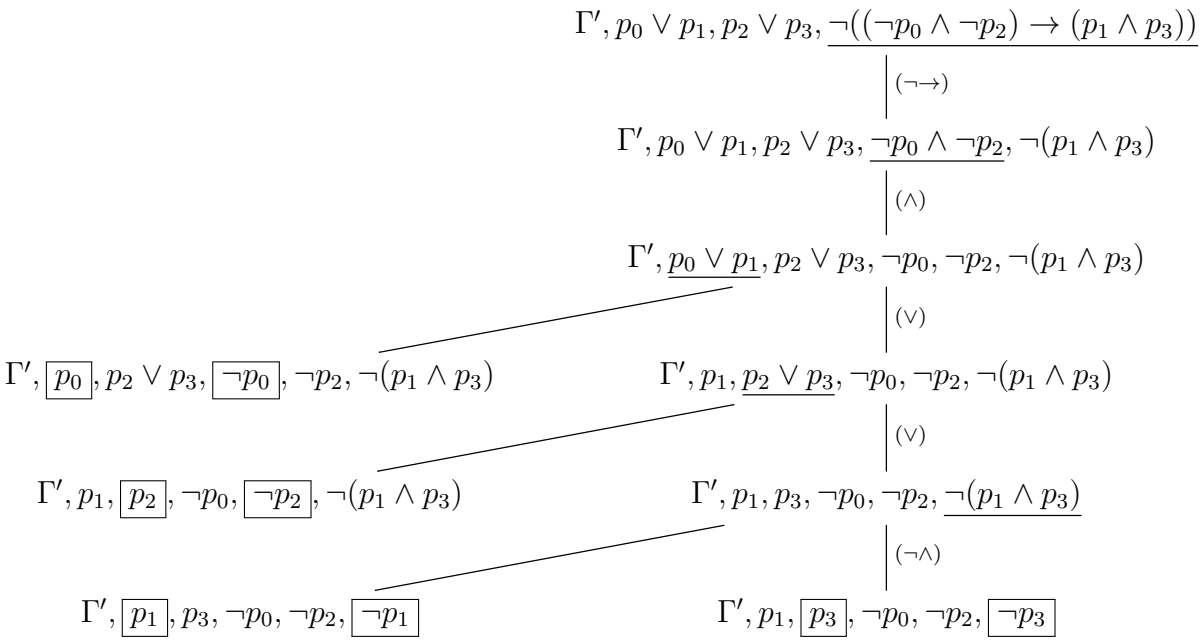


All leaves in this tableau are contradictory, so the initial judgement holds. ◁

It is again instructive to compare these examples with the semantic reasoning in the previous section: tableaux essentially organize the information in a structured way, making it easier to process and to avoid mistakes.

The case when Γ is infinite is not significantly different: we simply ignore the irrelevant formulas and proceed as before.

Example. Let $\Gamma = \{p_{2i} \vee p_{2i+1} \mid i \in \mathbb{N}\}$. We show that $\Gamma \vdash_S (\neg p_0 \wedge \neg p_2) \to (p_1 \wedge p_3)$; in the tableau below, we write Γ' for $\{p_{2i} \vee p_{2i+1} \mid i \geq 2\}$, so that $\Gamma = \Gamma' \cup \{p_0 \vee p_1, p_2 \vee p_3\}$.



Since all leaves of this tableau are contradictory, the initial judgement holds. ◁

This last example illustrates some of the options to minimize tableau size: when applying a binary rule, choose one such that at least one of the generated nodes is a contradiction (if

possible). Of course, sometimes one can not avoid to generate two branches that need to be expanded further, but often the size of the tableau can be kept under control by adequately choosing the rule to apply.

Exercise 12. Decide whether the following judgements hold.

(a) $\{p \vee q, p \wedge q\} \vdash_S p$ (c) $\{p \wedge q, \neg p\} \vdash_S r$ (e) $\{p \vee q, p \to r, q \to r\} \vdash_S r$

(b) $\{\neg p \to p\} \vdash_S p$ (d) $\{p\} \vdash_S q \to p$ (f) $\{p \to (q \wedge r), \neg p\} \vdash_S q \vee r$

In the cases where they do not hold, can you use a non-contradictory leaf to find a valuation that makes all hypotheses true and the conclusion false?

The previous examples and exercises already suggest that there is a deep connection between the judgements $\Gamma \models \varphi$ and $\Gamma \vdash_S \varphi$. Indeed, they are equivalent: if the tableaux calculus can prove that $\Gamma \vdash_S \varphi$, then $\Gamma \models \varphi$ (soundness), and if $\Gamma \models \varphi$ then the tableaux calculus can prove $\Gamma \vdash_S \varphi$ (completeness). The proofs of these properties both rely on the following lemma.

Lemma 4. Let Γ be a set of formulas, V be a valuation and r be a rule in the tableaux calculus for propositional logic. Then $V \models \Gamma$ iff $V \models \Delta$ for some Δ among the sets of formulas obtained by applying r to Γ.

Proof. The proof proceeds by showing the result for every rule. We detail two cases.

Consider rule $(\neg\neg)$. Then

$$\begin{aligned} V \models \Gamma, \neg\neg\varphi \text{ iff } & V \models \Gamma \text{ and } V \models \neg\neg\varphi \\ \text{iff } & V \models \Gamma \text{ and } V \not\models \neg\varphi \\ \text{iff } & V \models \Gamma \text{ and } V \models \varphi \end{aligned}$$

so $V \models \Gamma, \neg\neg\varphi$ iff $V \models \Gamma, \varphi$.

Consider rule $(\to)$. Then

$$\begin{aligned} V \models \Gamma, \varphi \to \psi \text{ iff } & V \models \Gamma \text{ and } V \models \varphi \to \psi \\ \text{iff } & V \models \Gamma \text{ and } \Big[V \not\models \varphi \text{ or } V \models \psi\Big] \\ \text{iff } & V \models \Gamma \text{ and } \Big[V \models \neg\varphi \text{ or } V \models \psi\Big] \\ \text{iff } & \Big[V \models \Gamma \text{ and } V \models \neg\varphi\Big] \text{ or } \Big[V \models \Gamma \text{ and } V \models \psi\Big] \end{aligned}$$

so $V \models \Gamma, \varphi \to \psi$ iff $V \models \Gamma, \neg\varphi$ or $V \models \Gamma, \psi$. □

Exercise 13. Prove the remaining cases of Lemma 4.

This result extends to tableaux by induction.

Lemma 5. Let Γ be a set of formulas, V be a valuation, and T be a tableau with root labeled by Γ. Then $V \models \Gamma$ iff $V \models \Delta$ for some Δ labeling a leaf of T.

Proof. By induction on the construction of T. If T consists of a single node, then the statement is trivial, since its only leaf is the root, which is labeled by Γ. Otherwise, T can be obtained from a tableau T' by applying a rule to one of the leafs of T'. Let Ψ be the set of formulas labeling that leaf.

By induction hypothesis, $V \models \Gamma$ iff $V \models \Delta$ for some Δ labeling a leaf of T'. There are three cases to consider. (1) Suppose that $V \models \Gamma$ and $V \models \Delta$ for some $\Delta \neq \Psi$ labeling a leaf of T'. Since Δ also labels a leaf of T, the thesis immediately follows. (2) Suppose that $V \models \Gamma$ and that the only set labeling a leaf of T' that is satisfied by V is precisely Ψ. By Lemma 4, $V \models \Delta$ for some Δ labeling one descendant of Ψ in T, which is labeling a leaf of T. (3) Suppose that $V \not\models \Gamma$. Then $V \not\models \Delta$ for all Δ labeling leaves of T'. Since any leaf of T that is not a descendant of Ψ is a leaf of T', this immediately establishes the thesis for all those leaves. If Δ labels a descendent of Ψ in T, then by Lemma 4 also $V \not\models \Delta$. □

Using this lemma, we can now prove the main results about this calculus.

Theorem 3 (Soundness)**.** Let Γ be a set of propositional formulas and φ be a formula. If $\Gamma \vdash_S \varphi$, then $\Gamma \models \varphi$.

Proof. Suppose that $\Gamma \vdash_S \varphi$. Then there is a tableau T with root labeled $\Gamma, \neg\varphi$ and such that all branches of T are finite and end in contradictory leaves. By the semantics of negation, no valuation can satisfy a contradictory leaf, hence $V \not\models \Gamma \cup \{\neg\varphi\}$ for all V by Lemma 5, i.e., $\Gamma \cup \{\neg\varphi\}$ is contradictory. Therefore $\Gamma \models \varphi$. □

The proof of completeness relies again on the fact that entailment in propositional logic is compact, which we will prove in a later section.

Theorem 4 (Completeness)**.** Suppose that $\Gamma \models \varphi$. Then $\Gamma \vdash_S \varphi$.

Proof. By compactness, there is a finite set $\Gamma_{\mathsf{fin}} \subseteq \Gamma$ such that $\Gamma_{\mathsf{fin}} \models \varphi$. Let T be a tableau with root labeled by $\Gamma_{\mathsf{fin}}, \neg\varphi$ such that no more rules can be applied.

Since the total number of connectives on all formulas labeling a node decreases by application of a rule, every branch of T is finite. If there is a non-contradictory leaf in T, then it consists solely of atomic formulas, and we can define a valuation V that satisfies all of them. By Lemma 5, $V \models \Gamma_{\mathsf{fin}} \cup \{\neg\varphi\}$, whence $\Gamma_{\mathsf{fin}} \not\models \varphi$, contradicting our choice of φ. Therefore all leaves of T must be contradictory. Adding all formulas in $\Gamma \setminus \Gamma_{\mathsf{fin}}$ to all labels of T shows that $\Gamma \vdash_S \varphi$. □

In practical implementations of semantic tableaux, it is common not to copy all formulas from one node to its descendants, but rather only to include the new formulas in the nodes that are created by application of rules. The disadvantage is that contradictions now must be checked along the branch from each leaf to the root. For small examples done by hand, the presentation chosen in these notes is simpler, albeit more verbose.

2.4 Sequent calculus

Sequent calculi are another important family of deductive systems that has many important applications. They allow very low-level fine-tuning of proofs, which makes them interesting for logics where there are constraints on when and how hypotheses may be used; and they

represent proofs of logical entailments in a way that has close connections to the functional programming paradigm.

Like tableaux calculi, sequent calculi are rule-based systems where derivations are trees build from repeatedly applying rules. Unlike tableaux calculi, they are not based on contradictions: they build a direct proof of a judgement applying rules that also model the semantics of the connectives in a different way. Thus, nodes in derivations represent judgements that must hold, and the interpretation of branching nodes is conjunctive rather than disjunctive – in other words, in order for the judgement at the root to hold, all judgments in all nodes (and in particular in all leaves) must be true.

Judgements in the sequent calculus are called sequents, and they generalize entailments.

Definition. A *sequent* is a pair of sequences of formulas, written as $\Gamma \vdash \Delta$.

The intended semantics of $\Gamma \vdash \Delta$ is implicative: if all formulas in Γ are true, then some formula in Δ must be true.

Definition. Let V be a valuation. We say that V satisfies sequent $\Gamma \vdash \Delta$, written $V \models \Gamma \vdash \Delta$, if $V \not\models \varphi$ for some $\varphi \in \Gamma$ or $V \models \varphi$ for some $\varphi \in \Delta$.

Rules in sequent calculi are typically classified in three categories: structural rules, logical rules, and special rules. Structural rules are used to manipulate sequents internally, without changing the formulas in them. Logical rules, by contrast, change formulas depending on their main connective and whether they occur on the left- or righthandside of a sequent. The special rules are axiom and cut rules, which are applied to terminate (a branch of) a proof or to introduce hypotheses. With the exception of the special rules, all rules appear in two variants – left and right, denoted by an 'L' or 'R' in their name.

Definition. A derivation of a sequent s in the sequent calculus for propositional logic is a tree with root s and such that every node is has descendents obtained by applying one of the following rules.

Structural rules:

$$\frac{\Gamma \vdash \Delta}{\Gamma, \varphi \vdash \Delta}\,\mathrm{WL} \qquad \frac{\Gamma, \varphi, \psi, \Gamma' \vdash \Delta}{\Gamma, \psi, \varphi, \Gamma' \vdash \Delta}\,\mathrm{PL} \qquad \frac{\Gamma, \varphi, \varphi \vdash \Delta}{\Gamma, \varphi \vdash \Delta}\,\mathrm{CL}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \varphi}\,\mathrm{WR} \qquad \frac{\Gamma \vdash \Delta, \varphi, \psi, \Delta'}{\Gamma \vdash \Delta, \psi, \varphi, \Delta'}\,\mathrm{PR} \qquad \frac{\Gamma \vdash \Delta, \varphi, \varphi}{\Gamma \vdash \Delta, \varphi}\,\mathrm{CR}$$

Logical rules:

$$\frac{\Gamma \vdash \Delta, \varphi}{\Gamma, \neg\varphi \vdash \Delta}\,\neg\mathrm{L} \qquad \frac{\Gamma \vdash \Delta, \varphi \qquad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \to \psi \vdash \Delta}\,\to\mathrm{L}$$

$$\frac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \Delta, \neg\varphi}\,\neg\mathrm{R} \qquad \frac{\Gamma, \varphi \vdash \Delta, \psi}{\Gamma \vdash \Delta, \varphi \to \psi}\,\to\mathrm{R}$$

$$\frac{\Gamma, \varphi \vdash \Delta \qquad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \vee \psi \vdash \Delta}\,\vee\mathrm{L} \qquad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta}\,\wedge\mathrm{L}_1 \qquad \frac{\Gamma, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta}\,\wedge\mathrm{L}_2$$

$$\frac{\Gamma \vdash \Delta, \varphi}{\Gamma \vdash \Delta, \varphi \vee \psi}\,\vee\mathrm{R}_1 \qquad \frac{\Gamma \vdash \Delta, \psi}{\Gamma \vdash \Delta, \varphi \vee \psi}\,\vee\mathrm{R}_2 \qquad \frac{\Gamma \vdash \Delta, \varphi \qquad \Gamma \vdash \Delta, \psi}{\Gamma \vdash \Delta, \varphi \wedge \psi}\,\wedge\mathrm{R}$$

Special rules:

$$\frac{}{\varphi \vdash \varphi}\,\mathrm{Ax} \qquad\qquad \frac{\Gamma \vdash \Delta, \varphi \qquad \varphi, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\,\mathrm{Cut}$$

We write $\vdash_G s$ whenever there exists a derivation of sequent s. In particular, we write $\Gamma \vdash_G \varphi$ for $\vdash_G (\Gamma \vdash \varphi)$ and $\vdash_G \varphi$ for $\emptyset \vdash_G \varphi$. Furthermore, we also write $\vdash_G \Gamma \vdash \Delta$ if there exist sequences Γ' and Δ' containing only elements from Γ and Δ, respectively, such that there exists a derivation of $\vdash_G \Gamma' \vdash \Delta'$.

The subscript G in the notation for derivations in the sequent calculus is a tribute to the logician Gerhard Gentzen, who first introduced these systems in the 1930s.

Unlike in tableaux calculi, trees in sequent calculi are written with the root at the bottom, and built upwards by applying rules – rather like trees in nature, but not as most commonly in Computer Science. This is the tradition in logic, with the notorious exception of tableaux calculi.

The different types of rules in the sequent calculus are used for different purposes. The intuition behind the logical rules is very similar to that for the rules of the tableaux calculus: each rule considers the semantics of a connective depending on whether it occurs on the left part of a sequent (where the formula should be false) or on the right (where the formula should be true). Thus, rule ($\to$R) can be read as "$\varphi \to \psi$ is true if φ is false or ψ is true" – corresponding to the semantics of the connective. Likewise, rule ($\to$L) reads "$\varphi \to \psi$ is false if φ is true and ψ is false" – and the way to express this conjunction is by generating two descendants, one for each premise.

Exercise 14. Go through the remaining logical rules and check that they do correspond to the semantics of the connectives, as exemplified for implication.

Sequent calculi operate on sequences of formulas, rather than sets (as is the case in tableaux calculi). Although for propositional logic this is not essential – we could have presented the sequent calculus using sets, and dispensing with the structural rules – it is a key feature of this family of calculi. An important consequence is that the logical rules, as they are stated, can only be applied when the relevant formula is the last one in its side of the sequent. This is where structural rules come into play. The *permutation* rules (PL) and (PR) are necessary to move formulas to the end position. Furthermore, duplicate formulas may arise, and the *contraction* rules (CL) and (CR) eliminate them. Finally, since the axiom rule requires exactly one formula on the left- and righthandside of the sequent, the *weakening* rules (WL) and (WR) allow additional formulas to be added to both sides at will.

We now illustrate the mechanism of proofs in this calculus.

Example. Our first example is one of the paradoxes of implication.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p \vdash p}\,\mathrm{Ax}}{\vdash p, \neg p}\,\neg\mathrm{R} \qquad \dfrac{}{p \vdash p}\,\mathrm{Ax}}{\neg p \to p \vdash p}\,\to\mathrm{L}}{\vdash (\neg p \to p) \to p}\,\to\mathrm{R}$$

In this derivation, the only formula in each sequent that is not a propositional symbol is always the last one in its side of the sequent. Therefore, we can always apply the logical rules directly, and no structural rules are needed. ◁

Example. Next, we prove that $\vdash_G p \to (q \to p)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p \vdash p}\,\mathrm{Ax}}{p, q \vdash p}\,\mathrm{WL}}{p \vdash q \to p}\,{\to}\mathrm{R}}{\vdash p \to (q \to p)}\,{\to}\mathrm{R}$$

Recall that the derivation is built bottom-up. At the next-to-last step (at the top), we used weakening to remove q from the lefthandside of the sequent in order to obtain an axiom. ◁

Example. We now show that $\{(p \vee q) \to \neg p\} \vdash_G \neg p$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p \vdash p}\,\mathrm{Ax}}{p \vdash (p \vee q)}\,{\vee}\mathrm{R}_1 \quad \dfrac{\dfrac{}{p \vdash p}\,\mathrm{Ax}}{p, \neg p \vdash}\,\neg\mathrm{L}}{p, (p \vee q) \to \neg p \vdash}\,{\to}\mathrm{L}}{(p \vee q) \to \neg p, p \vdash}\,\mathrm{PL}}{(p \vee q) \to \neg p \vdash \neg p}\,\neg\mathrm{R}$$

This derivation illustrates the use of a permutation rule to move the formula we are interested in simplifying ($(p \vee q) \to \neg p$) to the correct place. It also shows that the sequence on the righthandside of the sequent may also be empty: this expresses that the set of formulas on the lefthandside is contradictory. ◁

Example. The next derivation shows that $\{p \vee q, p \to r, q \to s\} \vdash_G r \vee s$. Building this derivation requires a bit of care, because of the disjunction on the righthandside: either we use contraction to obtain both r and s on the resulting sequent, or we need to delay processing that disjunction until we know which of r or s is needed to obtain an axiom in each branch.

We first show a derivation using this last approach.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p \vdash p}\,\mathrm{Ax}}{p \vdash p, r}\,\mathrm{WR}}{p \vdash r, p}\,\mathrm{PR} \quad \dfrac{\dfrac{\dfrac{}{r \vdash r}\,\mathrm{Ax}}{r, p \vdash r}\,\mathrm{WL}}{p, r \vdash r}\,\mathrm{PL}}{p, p \to r \vdash r}\,{\to}\mathrm{R}}{p, p \to r \vdash r \vee s}\,{\vee}\mathrm{R}_1}{p, p \to r \vdash r \vee s, q}\,\mathrm{WR}}{p \to r, p \vdash r \vee s, q}\,\mathrm{PL} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{q \vdash q}\,\mathrm{Ax}}{q \vdash q, r \vee s}\,\mathrm{WR}}{q \vdash r \vee s, q}\,\mathrm{PR}}{q, p \to r \vdash r \vee s, q}\,\mathrm{WL}}{p \to r, q \vdash r \vee s, q}\,\mathrm{PL}}{p \to r, p \vee q \vdash r \vee s, q}\,{\vee}\mathrm{L}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{s \vdash s}\,\mathrm{Ax}}{s, p \vee q \vdash s}\,\mathrm{WL}}{p \vee q, s \vdash s}\,\mathrm{PL}}{p \vee q, s, p \to r \vdash s}\,\mathrm{WL}}{p \vee q, p \to r, s \vdash s}\,\mathrm{PL}}{p \vee q, p \to r, s \vdash r \vee s}\,{\vee}\mathrm{R}_2$$

$$\dfrac{p \vee q, p \to r \vdash r \vee s, q}{p \vee q, p \to r, q \to s \vdash r \vee s}\,\mathrm{PL} \qquad \dfrac{p \vee q, p \to r, s \vdash r \vee s}{p \vee q, p \to r, q \to s \vdash r \vee s}\,{\to}\mathrm{L}$$

In the leftmost branch, we used weakening to remove the unnecessary hypothesis q.

Alternatively, we could use contraction to obtain both r and s on the righthandside.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{p \vee q, p \to r, q \to s \vdash r, s}}{p \vee q, p \to r, q \to s \vdash r, r \vee s}\,{\vee}\mathrm{R}_2}{p \vee q, p \to r, q \to s \vdash r \vee s, r}\,\mathrm{PR}}{p \vee q, p \to r, q \to s \vdash r \vee s, r \vee s}\,{\vee}\mathrm{R}_1}{p \vee q, p \to r, q \to s \vdash r \vee s}\,\mathrm{CR}$$

This derivation could then be continued by applying rules to the formulas on the left. ◁

Exercise 15. Finish the last derivation in the previous example.

Example. We present an alternative proof of the same sequent, using the Cut rule. The idea is: from the hypotheses $p \vee q$ and $p \to r$, we can derive $q \vee r$ – which is "closer" to the result.

The proof thus looks like

$$\dfrac{\dfrac{\mathcal{D}_1}{p \vee q, p \to r \vdash q \vee r} \qquad \dfrac{\mathcal{D}_2}{q \vee r, q \to s \vdash r \vee s}}{p \vee q, p \to r, q \to s \vdash r \vee s}\,\mathrm{Cut}$$

where $\mathcal{D}_1$ is the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p \vdash p}\,\mathrm{Ax}}{p \vdash p, q \vee r}\,\mathrm{WR}}{p \vdash q \vee r, p}\,\mathrm{PR} \qquad \dfrac{\dfrac{\dfrac{}{q \vdash q}\,\mathrm{Ax}}{q \vdash q \vee r}\,\vee\mathrm{R}_1}{q \vdash q \vee r, p}\,\mathrm{WR}}{p \vee q \vdash q \vee r, p}\,\vee\mathrm{L} \qquad \dfrac{\dfrac{\dfrac{}{r \vdash r}\,\mathrm{Ax}}{r, p \vee q \vdash r}\,\mathrm{WL}}{p \vee q, r \vdash r}\,\mathrm{PL}}{p \vee q, r \vdash q \vee r}\,\vee\mathrm{R}_2}{p \vee q, p \to r \vdash q \vee r}\,{\to}\mathrm{L}$$

and $\mathcal{D}_2$ is

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q \vdash q}\,\mathrm{Ax}}{q \vdash q, r \vee s}\,\mathrm{WR}}{q \vdash r \vee s, q}\,\mathrm{PR} \qquad \dfrac{\dfrac{\dfrac{}{r \vdash r}\,\mathrm{Ax}}{r \vdash r \vee s}\,\vee\mathrm{R}_1}{r \vdash r \vee s, q}\,\mathrm{WR}}{q \vee r \vdash r \vee s, q}\,\vee\mathrm{L} \qquad \dfrac{\dfrac{\dfrac{}{s \vdash s}\,\mathrm{Ax}}{s, q \vee r \vdash s}\,\mathrm{WL}}{q \vee r, s \vdash s}\,\mathrm{PL}}{q \vee r, s \vdash r \vee s}\,\vee\mathrm{R}_2}{q \vee r, q \to s \vdash r \vee s}\,{\to}\mathrm{L}$$

This example illustrates how (Cut) can be used to identify intermediate formulas in the derivation, allowing the proof to be split in steps. Note that the resulting proof is significantly simpler than the previous ones – in the sense that its branches are shorter. ◁

These examples show that, when the number of formulas in the sequent grows, structural rules start making up most of the derivation. In order to simplify proofs, it is customary to add extra rules to the sequent calculus that do not change the set of derivable formulas. We call such rules *admissible*; to prove that a rule is admissible, it suffices to show a schematic derivation for it – then this schematic derivation can be inserted everywhere that the admissible rule is used.

Lemma 6. The following rules are admissible in the sequent calculus for classical propositional logic.

$$\dfrac{\Gamma \vdash \Delta, \varphi, \psi}{\Gamma \vdash \Delta, \varphi \vee \psi}\,\vee\mathrm{R} \qquad \dfrac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta}\,\wedge\mathrm{L}$$

Proof. The following derivations show that these rules are admissible.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \Delta, \varphi, \psi}}{\Gamma \vdash \Delta, \varphi, \varphi \vee \psi}\vee R_2}{\Gamma \vdash \Delta, \varphi \vee \psi, \varphi}\mathrm{PR}}{\Gamma \vdash \Delta, \varphi \vee \psi, \varphi \vee \psi}\vee R_1}{\Gamma \vdash \Delta, \varphi \vee \psi}\mathrm{CR} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, \varphi, \psi \vdash \Delta}}{\Gamma, \varphi, \varphi \wedge \psi \vdash \Delta}\wedge L_2}{\Gamma, \varphi \wedge \psi, \varphi \vdash \Delta}\mathrm{PR}}{\Gamma, \varphi \wedge \psi, \varphi \wedge \psi \vdash \Delta}\wedge L_1}{\Gamma, \varphi \wedge \psi \vdash \Delta}\mathrm{CL}$$

The derivation for rule ($\vee$R) was already used in a previous example; the derivation for rule ($\wedge$L) is similar. □

As for structural rules, we can also use some powerful rules.

Lemma 7. The following rule is admissible in the sequent calculus for classical propositional logic.

$$\dfrac{\Gamma \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\mathrm{W}$$

Proof. Intuitively, an application of (W) can be replaced by a chain of applications of (WL) and (WR), where the formulas from Γ' and Δ' are added one at a time. In order to give a formal proof, we start by showing that the following two rules are admissible.

$$\dfrac{\Gamma \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta}\mathrm{WL'} \qquad \dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \Delta'}\mathrm{WR'}$$

We prove admissibility of (WL') by induction on the size of Γ'. If Γ' is empty, then this is straightforward. Assume that Γ' is Γ'', φ, and that $\vdash_G \Gamma \vdash \Delta$. By induction hypothesis, $\vdash_G \Gamma, \Gamma'' \vdash \Delta$; applying rule (WL) we conclude that $\vdash_G \Gamma, \Gamma'', \varphi \vdash \Delta$ – which is the conclusion of rule (WL').

The proof of admissibility of (WR') is similar.

Using these rules, we can construct the derivation

$$\dfrac{\dfrac{\Gamma \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta}\mathrm{WL'}}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\mathrm{WR'}$$

that shows that (W) is admissible. □

Exercise 16. Consider the following rule.

$$\dfrac{\Gamma \vdash \Delta}{\Gamma', \Gamma \vdash \Delta', \Delta}\mathrm{W'}$$

(a) Show that this rule is admissible in the sequent calculus for classical propositional logic. Note that, on top of the strategy for the previous proof, you need to apply the permutation rules in order to move the newly added formulas to the right place in the sequent.

(b) Can you show directly (i.e., by constructing an explicit derivation without any induction proof) that this rule is admissible?

Hint. The conclusion of rule (Cut) is identical to the conclusion of rule W'.

Lemma 8. The following rule is admissible in the sequent calculus for classical propositional logic.

$$\frac{\Gamma \cap \Delta \neq \emptyset}{\Gamma \vdash \Delta}\,\text{Ax'}$$

Proof. If $\Gamma \cap \Delta \neq \emptyset$, then we can write Γ as $\Gamma_1, \varphi, \Gamma_2$ and Δ as $\Delta_1, \varphi, \Delta_2$. We can then apply rules (W) and (W') to obtain the derivation

$$\dfrac{\dfrac{\dfrac{}{\varphi \vdash \varphi}\,\text{Ax}}{\varphi, \Gamma_2 \vdash \varphi, \Delta_2}\,\text{W}}{\Gamma_1, \varphi, \Gamma_2 \vdash \Delta_1, \varphi, \Delta_2}\,\text{W'}$$

that shows that this rule is admissible. □

Example. We show how the three proofs given earlier of $q \vee r, q \to s \vdash r \vee s$ can be simplified using these derived rules. The first proof becomes

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p \vdash r, p}\,\text{Ax'} \quad \dfrac{}{p, r \vdash r}\,\text{Ax'}}{p, p \to r \vdash r}\,{\to}\text{R}}{p, p \to r \vdash r \vee s}\,{\vee}\text{R}_1}{p, p \to r \vdash r \vee s, q}\,\text{WR} \quad \dfrac{}{q, p \to r \vdash r \vee s, q}\,\text{Ax'}}{p \vee q, p \to r \vdash r \vee s, q}\,{\vee}\text{L} \quad \dfrac{\dfrac{}{p \vee q, p \to r, s \vdash s}\,\text{Ax'}}{p \vee q, p \to r, s \vdash r \vee s}\,{\vee}\text{R}_2}{p \vee q, p \to r, q \to s \vdash r \vee s}\,{\to}\text{L}$$

The beginning of the second proof becomes

$$\dfrac{\vdots}{p \vee q, p \to r, q \to s \vdash r, s}\,{\vee}\text{R}$$
$$p \vee q, p \to r, q \to s \vdash r \vee s$$

and the proof using (Cut) can be simplified to

$$\dfrac{\dfrac{\mathcal{D}_1'}{p \vee q, p \to r \vdash q \vee r} \quad \dfrac{\mathcal{D}_2'}{q \vee r, q \to s \vdash r \vee s}}{p \vee q, p \to r, q \to s \vdash r \vee s}\,\text{Cut}$$

where $\mathcal{D}_1'$ is the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q \vdash q}\,\text{Ax}}{q \vdash q \vee r}\,{\vee}\text{R}_1}{q \vdash q \vee r, p}\,\text{WR} \quad \dfrac{}{p \vdash q \vee r, p}\,\text{Ax'}}{p \vee q \vdash q \vee r, p}\,{\vee}\text{L} \quad \dfrac{\dfrac{}{p \vee q, r \vdash r}\,\text{Ax'}}{p \vee q, r \vdash q \vee r}\,{\vee}\text{R}_2}{p \vee q, p \to r \vdash q \vee r}\,{\to}\text{L}$$

and $\mathcal{D}_2'$ is

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{r \vdash r}\,\text{Ax}}{r \vdash r \vee s}\,{\vee}\text{R}_1}{r \vdash r \vee s, q}\,\text{WR} \quad \dfrac{}{q \vdash r \vee s, q}\,\text{Ax'}}{q \vee r \vdash r \vee s, q}\,{\vee}\text{L} \quad \dfrac{\dfrac{}{q \vee r, s \vdash s}\,\text{Ax'}}{q \vee r, s \vdash r \vee s}\,{\vee}\text{R}_2}{q \vee r, q \to s \vdash r \vee s}\,{\to}\text{L}$$

Additional derived rules could be used to simplify proofs further. ◁

Exercise 17. Prove that all logical rules can be generalized to be applicable to any formula in the sequent. For example, for $\neg$ we would obtain the following two rules.

$$\frac{\Gamma, \Gamma' \vdash \Delta, \varphi, \Delta'}{\Gamma, \neg\varphi, \Gamma' \vdash \Delta, \Delta'} \neg\mathrm{L} \qquad \frac{\Gamma, \varphi, \Gamma' \vdash \Delta, \Delta'}{\Gamma, \Gamma' \vdash \Delta, \neg\varphi, \Delta'} \neg\mathrm{R}$$

Can you use this rules to simplify the previous proofs further?

Exercise 18. Prove that the following rule is admissible in the sequent calculus for classical propositional logic:

$$\frac{\Gamma \vdash \Delta}{\Gamma' \vdash \Delta'} \mathrm{P}$$

where Γ' and Δ' are permutations of Γ and Δ, respectively (that is, each sequent contain exactly the same formulas on each side, but in a different order).

Soundness of the sequent calculus for classical propositional logic is proved by the general technique for deduction systems based on rules.

Theorem 5 (Soundness)**.** Let $\Gamma \vdash \Delta$ be a sequent. If $\vdash_G \Gamma \vdash \Delta$, then $\models \Gamma \vdash \Delta$.

Proof. By induction on the derivation for $\vdash_G \Gamma \vdash \Delta$. For each rule, we need to show that: if all valuations satisfy all the premises, then all valuations satisfy the conclusion. Note that, for the structural rules, this is trivially the case.

We detail two cases of logical rules.

Consider rule ($\neg$R), and assume that all valuations satisfy $\Gamma, \varphi \vdash \Delta$. Let V be such a valuation. If $V \models \delta$ for some $\delta \in \Delta$, then $V \models \delta$ for some $\delta \in \Delta, \neg\varphi$, so $V \models \Gamma \vdash \Delta, \neg\varphi$. Otherwise, if $V \not\models \gamma$ for some $\gamma \in \Gamma, \varphi$, then there are two possibilities. If $\gamma \in \Gamma$, then $V \models \Gamma \vdash \Delta, \neg\varphi$; else, γ is φ, and $V \models \neg\varphi$, whence again $V \models \Gamma \vdash \Delta, \neg\varphi$.

Consider now rule $\to L$ and assume all valuations satisfy *both* $\Gamma \vdash \Delta, \varphi$ and $\Gamma, \psi \vdash \Delta$. Let V be a valuation; then either (1a) $V \not\models \gamma$ for some $\gamma \in \Gamma$ or (1b) $V \models \delta$ for some $\delta \in \Delta, \varphi$, and (2a) $V \not\models \gamma$ for some $\gamma \in \Gamma, \psi$ or (2b) $V \models \delta$ for some $\delta \in \Delta$. If (1a) or (2b) hold, then immediately $V \models \Gamma, \varphi \to \psi \vdash \Delta$; so assume that (1a) and (2b) do not hold. Then (1b) and (2a) must both hold, and we conclude that $V \models \varphi$ and $V \not\models \psi$, whence $V \not\models \varphi \to \psi$ and therefore $V \models \Gamma, \varphi \to \psi \vdash \Delta$.

As for special rules, we consider the case of (Ax). If $\varphi \vdash \varphi$ is an instance of this rule and V is a valuation, then clearly either $V \models \varphi$ or $V \not\models \varphi$; therefore, $V \models \varphi \vdash \varphi$. □

Exercise 19. Prove the remaining cases of the Soundness Theorem for sequent calculus. Do not forget the Cut rule.

The completeness proof is more sophisticated: it simulates building all possible derivations for a given sequent in parallel to conclude that a valid proof must exist. We detail this proof only for the $\{\neg, \to\}$-fragment of propositional logic; the general case is a straightforward generalization.

Theorem 6 (Completeness)**.** Let $\Gamma \vdash \Delta$ be a sequent in the $\{\neg, \to\}$-fragment of propositional logic. If $\models \Gamma \vdash \Delta$, then $\vdash_G \Gamma \vdash \Delta$.

Proof. First assume that Γ, Δ are both finite. We build a *reduction tree* for $\Gamma \vdash \Delta$ as follows: the root is labeled with $\Gamma \vdash \Delta$ and marked unfinished; we iteratively repeat the following steps.

1. To every unfinished leaf $\Gamma' \vdash \Delta'$, add a descendant $\Gamma' \vdash \Delta', \varphi_1, \ldots, \varphi_k$, where $\neg\varphi_1, \ldots, \neg\varphi_k$ are all the formulas of the form $\neg\varphi$ in Γ' that have not been processed earlier in this branch.

2. To every unfinished leaf $\Gamma' \vdash \Delta'$, add a descendant $\Gamma', \varphi_1, \ldots, \varphi_k \vdash \Delta'$, where $\neg\varphi_1, \ldots, \neg\varphi_k$ are all the formulas of the form $\neg\varphi$ in Δ' that have not been processed earlier in this branch.

3. To every unfinished leaf $\Gamma' \vdash \Delta'$, add 2^n descendants of the form $\Gamma', \psi_{i_1}, \ldots, \psi_{i_m} \vdash \Delta', \varphi_{j_1}, \ldots, \varphi_{j_n}$, where $i_1, \ldots, i_m, j_1, \ldots, j_n$ is a permutation of the numbers between 1 and k, and $\varphi_1 \to \psi_1, \ldots, \varphi_k \to \psi_k$ are all the formulas of the form $\varphi \to \psi$ in Γ' that have not been processed earlier in this branch.

4. To every unfinished leaf $\Gamma' \vdash \Delta'$, add a descendant $\Gamma', \varphi_1, \ldots, \varphi_k \vdash \Delta', \psi_1, \ldots, \psi_k$, where $\varphi_1 \to \psi_1, \ldots, \varphi_k \to \psi_k$ are all the formulas of the form $\varphi \to \psi$ in Δ' that have not been processed earlier in this branch.

5. Mark every unfinished leaf $\Gamma' \vdash \Delta'$ where $\Gamma' \cap \Delta' \neq \emptyset$ as finished.

There are two possibilities. If the reduction tree for $\Gamma \vdash \Delta$ is finite, then it can be transformed into a derivation of $\Gamma \vdash \Delta$, since the node added in each of the earlier steps is always derivable using the rules of the sequent calculus.

Otherwise, there is an infinite branch in this tree. Define a valuation V by $V(p) = \top$ if p occurs on the left of some sequent on this branch, and $V(p) = \bot$ if p appears on the right of some sequent on this branch. By construction of the tree, (1) V is a well-defined valuation, (2) $V \models \gamma$ for every $\gamma \in \Gamma$ and (3) $V \not\models \delta$ for every $\delta \in \Delta$. Therefore $V \not\models \Gamma \vdash \Delta$, which is absurd, since we assumed that $\models \Gamma \vdash \Delta$.

If Γ or Δ are infinite, we invoke compactness of propositional logic to find finite $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ such that $\models \Gamma' \vdash \Delta'$. Then we can apply the previous construction to show that $\vdash_G \Gamma' \vdash \Delta'$, whence $\vdash_G \Gamma \vdash \Delta$. □

Exercise 20. Prove that properties (1), (2) and (3) in the previous proof indeed hold.

Exercise 21. Extend the Completeness Proof for sequent calculus to the full set of propositional connectives.

Rule (Cut) plays a special role in the sequent calculus. Indeed, the calculus without this rule enjoys the *subformula property*: every formula in any sequent in a derivation of $\Gamma \vdash \Delta$ is a subformula of a formula in either Γ and Δ. This property is extremely useful for automated proof search; however, as illustrated in an earlier example, (Cut) can introduce new formulas that break this property. Therefore, an extremely important property in sequent calculi is *Cut*

elimination: can this rule be removed without changing the set of derivable formulas? For propositional logic, this is trivially the case.

Theorem 7 (Cut-elimination)**.** Let $\Gamma \vdash \Delta$ be a sequent. If $\vdash_G \Gamma \vdash \Delta$, then there is a proof of $\Gamma \vdash \Delta$ that does not use rule Cut.

Proof. The construction in the proof of the Completeness Theorem yields a proof that does not use Cut. □

However, this result does not imply that we should simply disregard rule (Cut): a direct proof of this result (by induction on the derivation of $\Gamma \vdash \Delta$, showing how any application of (Cut) can systematically be replaced by application of other rules) shows that (Cut) can exponentially decrease the size of a proof. This makes this rule useful for efficiency purposes; the formulas introduced by (Cut) can either be discovered automatically by means of heuristics, or be suggested by a human.

2.5 Hilbert calculus

The next deductive system for propositional logic that we will discuss is the Hilbert calculus. Hilbert calculi are a different type of rule-based deductive systems, and they are some of the oldest deductive systems in use: they were originally developed as an attempt to formalize the traditional way of doing mathematics.

Hilbert calculi are forward-oriented calculus, where one starts from premises and derives more and more complex conclusions – as opposed to the backwards-oriented reasoning in tableaux and sequent calculi, where one starts with the conclusions and simplifies them until one reaches a contradiction.

These calculi are also based on direct proofs: they show that φ can be derived from Γ by applying rules to formulas in Γ until φ is produced, instead of starting from $\Gamma \cup \{\neg\varphi\}$ and attempting to reach a contradiction. This makes them applicable in more families of logics, as they do not embody the principles of classical reasoning – in particular, they can be used in intuitionistic logics, where negation is weaker.

In the area of automated reasoning, Hilbert calculi are not very useful, as finding proofs requires a lot of ingenuity and creativity – which is not easy to automate. However, their simple structure makes them extremely useful both for automatically checking proof validity and for proving theoretical properties of logics.

2.5.1 Axioms, rules, and soundness

A Hilbert calculus for a logic is specified by a set of schematic axioms and a set of inference rules. The axioms are schematic in the sense that they typically specify a structure that can be instantiated in different ways, rather than being a concrete formula. The Hilbert calculus for classical propositional logic includes the following three schematic axioms.

$$\varphi \to (\psi \to \varphi) \tag{Ax.1}$$
$$(\varphi \to (\psi \to \gamma)) \to ((\varphi \to \psi) \to (\varphi \to \gamma)) \tag{Ax.2}$$
$$(\neg\psi \to \neg\varphi) \to (\varphi \to \psi) \tag{Ax.3}$$

In these axioms, the formulas φ, ψ and γ can be instantiated by any propositional formula.

The only inference rule for propositional logic is

$$\text{from } \varphi \text{ and } \varphi \to \psi \text{ infer } \psi \qquad \text{(MP)}$$

"MP" stands for *Modus Ponens*, the classical name for the reasoning principle embodied in this rule.

Definition. A *derivation* of φ from Γ in the Hilbert calculus for propositional logic is a sequence of formulas $\varphi_1, \ldots, \varphi_n$ such that $\varphi_n = \varphi$ and every φ_i satisfies one of the following conditions:

- $\varphi_i \in \Gamma$;
- φ_i is an instance of an axiom;
- φ_i is obtained from φ_j and φ_k by application of (MP), with $j, k < i$.

We write $\Gamma \vdash_L \varphi$ to denote that there is a derivation of φ from Γ.

As usual, we omit Γ when it is empty.

It is customary to number the steps in derivations and to include *justifications* for each step: an indication of why the formula in that step is valid.

Example. The following derivation shows that $\vdash_L p \to p$.

1. $p \to ((p \to p) \to p)$	(Ax.1)
2. $(p \to ((p \to p) \to p)) \to ((p \to (p \to p)) \to (p \to p))$	(Ax.2)
3. $(p \to (p \to p)) \to (p \to p)$	MP(2,1)
4. $p \to (p \to p)$	(Ax.1)
5. $p \to p$	MP(3,4)

In the applications of (MP), it is customary to write the step number of the implication in the first place, and the step number of the antecedent in the second place.

In order to understand how to come up with this very unintuitive derivation, we observe that the only way to obtain $p \to p$ is by means of (MP), since $p \to p$ is not an instance of an axiom, and there are no hypotheses. Thinking backwards, we can try to match $p \to p$ with the conclusion of some axiom whose premise we can prove. Axiom (Ax.3) does not help here, since its premise would be $\neg p \to \neg p$, which is just as hard to prove. Axiom (Ax.1) would give $p \to (p \to p)$, and we would be left with proving p – which intuitively should not be possible, since it is not a valid formula.

This suggests that we use (Ax.2). In order for its last implication to be $p \to p$, we need to take φ and γ both to be p; the formula then will be of the form $(p \to (\psi \to p)) \to ((p \to \psi) \to (p \to p))$.

Now we are left with choosing ψ such that both $p \to (\psi \to p)$ and $p \to \psi$ are easy to prove. The first of these formulas is an instance of (Ax.1) regardless of ψ; the second is an instance of (Ax.1) as long as ψ has the form $\theta \to p$ for some θ. Choosing θ to be p yields the derivation above. ◁

This derivation illustrates the characteristics of the Hilbert calculus that we discussed earlier. The derivation of $p \to p$ is a direct derivation (not by contradiction) that is built up

from axioms by application of inference rules. It also shows how the Hilbert calculus is non-mechanical: every step of the derivation requires some creativity, and it is not always obvious how to build it.

Example. Let us now prove that $\{p \to q, q \to r\} \vdash_L p \to r$. We start by trying to understand what a derivation could look like before presenting it.

In order to prove $p \to r$, we again need to end with an application of (MP), so we start by looking for an instance of an axiom whose conclusion is $p \to r$. Again, (Ax.3) does not seem promising, since there are no negations in the hypotheses, while (Ax.1) requires us to prove r, which does not sound reasonable. So we are again left with an instance of (Ax.2), which should have the form $(p \to (\psi \to r)) \to ((p \to \psi) \to (p \to r))$.

In order to use this axiom, we can take ψ to be q: the antecendent of the second implication then becomes $p \to q$, which is a hypothesis. We are left with proving $p \to (q \to r)$; but since we have $q \to r$ as a hypothesis, we can use (Ax.1) in the form $(q \to r) \to (p \to (q \to r))$.

We can then connect these ideas into the following derivation.

1. $q \to r$	(Hyp)
2. $(q \to r) \to (p \to (q \to r))$	(Ax.1)
3. $p \to (q \to r)$	MP(2,1)
4. $p \to q$	(Hyp)
5. $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$	(Ax.2)
6. $(p \to q) \to (p \to r)$	MP(5,3)
7. $p \to r$	MP(6,4)

This derivation formalizes the informal ideas above in a forward-oriented manner. Steps 1 to 3 derive $p \to (q \to r)$ from (Ax.1) and the hypothesis $q \to r$, and steps 4 to 7 apply (Ax.2) to this conclusion and the hypothesis $p \to q$ in order to yield the formula we were looking for. $\triangleleft$

Exercise 22. Show that the following judgements hold.

(a) $\vdash_L p \to (q \to (p \to q))$

(b) $\{p\} \vdash_L (p \to q) \to q$

(c) $\{r \to p, r\} \vdash_L (q \to r) \to (q \to p)$

(d) $\{p \to (p \to q)\} \vdash_L p \to q$

(e) $\{p \to q, q \to r, r \to s\} \vdash_L p \to s$

It should not come as a surprise that this calculus is also sound.

Lemma 9. All instances of the axioms of the Hilbert calculus are valid formulas.

Proof. This is a simple exercise in semantics of propositional logic.

For (Ax.1), we have that $V \not\models \varphi \to (\psi \to \varphi)$ iff $V \models \varphi$ and $V \not\models \psi \to \varphi$ iff $V \models \varphi$ and $V \models \psi$ and $V \not\models \varphi$. Since it is impossible for V both to satisfy and not to satisfy φ, we conclude that no V can falsify any instance of this axiom. Therefore all instances of $(Ax.1)$ are valid.

The proof for (Ax.2) is similar.

$$\begin{aligned} V \not\models (\varphi \to (\psi \to \gamma)) \to ((\varphi \to \psi) \to (\varphi \to \gamma)) & \\ \text{iff } V \models \varphi \to (\psi \to \gamma) \text{ and } V \not\models (\varphi \to \psi) \to (\varphi \to \gamma) & \\ \text{iff } V \models \varphi \to (\psi \to \gamma) \text{ and } V \models \varphi \to \psi \text{ and } V \not\models \varphi \to \gamma & \\ \text{iff } \underbrace{V \models \varphi \to (\psi \to \gamma)}_{(1)} \text{ and } \underbrace{V \models \varphi \to \psi}_{(2)} \text{ and } \underbrace{V \models \varphi}_{(3)} \text{ and } \underbrace{V \not\models \gamma}_{(4)} & \end{aligned}$$

From (3) and (2) we conclude that $V \models \psi$. From (3) and (1) we conclude that $V \models \psi \to \gamma$, whence also $V \models \gamma$ – contradicting (4). Therefore no valuation can falsify any instance of (Ax.2).

Finally, for (Ax.3) we have that $V \not\models (\neg\psi \to \neg\varphi) \to (\varphi \to \psi)$ iff $V \models \neg\psi \to \neg\varphi$ and $V \not\models \varphi \to \psi$ iff $V \models \neg\psi \to \neg\varphi$ and $V \models \varphi$ and $V \not\models \psi$. But if $V \not\models \psi$ then $V \models \neg\psi$, from which we conclude that also $V \models \neg\varphi$, contradicting $V \models \varphi$. Therefore yet again no valuation can falsify an instance of (Ax.3). □

Theorem 8 (Soundness)**.** If $\Gamma \vdash_L \varphi$, then $\Gamma \models \varphi$.

Proof. By induction on the proof of $\Gamma \vdash_L \varphi$. If $\varphi \in \Gamma$, the thesis is trivial. If φ is an instance of an axiom, then the result follows from the previous lemma.

If φ is obtained by MP from ψ and $\psi \to \varphi$, then by induction hypothesis $\Gamma \models \psi$ and $\Gamma \models \psi \to \varphi$. Pick V such that $V \models \Gamma$. Then $V \models \psi$ and $V \models \psi \to \varphi$, whence $V \models \varphi$. □

2.5.2 Metatheorems

In a later section, we will show that this calculus is also complete. Before that, however, we prove and discuss some useful properties that make it easier to find derivations for particular formulas – and which will be instrumental in some steps of the completeness proof.

The first result is useful for reusing proofs of previous formulas.

Theorem 9 (Hypothetical Reasoning)**.** Let Γ be a set of propositional formulas, and φ and ψ be propositional formulas. If $\Gamma \vdash_L \psi$ and $\Gamma \cup \{\psi\} \vdash_L \varphi$, then $\Gamma \vdash_L \varphi$.

Proof. Suppose that $\mathcal{D}_\psi$ is a derivation of $\Gamma \vdash_L \psi$ and that $\mathcal{D}_\varphi$ is a derivation of $\Gamma \cup \{\psi\} \vdash_L \varphi$. From these we can construct a derivation showing that $\Gamma \vdash_L \varphi$ as follows: at every step of $\mathcal{D}_\varphi$ where ψ occurs, replace it with the derivation $\mathcal{D}_\psi$.

We now show that the resulting derivation is a valid derivation. First, we observe that the only steps in $\mathcal{D}_\varphi$ that possibly were not valid in constructing a derivation from Γ were those where ψ occurred (since ψ is no longer a hypothesis), and these have been replaced by valid derivations from Γ that also end with ψ. Furthermore, the derivation thus constructed still ends with φ, so it proves that $\Gamma \vdash_L \varphi$. □

Note that justifications are not a part of the formal definition of derivation. If they were, the reference to previous steps in applications of (MP) would need to be updated accordingly.

This theorem allows us to use any formulas previously proved as lemmas. It also gives us the possibility of splitting a proof into several smaller proofs, to make them more manageable.

Example. Recall that we already established that $\vdash_L p \to p$. Using this fact, we can construct the following derivation

1. $p \to p$	Lemma
2. $(p \to p) \to (q \to (p \to p))$	(Ax.1)
3. $q \to (p \to p)$	MP(2,1)

showing that also $\vdash_L q \to (p \to p)$. ◁

Often, this result is used in combination with the next result.

Theorem 10 (Replacement)**.** Let Γ be a set of formulas, $q_1, \ldots, q_k$ be propositional symbols, and $\psi, \varphi_1, \ldots, \varphi_k$ be propositional formulas.

Denote by $\psi[q_1/\varphi_1, \ldots, q_k/\varphi_k]$ the result of simultaneously replacing every occurrence of q_i in ψ by φ_i, and by $\Gamma[q_1/\varphi_1, \ldots, q_k/\varphi_k]$ the result of applying the same transformation to every formula in Γ.

If $\Gamma \vdash_L \psi$, then $\Gamma[q_1/\varphi_1, \ldots, q_k/\varphi_k] \vdash_L \psi[q_1/\varphi_1, \ldots, q_k/\varphi_k]$.

Proof. By induction on the derivation of $\Gamma \vdash_L \psi$.

Since axioms are schematic, if ψ is an instance of an axiom, then so is $\psi[q_1/\varphi_1, \ldots, q_k/\varphi_k]$.

If $\psi \in \Gamma$, then by construction also $\psi[q_1/\varphi_1, \ldots, q_k/\varphi_k] \in \Gamma[q_1/\varphi_1, \ldots, q_k/\varphi_k]$.

Finally, if ψ is obtained from γ and $\gamma \to \psi$ by application of (MP), then $\psi[q_1/\varphi_1, \ldots, q_k/\varphi_k]$ can likewise be obtained by applying (MP) to $\gamma[q_1/\varphi_1, \ldots, q_k/\varphi_k]$ and $(\gamma \to \psi)[q_1/\varphi_1, \ldots, q_k/\varphi_k]$. By induction hypothesis both of these formulas are derivable from $\Gamma[q_1/\varphi_1, \ldots, q_k\varphi_k]$, establishing the thesis. □

Example. Since we already showed that $\vdash_L p \to p$, the Replacement Theorem allows us to infer that also $\vdash_L q \to q$ (by replacing p with q), that $\vdash_L (p \to \neg q) \to (p \to \neg q)$ (by replacing p with $p \to \neg q$) and, in general, that $\vdash_L \varphi \to \varphi$ for any propositional formula φ. ◁

Example. Likewise, from the previous proof that $\{p \to q, q \to r\} \vdash_L p \to r$ we can conclude that also $\{(p \to q) \to \neg p, \neg p \to (q \to \neg p)\} \vdash_L (p \to q) \to (q \to \neg p)$, by replacing p with $p \to q$, q with $\neg p$, and r by $q \to \neg p$. Observe that the replacement is simultaneous, i.e., we replace p with $p \to q$ at the same time as we replace q with $\neg p$, so that the new occurrences of p and q do not get replaced recursively.

In general, the Replacement Theorem allows us to conclude that implication is transitive: $\{\varphi \to \psi, \psi \to \gamma\} \vdash_L \varphi \to \gamma$ for any propositional formulas φ, ψ and γ. ◁

Exercise 23. Prove the semantic counterparts of the two last theorems, i.e., show that:

(a) if $\Gamma \models \psi$ and $\Gamma \cup \{\psi\} \models \varphi$, then $\Gamma \models \varphi$;

(b) if $\Gamma \models \psi$, then $\Gamma[q_1/\varphi_1, \ldots, q_k/\varphi_k] \models \psi[q_1/\varphi_1, \ldots, q_k/\varphi_k]$.

The first of these properties is sometimes called *transitivity of entailment*.

The next result is a syntactic description of the semantics of implication: to prove that $\varphi \to \psi$, it suffices to prove ψ using φ as a hypothesis.

Theorem 11 (Deduction Theorem)**.** If $\Gamma \cup \{\varphi\} \vdash_L \psi$, then $\Gamma \vdash_L \varphi \to \psi$.

Proof. By induction on the derivation of $\Gamma \cup \{\varphi\} \vdash_L \psi$.

If ψ is an instance of an axiom, then the following derivation shows that $\Gamma \vdash \varphi \to \psi$.

1. ψ	(Ax.)
2. $\psi \to (\varphi \to \psi)$	(Ax.1)
3. $\varphi \to \psi$	MP(2,1)

If $\psi \in \Gamma$, then a similar derivation establishes the thesis – the justification for the first step now being (Hyp) instead of (Ax.).

If ψ is φ, then we can obtain a proof that $\Gamma \vdash_L \varphi \to \varphi$ by applying the Replacement Theorem to the proof of $\vdash_L p \to p$ shown above.

Finally, suppose that ψ is obtained by (MP) from θ and $\theta \to \psi$. By induction hypothesis, there are derivations $\mathcal{D}_\theta$ of $\Gamma \vdash_L \varphi \to \theta$ and $\mathcal{D}_{\theta\to\psi}$ of $\Gamma \vdash_L \varphi \to (\theta \to \psi)$, from which we can build the following derivation.

$\mathcal{D}_\theta$	
n. $\varphi \to \theta$	
$\mathcal{D}_{\theta\to\psi}$	
k. $\varphi \to (\theta \to \psi)$	
$k+1$. $(\varphi \to (\theta \to \psi)) \to ((\varphi \to \theta) \to (\varphi \to \psi))$	(Ax.2)
$k+2$. $(\varphi \to \theta) \to (\varphi \to \psi)$	MP(k+1,k)
$k+3$. $\varphi \to \psi$	MP(k+2,n)

This proof also sheds some light on the particular choices for (Ax.1) and (Ax.2): they are designed to make its two distinct cases easy. □

Example. As an example of how to use the Deduction Theorem, we prove that $\vdash_L \neg q \to (q \to p)$. Instead of constructing a derivation of this judgement directly, we instead prove that $\{\neg q\} \vdash_L q \to p$.

Since we have $\neg q$ as a hypothesis, we can try to use (Ax.3) to derive the conclusion, using the instantiation $(\neg p \to \neg q) \to (q \to p)$. The premise of this axiom is derivable from (Ax.1), since we know that $\neg q$ holds. We thus obtain the following derivation.

1. $\neg q$	(Hyp)
2. $\neg q \to (\neg p \to \neg q)$	(Ax.1)
3. $\neg p \to \neg q$	MP(2,1)
4. $(\neg p \to \neg q) \to (q \to p)$	(Ax.3)
5. $q \to p$	MP(4,3)

Since $\{\neg q\} \vdash_L q \to p$, the Deduction Theorem allows us to conclude that $\vdash_L \neg q \to (q \to p)$. ◁

Example. Applying the Replacement Theorem to the previous example, we can also conclude that $\vdash_L \neg\varphi \to (\varphi \to \psi)$ for any formulas φ and ψ. ◁

The proof of the Deduction Theorem is a constructive proof: if we have a derivation showing that $\Gamma \cup \{\varphi\} \vdash_L \psi$, it doesn't only tell us that $\Gamma \vdash_L \varphi \to \psi$, but it also describes how to construct a derivation of this judgement.

Example. We use the derivation in the previous example and the proof of the Deduction Theorem to construct a derivation directly showing that $\vdash_L \neg q \to (q \to p)$.

We include horizontal lines to separate the steps in the original proof.

1. $\neg q \to ((\neg q \to \neg q) \to \neg q)$	(Ax.1)
2. $(\neg q \to ((\neg q \to \neg q) \to \neg q)) \to ((\neg q \to (\neg q \to \neg q)) \to (\neg q \to \neg q))$	(Ax.2)
3. $(\neg q \to (\neg q \to \neg q)) \to (\neg q \to \neg q)$	MP(2,1)
4. $\neg q \to (\neg q \to \neg q)$	(Ax.1)
5. $\neg q \to \neg q$	MP(3,4)
6. $\neg q \to (\neg p \to \neg q)$	(Ax.1)
7. $(\neg q \to (\neg p \to \neg q)) \to (\neg q \to (\neg q \to (\neg p \to \neg q)))$	(Ax.1)
8. $\neg q \to (\neg q \to (\neg p \to \neg q))$	MP(7,6)
9. $(\neg q \to (\neg q \to (\neg p \to \neg q))) \to ((\neg q \to \neg q) \to (\neg q \to (\neg p \to \neg q)))$	(Ax.2)
10. $(\neg q \to \neg q) \to (\neg q \to (\neg p \to \neg q))$	MP(9,8)
11. $\neg q \to (\neg p \to \neg q)$	MP(10,5)
12. $(\neg p \to \neg q) \to (q \to p)$	(Ax.3)
13. $((\neg p \to \neg q) \to (q \to p)) \to (\neg q \to ((\neg p \to \neg q) \to (q \to p)))$	(Ax.1)
14. $\neg q \to ((\neg p \to \neg q) \to (q \to p))$	MP(13,12)
15. $(\neg q \to ((\neg p \to \neg q) \to (q \to p))) \to ((\neg q \to (\neg p \to \neg q)) \to (\neg q \to (q \to p)))$	(Ax.2)
16. $(\neg q \to (\neg p \to \neg q)) \to (\neg q \to (q \to p))$	MP(15,14)
17. $\neg q \to (q \to p)$	MP(16,11)

Observe that this proof is not clever at all; in particular, the first 11 steps are spent deriving the formula $\neg q \to (\neg p \to \neg q)$, which is an instance of (Ax.1). We could therefore write the much shorter derivation

1. $\neg q \to (\neg p \to \neg q)$	(Ax.1)
2. $(\neg p \to \neg q) \to (q \to p)$	(Ax.3)
3. $((\neg p \to \neg q) \to (q \to p)) \to (\neg q \to ((\neg p \to \neg q) \to (q \to p)))$	(Ax.1)
4. $\neg q \to ((\neg p \to \neg q) \to (q \to p))$	MP(3,2)
5. $(\neg q \to ((\neg p \to \neg q) \to (q \to p))) \to ((\neg q \to (\neg p \to \neg q)) \to (\neg q \to (q \to p)))$	(Ax.2)
6. $(\neg q \to (\neg p \to \neg q)) \to (\neg q \to (q \to p))$	MP(5,4)
7. $\neg q \to (q \to p)$	MP(6,1)

establishing the same formula. However, this is not the point: by invoking the Deduction Theorem we avoid the need to write down any of these derivations explicitly anyway. ◁

The Deduction Theorem can be applied as many times as desired. In practice, when proving implications, it is usually useful to assume all antecedents of the target formula and invoke the Deduction Theorem as many times as necessary.

Exercise 24. Apply the construction in the proof of the Deduction Theorem to the derivation of $\{p \to q, q \to r\} \vdash_L p \to r$ in order to obtain a derivation of $\{p \to q\} \vdash_L (q \to r) \to (p \to r)$. Can you simplify the resulting derivation? Could you have found it directly?

By applying the Deduction Theorem again, we conclude that $\vdash_L (p \to q) \to ((q \to r) \to (p \to r))$. How long would this derivation be?

Exercise 25. Using the Deduction Theorem (and possibly any of the other results shown above), prove that the following judgements hold.

(a) $\vdash_L (p \to (p \to q)) \to (p \to q)$ (b) $\{p \to \neg q\} \vdash_L p \to (q \to r)$

Proving validity of propositional formulas in the Hilbert calculus is trickier than in other systems, and we now show some examples that serve two purposes: they illustrate a bit more the kind of thinking required in this proofs, and they provide some more lemmas that we can use in later proofs.

One of the defining characteristics of classical logic is that double negations do not change the truth value of formulas. In other words, formulas φ and $\neg\neg\varphi$ are equivalent. The next examples show that, indeed, $\vdash_L \neg\neg p \to p$ and $\vdash_L p \to \neg\neg p$.

Example. We start by deriving $\neg\neg p \to p$. Since this formula contains negations, our derivation likely needs to use (Ax.3) at some point. If we try to use it for the final step, we are faced with the instance $(\neg p \to \neg\neg\neg p) \to (\neg\neg p \to p)$.

This might not look to promising, since the number of negations in this formula is even higher. But let us try to continue this line of reasoning: in order to prove $\neg p \to \neg\neg\neg p$, we could again try to use (Ax.3), this time instantiating it as $(\neg\neg\neg\neg p \to \neg\neg p) \to (\neg p \to \neg\neg\neg p)$.

Although this may look like a neverending process, the premise of this implication actually gives us a way out: the consequent of $\neg\neg\neg\neg p \to \neg\neg p$ is the antecedent of the formula that we are trying to prove $(\neg\neg p \to p)$. If we use the Deduction Theorem, we can use $\neg\neg p$ as hypothesis and derive this implication using (Ax.1). This yields the following derivation for $\{\neg\neg p\} \vdash_L p$.

1. $\neg\neg p \to (\neg\neg\neg\neg p \to \neg\neg p)$	(Ax.1)
2. $\neg\neg p$	(Hyp)
3. $\neg\neg\neg\neg p \to \neg\neg p$	MP(1,2)
4. $(\neg\neg\neg\neg p \to \neg\neg p) \to (\neg p \to \neg\neg\neg p)$	(Ax.3)
5. $\neg p \to \neg\neg\neg p$	MP(4,3)
6. $(\neg p \to \neg\neg\neg p) \to (\neg\neg p \to p)$	(Ax.3)
7. $\neg\neg p \to p$	MP(6,5)
8. p	MP(7,2)

Applying the Deduction Theorem, we conclude that $\vdash_L \neg\neg p \to p$, and by the Replacement Theorem it follows that $\vdash_L \neg\neg\varphi \to \varphi$ for any formula φ. ◁

This example includes a characteristic that we meet often in proofs by contradiction: we want to prove $\vdash_L \varphi \to \psi$ using the Deduction Theorem, but we actually start by deriving $\{\varphi\} \vdash_L \varphi \to \psi$. Although the final application of (MP) seems strange at first glance, since we already obtained the formula we are looking for, it is necessary: otherwise, the Deduction Theorem would give us $\vdash_L \varphi \to (\varphi \to \psi)$. Alternatively, one could also argue that from this the original result follows, since $\vdash_L (p \to (p \to q)) \to (p \to q)$.

Example. We now derive the converse implication, i.e., $\vdash_L p \to \neg\neg p$. This is much simpler: again the fact that there are negations suggests that we look into (Ax.3); the relevant instance is $(\neg\neg\neg p \to \neg p) \to (p \to \neg\neg p)$, and the antecedent of this implication is simply the formula in the previous example after we replace p with $\neg p$. This allows us to write the following derivation.

1. $\neg\neg\neg p \to \neg p$	Lemma
2. $(\neg\neg\neg p \to \neg p) \to (p \to \neg\neg p)$	(Ax.3)
3. $p \to \neg\neg p$	MP(2,1)

This establishes that $\vdash_L p \to \neg\neg p$, and by the Replacement Theorem we can also conclude that $\vdash_L \varphi \to \neg\neg\varphi$ for any formula φ. ◁

Exercise 26. As we saw earlier, negation can be defined by abbreviation, expanding $\neg\varphi$ as $\varphi \to \bot$. Show that $\vdash_L p \to \neg\neg p$ also if we define negation in this way, i.e., that $\vdash_L p \to ((p \to \bot) \to \bot)$ using only (Ax.1) and (Ax.2). Here $\bot$ is treated as a propositional symbol.

Example. The next example is one of the paradoxes of implication, embodying proof by contradiction: we show that $\vdash_L (\neg p \to p) \to p$.

In order to use the Deduction Theorem, we start by adding $\neg p \to p$ as a hypothesis. This gives us a few options: we can use the tautology $\neg p \to (p \to \varphi)$ and combine it with the appropriate instance of (Ax.2) to obtain $(\neg p \to p) \to (\neg p \to \varphi)$, whose antecedent is our hypothesis. In order to choose the relevant formula for φ, we note that $\neg p$ (the antecedent of $\neg p \to \varphi$) is the negation of the consequent of the formula we want to prove $((\neg p \to p) \to p)$; this suggests that we take φ to be $\neg(\neg p \to p)$ and use (Ax.3), obtaining the following derivation.

1. $\neg p \to (p \to (\neg(\neg p \to p)))$	Lemma
2. $(\neg p \to (p \to (\neg(\neg p \to p)))) \to ((\neg p \to p) \to (\neg p \to (\neg(\neg p \to p))))$	(Ax.2)
3. $(\neg p \to p) \to (\neg p \to (\neg(\neg p \to p)))$	MP(2,1)
4. $\neg p \to p$	(Hyp)
5. $\neg p \to (\neg(\neg p \to p))$	MP(3,4)
6. $(\neg p \to (\neg(\neg p \to p))) \to ((\neg p \to p) \to p)$	(Ax.3)
7. $(\neg p \to p) \to p$	MP(6,5)
8. p	MP(7,4)

Once again, we proved the desired formula using its antecedent as a hypothesis. This derivation shows that $\{\neg p \to p\} \vdash_L p$, and the Deduction Theorem now gives us $\vdash_L (\neg p \to p) \to p$. ◁

Example. Our last example is another of the paradoxes of implication: we show that $\vdash_L ((p \to q) \to p) \to p$. Using the Deduction Theorem, we first prove that $\{(p \to q) \to p\} \vdash_L p$. The key idea here is reusing previous results: we know that $\vdash_L \neg p \to (p \to q)$ and that $\vdash_L (\neg p \to p) \to p$; these results allow us to conclude the thesis by using (Ax.1) in conjunction with the hypothesis to derive the formula $\neg p \to ((p \to q) \to p)$.

1. $((p \to q) \to p) \to (\neg p \to ((p \to q) \to p))$	(Ax.1)
2. $(p \to q) \to p$	(Hyp)
3. $\neg p \to ((p \to q) \to p)$	MP(1,2)
4. $(\neg p \to ((p \to q) \to p)) \to ((\neg p \to (p \to q)) \to (\neg p \to p))$	(Ax.2)
5. $(\neg p \to (p \to q)) \to (\neg p \to p)$	MP(4,3)
6. $\neg p \to (p \to q)$	Lemma
7. $\neg p \to p$	MP(5,6)
8. $(\neg p \to p) \to p$	Lemma
9. p	MP(8,7)

Applying the Deduction Theorem we establish the desired result. ◁

Exercise 27. Show that the following properties of negation (the first three of which are variations upon (Ax.3)) can all be proved using the Hilbert calculus for propositional logic.

(a) $\vdash_L (p \to q) \to (\neg q \to \neg p)$

(b) $\vdash_L (\neg p \to q) \to (\neg q \to p)$

(c) $\vdash_L (p \to \neg q) \to (q \to \neg p)$

(d) $\{q \to p, \neg q \to p\} \vdash_L p$

Exercise 28. Show that $\vdash_L p \to (\neg q \to \neg(p \to q))$. This is another description of the semantics of implication, this time in negated form.

2.5.3 Completeness

We now have the necessary tools to show that the Hilbert calculus for classical propositional logic is complete. The proof that we show uses a technique originally developed by Leon Henkin, and which is widely used in many logics. It relies on building a *complete extension* of the Hilbert calculus, by adding axioms to it such that, for any formula, either it is provable or its negation is provable – but only one of these is the case.

In this section, we refer to the Hilbert calculus for classical propositional logic as defined above by *system L*.

Definition. An *extension* of system L is a Hilbert calculus obtained by adding axioms to system L.

We identify extensions of system L with the set of additional axioms. In particular, given two extensions L' and L'' of L, we write $L' \subseteq L''$ if all axioms of L' are also axioms of L''.

We write $\Gamma \vdash_{L'} \varphi$ to denote that φ can be derived from Γ using the axioms of the extension L', as well as the usual axioms and inference rule of system L. Note that Hypothetical Reasoning and the Deduction Theorem still apply to any extension of system L (since the former only manipulates derivations and the proof of the latter only uses (Ax.1) and (Ax.2)), but results such as the Replacement Theorem may not, since axioms in extensions of system L do not need to be schematic.

Definition. An extension L' of system L is *consistent* if there is no formula φ such that $\vdash_{L'} \varphi$ and $\vdash_{L'} \neg\varphi$.

Lemma 10. Let L' and L'' be two extensions of L such that $L' \subseteq L''$. For any set Γ and formula φ, if $\Gamma \vdash_{L'} \varphi$, then $\Gamma \vdash_{L''} \varphi$.

Proof. Straightforward from the definition of derivation, since all axioms of L' are axioms of L'' by hypothesis. □

Corollary 3. If $\Gamma \vdash_L \varphi$, then $\Gamma \vdash_{L'} \varphi$ for any extension L' of system L.

Lemma 11. An extension L' of system L is consistent iff there is a formula φ such that $\nvdash_{L'} \varphi$.

Proof. Suppose that L' is consistent. Then by definition $\nvdash_{L'} \neg\varphi$ for any formula φ that is an axiom of L'.

Suppose that L' is not consistent, and let φ be such that $\vdash_{L'} \varphi$ and $\vdash_{L'} \neg\varphi$. Since L' is an extension of system L, Corollary 3 ensures that also $\vdash_{L'} \neg\varphi \to (\varphi \to \psi)$ for every ψ. By combining these three derivations and applying (MP) twice, it follows that $\vdash_{L'} \psi$. □

Lemma 12. Let L' be an extension of system L and φ be a formula such that $\nvdash_{L'} \neg\varphi$. Then $L' \cup \{\varphi\}$ is consistent.

Proof. Let L'' denote $L' \cup \{\varphi\}$, and assume that L'' is inconsistent. By Lemma 11, $\vdash_{L''} \neg\varphi$, which is equivalent to $\{\varphi\} \vdash_{L'} \neg\varphi$. Furthermore, since $\{\neg\neg\varphi\} \vdash_L \varphi$, we also know that $\{\neg\neg\varphi\} \vdash_{L'} \varphi$. By Theorem 9 it follows that $\{\neg\neg\varphi\} \vdash_{L'} \neg\varphi$.

The Deduction Theorem for L' then yields that $\vdash_{L'} (\neg\neg\varphi) \to \neg\varphi$; but since $\vdash_L (\neg p \to p) \to p$, the Replacement Theorem allows us to conclude that $\vdash_L (\neg\neg\varphi \to \neg\varphi) \to \neg\varphi$, and by Corollary 3 it also follows that $\vdash_{L'} (\neg\neg\varphi \to \neg\varphi) \to \neg\varphi$. Therefore we conclude that $\vdash_{L'} \neg\varphi$ by (MP), contradicting the hypothesis. □

Theorem 12 (Completeness)**.** If $\Gamma \models \varphi$, then $\Gamma \vdash_L \varphi$.

Proof. Suppose that $\Gamma \nvdash_L \varphi$.

Let $\varphi_0, \ldots, \varphi_n, \ldots$ be an enumeration of all propositional formulas.[6] Define a sequence of extensions of L as follows.

$$L_0 = L \cup \Gamma \cup \{\neg\varphi\}$$

$$L_{k+1} = \begin{cases} L_k \cup \{\varphi_k\} & \text{if } \nvdash_{L_k} \neg\varphi_k \\ L_k & \text{otherwise} \end{cases}$$

[6]Since the set of propositional symbols is countable, the set of propositional formulas is also countable, and therefore it has an enumeration. Note that we do not require this enumeration to be computable, although such an effective enumeration can also be shown to exist.

Using Lemma 12, we can show straightforwardly that each L_k is consistent.

- If L_0 were inconsistent, then by Lemma 11 we would have that $\vdash_{L_0} \varphi$, or, equivalently, that $\Gamma \cup \{\neg\varphi\} \vdash_L \varphi$. By the Deduction Theorem it would then follow that $\Gamma \vdash_L \neg\varphi \to \varphi$, and since $\vdash_L (\neg\varphi \to \varphi) \to \varphi$ we would conclude that $\Gamma \vdash_L \varphi$, contradicting the hypothesis.

- Consistency of L_{n+1} follows immediately from consistency of L_n by the lemma.

Define $L^\infty = \cup_{k=0}^{+\infty} L_k$. Then L^∞ is also consistent: suppose that $\vdash_{L^\infty} \psi$ and $\vdash_{L^\infty \neg\psi}$. Since derivations are finite, the number of axioms used in these two derivations are finite. By choosing high enough k (in other words, such that L_k contains all these axioms), this implies that $\vdash_{L_k} \psi$ and $\vdash_{L_k} \neg\psi$, contradicting consistency of L_k.

Furthermore, by construction, L^∞ is *categorical*, meaning that for every ψ either $\vdash_{L^\infty} \psi$ or $\vdash_{L^\infty} \neg\psi$: taking k such that φ_k is ψ, if $\nvdash_{L^\infty} \neg\psi$ then ψ was added to L_{k+1}, and therefore $\vdash_{L^\infty} \psi$. Also observe that $\Gamma \subseteq L^\infty$.

Define a valuation V by $V(p) = \top$ iff $\vdash_{L^\infty} p$. By induction, it follows that $V \models \psi$ iff $\vdash_{L^\infty} \psi$.

- The case when ψ is an atomic formula follows by definition of V.

- The case when ψ is $\neg\psi'$ follows from induction hypothesis: $V \models \neg\psi'$ iff $V \not\models \psi'$ iff $\nvdash_{L^\infty} \psi'$ iff $\vdash_{L^\infty} \neg\psi'$ (where the last step uses categoricity of L^∞).

- The case when ψ is $\psi_1 \to \psi_2$ follows likewise from induction hypothesis. If $V \models \psi_1 \to \psi_2$, then $V \not\models \psi_1$ or $V \models \psi_2$.

  In the first case, by induction hypothesis $\nvdash_{L^\infty} \psi_1$, implying that $\vdash_{L^\infty} \neg\psi_1$ by categoricity; since $\vdash_L \neg q \to (q \to p)$, the thesis follows from the Replacement Theorem and Corollary 3.

  In the second case, by induction hypothesis $\vdash_{L^\infty} \psi_2$, and by (Ax.1) it follows that $\vdash_{L^\infty} \psi_1 \to \psi_2$, again establishing the thesis.

  Conversely, if $V \not\models \psi_1 \to \psi_2$, then $V \models \psi_1$ and $V \not\models \psi_2$. By induction hypothesis, $\vdash_{L^\infty} \psi_1$ and $\nvdash_{L^\infty} \psi_2$, and categoricity of L^∞ yields $\vdash_{L^\infty} \neg\psi_2$. Since $\vdash_L p \to (\neg q \to (\neg(p \to q)))$, applying the Replacement Theorem, Corollary 3 and (MP) twice yields that $\vdash_{L^\infty} \neg(\psi_1 \to \psi_2)$.

In particular, $V \models \Gamma$ but $V \not\models \varphi$, whence $\Gamma \not\models \varphi$. □

Using the soundness and completeness of the Hilbert calculus, we can obtain very simple proofs of complex results for propositional logic. The Replacement Theorem immediately gives us the following.

Theorem 13. If $\Gamma \models \psi$, then $\Gamma[p_1/\varphi_1, \ldots, p_k/\varphi_k] \models \psi[p_1/\varphi_1, \ldots, p_k/\varphi_k]$.

Proof. Suppose that $\Gamma \models \psi$. Then $\Gamma \vdash_L \psi$ by completeness of the Hilbert calculus for classical propositional logic. From the Replacement Theorem we conclude that $\Gamma[p_1/\varphi_1, \ldots, p_k/\varphi_k] \vdash_L \varphi[p_1/\varphi_1, \ldots, p_k/\varphi_k]$. We can now apply soundness of the calculus in order to obtain that also $\Gamma[p_1/\varphi_1, \ldots, p_k/\varphi_k] \models \varphi[p_1/\varphi_1, \ldots, p_k/\varphi_k]$. □

A more important property, which we already used in previous sections, is compactness.

Theorem 14 (Compactness)**.** If $\Gamma \models \varphi$, then there is a finite set $\Gamma_{\mathsf{fin}} \subset \Gamma$ such that $\Gamma_{\mathsf{fin}} \models \varphi$.

Proof. If $\Gamma \models \varphi$, then by completeness of the Hilbert calculus there is a derivation of $\Gamma \vdash_L \varphi$. But this derivation can only use a finite number of hypotheses from Γ; denoting the set of these hypotheses by Γ_{fin}, the same derivation also shows that $\Gamma_{\mathsf{fin}} \vdash_L \varphi$. Since this calculus is sound, it follows that $\Gamma_{\mathsf{fin}} \models \varphi$. □

2.6 Resolution

In this section we focus on a decision procedure of a much more algorithmic nature: resolution. Historically, this technique originated in the AI community in the 1960s as a reasoning method for first-order logic, and eventually gave birth to the field of Logic Programming. The application of its ideas to propositional logic forms the core of SAT solving, one of the most successful areas in automated reasoning.

Resolution works in the *clausal fragment* of propositional logic. Clauses are simply disjunctions of propositional symbols and their negations (jointly known as *literals*), e.g. $a \vee b \vee c$ or $p \vee \neg q$. By reasoning over sets of clauses, we can capture the full language of propositional logic.

Definition. A formula is in *conjunctive normal form* if it is a conjunction of clauses.

Alternatively, we can define conjunctive normal form by means of a BNF-style grammar.

$$C ::= \top \mid D \wedge C \qquad D ::= \bot \mid L \vee D \qquad L ::= p \mid \neg p$$

The formulas $\top$ and $\bot$ correspond to the empty cases and are traditionally omitted: an empty conjunction is true (all of its formulas are true), while an empty disjunction is false (none of its formulas is true). In the context of resolution, it is usual to represent the empty clause as $\Box$, rather than $\bot$.

Normal forms are used in many applications for restricting the set of formulas that needs to be considered in order to gain efficiency. Other normal forms used in the context of propositional logic are disjunctive normal form and implicative normal form. The key property is that any formula can be transformed in an equivalent one in normal form: in the case of CNF, this is stronger than just requiring $\{\neg, \vee, \wedge\}$ to be a functionally complete set of connectives – we also need to obtain a formula with the internal structure corresponding to a CNF.

In order to rewrite an arbitrary formula φ into CNF, we apply the following strategy.

1. Eliminate $\rightarrow$ and $\leftrightarrow$, by using the rules

$$(\varphi \rightarrow \psi) \rightsquigarrow_{\mathsf{CNF}} (\neg\varphi \vee \psi)$$
$$(\varphi \leftrightarrow \psi) \rightsquigarrow_{\mathsf{CNF}} ((\neg\varphi \vee \psi) \wedge (\varphi \vee \neg\psi))$$

2. Move all occurrences of $\neg$ inwards, so that they only occur at the atomic level, by using the rules

$$\neg(\varphi \vee \psi) \rightsquigarrow_{\mathsf{CNF}} (\neg\varphi \wedge \neg\psi)$$
$$\neg(\varphi \wedge \psi) \rightsquigarrow_{\mathsf{CNF}} (\neg\varphi \vee \neg\psi)$$
$$(\neg\neg\varphi) \rightsquigarrow_{\mathsf{CNF}} \varphi$$

3. Distribute $\vee$ over $\wedge$, by using the rules

$$(\varphi \vee (\psi \wedge \theta)) \rightsquigarrow_{\mathsf{CNF}} ((\varphi \vee \psi) \wedge (\varphi \vee \theta))$$
$$((\varphi \wedge \psi) \vee \theta) \rightsquigarrow_{\mathsf{CNF}} ((\varphi \vee \theta) \wedge (\psi \vee \theta))$$

Example. Consider the propositional formula $p \leftrightarrow (\neg(q \vee \neg p) \rightarrow \neg(q \wedge r))$. Rewriting this formula to CNF proceeds as follows, where at each step we underline the subformulas to which a rule is being applied.

$$
\begin{aligned}
&\underline{p \leftrightarrow (\neg(q \vee \neg p) \rightarrow \neg(q \wedge r))} \\
&\quad \rightsquigarrow_{\mathsf{CNF}} (\neg p \vee (\underline{\neg(q \vee \neg p)} \rightarrow \neg(q \wedge r))) \wedge (p \vee \neg(\underline{\neg(q \vee \neg p)} \rightarrow \neg(q \wedge r))) \\
&\quad \rightsquigarrow_{\mathsf{CNF}} (\neg p \vee (\underline{\neg\neg(q \vee \neg p)} \vee \underline{\neg(q \wedge r)})) \wedge (p \vee \neg(\underline{\neg\neg(q \vee \neg p)} \vee \underline{\neg(q \wedge r)})) \\
&\quad \rightsquigarrow_{\mathsf{CNF}} (\neg p \vee ((q \vee \neg p) \vee (\neg q \vee \neg r))) \wedge (p \vee \underline{\neg((q \vee \neg p) \vee (\neg q \vee \neg r))}) \\
&\quad \rightsquigarrow_{\mathsf{CNF}} (\neg p \vee ((q \vee \neg p) \vee (\neg q \vee \neg r))) \wedge (p \vee (\underline{\neg(q \vee \neg p)} \wedge \underline{\neg(\neg q \vee \neg r)})) \\
&\quad \rightsquigarrow_{\mathsf{CNF}} (\neg p \vee ((q \vee \neg p) \vee (\neg q \vee \neg r))) \wedge (p \vee ((\neg q \wedge \underline{\neg\neg p}) \wedge (\underline{\neg\neg q} \wedge \underline{\neg\neg r}))) \\
&\quad \rightsquigarrow_{\mathsf{CNF}} (\neg p \vee ((q \vee \neg p) \vee (\neg q \vee \neg r))) \wedge (\underline{p \vee ((\neg q \wedge p) \wedge (q \wedge r))}) \\
&\quad \rightsquigarrow_{\mathsf{CNF}} (\neg p \vee ((q \vee \neg p) \vee (\neg q \vee \neg r))) \wedge ((\underline{p \vee (\neg q \wedge p)}) \wedge (\underline{p \vee (q \wedge r)})) \\
&\quad \rightsquigarrow_{\mathsf{CNF}} (\neg p \vee ((q \vee \neg p) \vee (\neg q \vee \neg r))) \wedge (((p \vee \neg q) \wedge (p \vee p)) \wedge ((p \vee q) \wedge (p \vee r)))
\end{aligned}
$$

The last formula is in CNF. ◁

The parentheses in the last formula are cumbersome, and moreover they are not really necessary. Indeed, disjunction and conjunction in propositional logic are associative, commutative and idempotent; that is, the following pairs of formulas are equivalent.

$\varphi \vee (\psi \vee \gamma)$ and $(\varphi \vee \psi) \vee \gamma$	$\varphi \vee \psi$ and $\psi \vee \varphi$	$\varphi \vee \varphi$ and φ
$\varphi \wedge (\psi \wedge \gamma)$ and $(\varphi \wedge \psi) \wedge \gamma$	$\varphi \wedge \psi$ and $\psi \wedge \varphi$	$\varphi \wedge \varphi$ and φ

This allows us to normalize formulas further by ignoring permutations and duplicates of clauses and of literals inside the same clause. In other words, we can view clauses as sets of literals and a CNF as a set of clauses.

Formally, if $D = L_1 \vee \ldots \vee L_n$ is a clause, then we define $\Gamma_D = \{L_1, \ldots, L_n\}$, and if $C = D_1 \wedge \ldots \wedge D_m$ is a CNF, then we define $\Gamma_C = \{\Gamma_{D_1}, \ldots, \Gamma_{D_m}\}$. For an arbitrary formula φ, we define $\Gamma_\varphi = \Gamma_{C_\varphi}$, where C_φ is the result of rewriting φ into CNF.

Example. Continuing with the previous example, if we take φ to be $p \leftrightarrow (\neg(q \vee \neg p) \rightarrow \neg(q \wedge r))$, then C_φ is $(\neg p \vee ((q \vee \neg p) \vee (\neg q \vee \neg r))) \wedge (((p \vee \neg q) \wedge (p \vee p)) \wedge ((p \vee q) \wedge (p \vee r)))$.

Disregarding parenthesis, we can rewrite C_φ more simply as

$$
\underbrace{(\neg p \vee q \vee \neg p \vee \neg q \vee \neg r)}_{\{\neg p, q, \neg q, r\}} \wedge \underbrace{(p \vee \neg q)}_{\{p, \neg q\}} \wedge \underbrace{(p \vee p)}_{\{p\}} \wedge \underbrace{(p \vee q)}_{\{p, q\}} \wedge \underbrace{(p \vee r)}_{\{p, r\}}
$$

obtaining $\Gamma_\varphi = \{\{\neg p, q, \neg q, r\}, \{p, \neg q\}, \{p\}, \{p, q\}, \{p, r\}\}$, which is equivalent to the original formula. ◁

Exercise 29. Rewrite the following formulas into CNF.

(a) $p \rightarrow (q \vee p)$

(b) $(p \wedge q) \rightarrow (p \vee q)$

(c) $(p \wedge q) \vee (\neg p \wedge \neg q)$

(d) $(p \leftrightarrow q) \rightarrow (\neg p \rightarrow (r \wedge s))$

We now show that rewriting into conjunctive normal form is sound. For convenience, we split this into several steps.

Lemma 13 (Soundness of rewriting)**.** For all formulas φ and ψ and valuation V, if $\varphi \rightsquigarrow_{\mathsf{CNF}} \psi$, then $V \models \varphi$ iff $V \models \psi$.

Proof. This is a simple exercise in semantics. We provide two representative examples.

$$\begin{aligned} V \models \varphi \to \psi &\text{ iff } V \not\models \varphi \text{ or } V \models \psi \\ &\text{iff } V \models \neg\varphi \text{ or } V \models \psi \\ &\text{iff } V \models \neg\varphi \vee \psi \end{aligned} \qquad \begin{aligned} V \models \neg(\varphi \vee \psi) &\text{ iff } V \not\models \varphi \vee \psi \\ &\text{iff } V \not\models \varphi \text{ and } V \not\models \psi \\ &\text{iff } V \models \neg\varphi \text{ and } V \models \neg\psi \\ &\text{iff } V \models \neg\varphi \wedge \neg\psi \end{aligned}$$ □

Exercise 30. Prove the remaining cases of Lemma 13.

Lemma 14 (Soundness of normalization)**.** For every formula φ and valuation V, $V \models \varphi$ iff $V \models C_\varphi$.

Proof. By induction on the number of steps in the transformation of φ into C_φ. If φ is already in CNF, then the thesis is trivial. Otherwise, we can apply a rewriting rule to φ, obtaining a new formula ψ such that C_φ and C_ψ coincide.

Since the semantics of classical propositional logic is truth-functional, Lemma 13 implies that $V \models \varphi$ iff $V \models \psi$ for any valuation V. By induction hypothesis, the latter is the case iff $V \models C_\psi$; since C_ψ and C_φ are the same, the thesis follows. □

Lemma 15. Let $D = L_1 \vee \ldots \vee L_n$ be a clause and V be a valuation. Then $V \models D$ iff $V \models L_i$ for some $1 \leq i \leq n$.

Proof. The result is nearly trivial due to the semantics of disjunction. We give a formal proof by induction on n. If $n = 0$, then D is $\Box$; by definition $V \not\models D$, and since there is no value $1 \leq i \leq 0$ it is also the case that there is no i for which $V \models L_i$.

Suppose that the thesis holds for $n-1$. For any valuation V, $V \models (L_1 \vee \ldots \vee L_{n-1}) \vee L_n$ iff $V \models L_1 \vee \ldots \vee L_{n-1}$ or $V \models L_n$. By induction hypothesis, the first condition is equivalent to $V \models L_i$ for some $1 \leq i \leq n-1$, so $V \models D$ iff $V \models L_i$ for some $1 \leq i \leq n-1$ or $V \models L_n$, which is equivalent to $V \models L_i$ for some $1 \leq i \leq n$. □

Lemma 16 (Soundness of the representation)**.** Let C be a formula in CNF and V be a valuation. Then $V \models C$ iff, for every $\Gamma \in \Gamma_C$, there exists $L \in \Gamma$ such that $V \models L$.

Proof. Let $C = D_1 \wedge \ldots \wedge D_n$. Again we proceed by induction on n. For $n = 0$, the result is trivial, since $V \models \top$ and the condition on $\Gamma_C = \emptyset$ is vacuously true.

Suppose the result holds for $n-1$. Then $V \models (D_1 \wedge \ldots \wedge D_{n-1}) \wedge D_n$ iff $V \models D_1 \wedge \ldots \wedge D_{n-1}$ and $V \models D_n$. By induction hypothesis, the first condition is equivalent to stating that, for every $1 \leq i \leq n-1$, there exists $L_i \in \Gamma_{D_i}$ such that $V \models L_i$; by Lemma 15, the second condition is equivalent to stating that $V \models L$ for some $L \in \Gamma_{D_n}$. Therefore $V \models C$ iff for every $\Gamma \in \Gamma_D$ there exists $L \in \Gamma$ such that $V \models L$. □

Corollary 4. For any formula φ and valuation V, $V \models \varphi$ iff for every $\Gamma \in \Gamma_\varphi$ there exists $L \in \Gamma$ such that $V \models L$.

Proof. Consequence of Lemmas 14 and 16. □

Following standard practice, hereafter we represent sets simply as lists and set union by a comma; we also identify clauses with sets. Thus, we write p, D for $\{p\} \cup D$, and D, D' for $D \cup D'$ or $D \vee D'$.

Definition. The *resolution* rule is the inference rule

$$\frac{p, D \qquad \neg p, D'}{D, D'}$$

where p is a propositional symbol, D and D' are clauses.

Definition. Let Γ be a set of formulas and φ be a formula. A *resolution proof* of φ from Γ is a sequence of clauses $D_1, \ldots, D_n$ such that:

- $D_n = \square$;
- each D_i is of the form D, D' where D, D' is obtained by applying the resolution rule with premises among the CNFs corresponding to the formulas in $\Gamma \cup \{\neg\varphi\}$ and $\{D_k \mid k < i\}$.

We write $\Gamma \vdash_R \varphi$ to denote that there is a resolution proof of φ from Γ.

Resolution proofs are usually depicted as a directed graph where each clause has either two incoming edges (from the premises of the application of the resolution rule that derives it) or none (if it comes from Γ or $\neg\varphi$). The graph is oriented downwards, i.e. new formulas are added below previous formulas (in order to be able to write the edges without arrowtips).

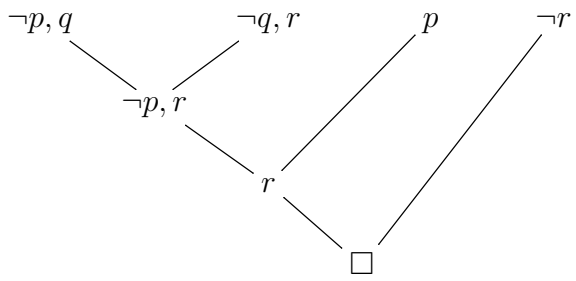
Example. The resolution rule is motivated by the classical syllogism:

$$\{p \to q, q \to r\} \models p \to r\,.$$

Taking $\Gamma = \{p \to q, q \to r\}$ and φ to be $p \to r = \varphi$, we show that $\Gamma \vdash_R \varphi$. As a first step, we convert $\Gamma \cup \{\neg\varphi\}$ to a set of CNFs, directly obtaining

$$\{\{\neg p, q\}, \{\neg q, r\}, \{p\}, \{\neg r\}\}\,.$$

We can then build the following graph.



In order to build a sequence of clauses as in the formal definition of resolution proof, we can simply take any topological ordering of the nodes in the graph. We write this proof to

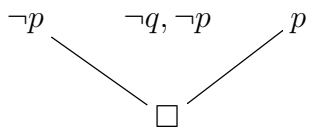
illustrate this construction, writing as justifications Γ or φ if the clause comes from one of those formulas, and $\mathrm{R}(m, n)$ if it comes from applying the resolution rule to the clauses in steps m and n.

1. $\neg p, q$	Γ
2. $\neg q, r$	Γ
3. p	φ
4. $\neg r$	φ
5. $\neg p, r$	R(1,2)
6. r	R(3,5)
7. $\square$	R(6,4)

Other topological orderings will give variations of this proof; these differences are immaterial – the same graph corresponds to several derivations that are all permutations of each other. $\triangleleft$

Example. We now show that $\{(p \vee q) \to \neg p\} \vdash_R \neg p$. Converting $(p \vee q) \to \neg p$ to CNF gives $(p \vee q) \to \neg p \rightsquigarrow_{\mathsf{CNF}} \neg(p \vee q) \vee \neg p \rightsquigarrow_{\mathsf{CNF}} (\neg p \wedge \neg q) \vee \neg p \rightsquigarrow_{\mathsf{CNF}} (\neg p \vee \neg p) \wedge (\neg q \vee \neg p)$, which yields the clauses $\neg p$; $\neg q, \neg p$; and p.

The resolution proof now looks like

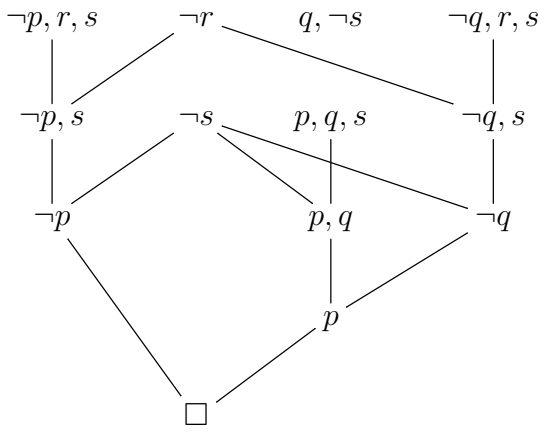


showing that we can derive the empty clause without even using all the available clauses. $\triangleleft$

Example. Consider now the judgement $\{(p \vee q) \to (r \vee s), s \to q, (\neg q \wedge \neg s) \to p\} \vdash_R \neg r \to s$. We start by converting each formula to CNF as follows:

$$\begin{aligned}
(p \vee q) \to (r \vee s) &\rightsquigarrow_{\mathsf{CNF}} \neg(p \vee q) \vee (r \vee s) \rightsquigarrow_{\mathsf{CNF}} (\neg p \wedge \neg q) \vee (r \vee s) \\
&\rightsquigarrow_{\mathsf{CNF}} (\neg p \vee r \vee s) \wedge (\neg q \vee r \vee s) \\
s \to q &\rightsquigarrow_{\mathsf{CNF}} \neg s \vee q \\
(\neg q \wedge \neg s) \to p &\rightsquigarrow_{\mathsf{CNF}} \neg(\neg q \wedge \neg s) \vee p \rightsquigarrow_{\mathsf{CNF}} \neg\neg q \vee \neg\neg s \vee p \rightsquigarrow_{\mathsf{CNF}} q \vee s \vee p \\
\neg(\neg r \to s) &\rightsquigarrow_{\mathsf{CNF}} \neg(\neg\neg r \vee s) \rightsquigarrow_{\mathsf{CNF}} \neg(r \vee s) \rightsquigarrow_{\mathsf{CNF}} \neg r \wedge \neg s
\end{aligned}$$

from which we can build the following resolution proof.



For completeness, we give a possible formal derivation corresponding to this graph.

1. $\neg p, r, s$	Γ
2. $\neg r$	φ
3. $\neg p, s$	R(1,2)
4. $\neg q, r, s$	Γ
5. $\neg q, s$	R(2,4)
6. $\neg s$	φ
7. $\neg p$	R(3,6)
8. $\neg q$	R(5,6)
9. p, q, s	Γ
10. p, q	R(6,9)
11. p	R(8,10)
12. $\square$	R(7,11)

We omitted the clause $q, \neg s$ from this proof, as it is not used to derive $\square$. $\triangleleft$

We stress that, for communicating between humans, the representation of resolution proofs as sequences is extremely unusual: its formal definition is very useful for proving properties of resolution, but its graph representation is much easier to read. However, resolution is mostly applied in the context of SAT solvers: automated tools for solving the satisfiability problem for propositional logic. For the purpose of representing proofs discovered by a SAT solver, their representation as a sequence is much more convenient. SAT solvers used in practice use many other techniques beyond the resolution rule, but they output proofs using formats that are essentially extensions of the format above.

In the previous examples we nearly always performed a particular kind of resolution where one of the clauses is a singleton literal. These steps are often called *unit resolution* steps (as a clause consisting of a single literal is called a *unit clause*), and are usually preferred: resolution proofs can easily become large if one blindly applies resolution steps randomly; unit resolution steps always generate smaller clauses and help eliminate propositional symbols from those that need to be considered. In the previous example, this is very clear: after performing resolution with $\neg r$, we only need to consider clauses involving p, q and s; after performing resolution with $\neg s$, we are left with clauses that only involve p and q. Thus the problem size is guaranteed to reduce, hopefully reaching the empty clause quickly.

Exercise 31. Decide whether the following judgements hold.

(a) $\{p \vee q, p \wedge q\} \vdash_R p$

(b) $\{\neg p \to p\} \vdash_R p$

(c) $\{p \wedge q, \neg p\} \vdash_R r$

(d) $\{p\} \vdash_R q \to p$

(e) $\{p \vee q, p \to r, q \to r\} \vdash_R r$

(f) $\{p \to (q \wedge r), \neg p\} \vdash_R q \vee r$

In the cases where they do not hold, can you use the resolution graph to find a valuation that invalidates the corresponding entailment?

Like all other deductive systems we saw, resolution is sound, and its soundness is relatively simple to prove.

Theorem 15 (Soundness)**.** If $\Gamma \vdash_R \varphi$, then $\Gamma \models \varphi$.

Proof. We first show that the resolution rule is sound: for every valuation V, if $V \models a, D$ and $V \models \neg a, D'$, then $V \models D, D'$.[7] If $V(a) = \top$, then $V \not\models \neg a$, hence $V \models D'$ and therefore $V \models D, D'$. If $V(a) = \bot$, then $V \not\models a$, hence $V \models D$ and therefore again $V \models D, D'$. Since one of these cases always holds, we conclude that $V \models D, D'$ whenever $V \models a, D$ and $V \models \neg a, D'$.

Using this result, we show by induction that, if $V \models \Gamma \cup \{\neg\varphi\}$, then $V \models C$ for every clause derived in a resolution proof of $\Gamma \vdash_R \neg\varphi$. For the clauses originating from the conversion of Γ or $\neg\varphi$ to CNF, this is a consequence of Corollary 4. For all other clauses, this is a consequence of the soundness of the resolution rule. In particular, this shows that $V \models \square$, which is a contradiction. Therefore $V \not\models \Gamma \cup \{\neg\varphi\}$ for every V. By Lemma 3, we conclude that $\Gamma \models \varphi$. □

A consequence of the proof of this theorem is that, if resolution fails, we can try to produce a counter-example to the original judgement by using the generated clauses to build a valuation. Unit clauses are particularly useful in this step, since each unit clause immediately determines the values of one propositional symbol. This is the basic principle behind all SAT solvers.

Theorem 16 (Completeness)**.** If $\Gamma \models \varphi$, then $\Gamma \vdash_R \varphi$.

Proof. As before, we invoke compactness and prove the result only for finite Γ.

Let Γ^+ be the set obtained from $\Gamma \cup \neg\varphi$ by converting all the formulas in this set to CNF and applying the resolution rule until no new clauses are generated. This procedure is guaranteed to terminate, since there are only finitely many different clauses that can be written with the propositional symbols used in Γ. (Note that it is essential that we view clauses as sets, in order to disregard duplicate literals.)

Suppose by contradiction that the resulting set does not contain the empty clause. Let $\{p_1, \ldots, p_n\}$ be the propositional symbols appearing in $\Gamma \cup \{\neg\varphi\}$, and define a valuation V incrementally (that is, by establishing its value on $p_1, \ldots, p_n$ in sequence) as follows: for each i, $V(p_i) = \bot$ if there is a clause in Γ^+ that contains $\neg p_i$ and such that all other literals are already assigned to false, and $V(p_i) = \top$ otherwise.

Let Γ_i^+ be the subset of Γ^+ containing all clauses that only use propositional symbols in $\{p_1, \ldots, p_i\}$. We show by induction on i that V is correctly defined and satisfies all clauses in Γ_i^+.

For $i = 0$ there is nothing to prove, since Γ^+ does not contain the empty clause. Assume now that the result holds for Γ_{i-1}^+ and that V assigns two different values to p_i. Then there are two clauses C_1 and C_2 such that C_1 contains p_i, C_2 contains $\neg p_i$, and all other literals in C_1 and C_2 are already assigned to $\bot$ under V. Therefore C_1 and C_2 only use propositional symbols in $\{p_1, \ldots, p_{i-1}\}$. Since Γ^+ is closed under resolution, the resolvent C' of C_1 and C_2 must also be in Γ^+, and since it also uses only propositional symbols in $\{p_1, \ldots, p_{i-1}\}$ it is in Γ_i^+; but then $V \not\models C'$, contradicting the induction hypothesis.

Therefore it is possible to define $V(p_i)$ such that V satisfies all clauses in Γ_i^+. In the particular case that $i = n$, we conclude that V is a model of $\Gamma \cup \{\neg\varphi\}$, which is a contradiction. Therefore Γ^+ must contain the empty clause. □

In an automated setting, we only work with finite sets of formulas, so completeness of resolution guarantees that this process always terminates. However, since the number of clauses

[7] Note that clauses are disjunctions, so that when we write $V \models D$ we mean that V makes at least one of the literals in D true. This is different from the usual notation with sets of formulas.

that can be written with n symbols is 3^n, automated provers based on resolution must use extremely powerful heuristics in order to reach $\square$ efficiently, if possible, or to find a valuation that satisfies the original set and stop.

2.7 Exercises

Exercise 32. Using the semantics of propositional logic, decide whether each of the following formulas is valid, satisfiable, falsifiable or unsatisfiable.

(a) $(a \to b) \to ((\neg a \to b) \to (a \vee b))$

(b) $((a \to b) \wedge (b \to c)) \to (c \to a)$

(c) $((a \to b) \wedge (b \to c)) \to (a \to c)$

(d) $((a \vee b) \wedge c) \to (\neg c \wedge (\neg a \to b))$

(e) $(\neg b \to \neg a) \to (a \to b)$

(f) $(a \wedge b) \wedge ((c \wedge \neg a) \vee (c \wedge \neg b))$

(g) $(a \to (b \to c)) \to ((a \to b) \to (b \to c))$

(h) $((a \to c) \wedge (b \to c)) \to ((a \vee b) \to c)$

(i) $(a \vee (a \to b)) \to (a \wedge (b \to a))$

(j) $(a \wedge b) \wedge ((c \wedge \neg a) \wedge (c \wedge \neg b))$

Exercise 33. Rewrite the formulas in Exercise 32 using only connectives in:

(a) $\{\neg, \to\}$ (b) $\{\neg, \wedge, \vee\}$ (c) $\{\bot, \to\}$

Exercise 34. Show that conjunction and disjunction are commutative, associative and idempotent, i.e., that the following pairs of formulas are equivalent.

(a) $\varphi \vee \psi$ and $\psi \vee \varphi$

(b) $\varphi \wedge \psi$ and $\psi \wedge \varphi$

(c) $(\varphi \vee \psi) \vee \theta$ and $\varphi \vee (\psi \vee \theta)$

(d) $(\varphi \wedge \psi) \wedge \theta$ and $\varphi \wedge (\psi \wedge \theta)$

(e) $\varphi \vee \varphi$ and φ

(f) $\varphi \wedge \varphi$ and φ

Furthermore, letting $\top$ and $\bot$ be 0-ary connectives representing formulas that are always true and always false (respectively), show that $\top$ is a unit for conjunction and absorvent for disjunction and that $\bot$ is a unit for disjuntion and absorbent for conjunction, i.e., that the following pairs of formulas are also equivalent.

(g) $\varphi \wedge \top$ and φ

(h) $\varphi \wedge \bot$ and $\bot$

(i) $\varphi \vee \top$ and $\top$

(j) $\varphi \vee \bot$ and φ

Exercise 35. Use the semantics of propositional logic to decide whether the following entailments hold.

(a) $\{a, \neg(b \to c)\} \models ((a \to b) \wedge (b \to c)) \to (c \to a)$

(b) $\{a, b \to \neg a, a \to b\} \models c$

(c) $\{a, a \to b\} \models c$

Exercise 36. Given a set of formulas Γ, denote by $\Gamma^{\models}$ the set $\{\varphi \mid \Gamma \models \varphi\}$. Show that the following properties hold.

(a) Reflexivity: $\Gamma \subseteq \Gamma^{\models}$

(b) Monotonicity: if $\Gamma \subseteq \Delta$, then $\Gamma^{\models} \subseteq \Delta^{\models}$

(c) Idempotence: $\Gamma^{\models} = \left(\Gamma^{\models}\right)^{\models}$

(d) Transitivity: if $\Delta \subseteq \Gamma^{\models}$, then $\Delta^{\models} \subseteq \Gamma^{\models}$

(e) *Ex falso quodlibet*: if Γ is inconsistent, then $\Gamma^{\models} = \{\text{all wffs}\}$

Exercise 37. Repeat the analysis in Exercises 32 and 35 using the tableaux calculus for propositional logic. Recall that deciding whether a formula is satisfiable or unsatisfiable requires building a tableau for that formula (rather than for its negation).

Exercise 38. Find sequent calculus proofs for each of the valid formulas in Exercise 32 and for each of the valid entailments in Exercise 35.

Exercise 39. Show that: if we omit rules $\wedge L_1$ and $\wedge L_2$ in the sequent calculus for classical propositional logic and use $\wedge L$ instead, then rules $\wedge L_1$ and $\wedge L_2$ are admissible in the resulting system.

Exercise 40. Complete the proof of the Completeness Theorem for the Hilbert calculus by showing that the valuation V constructed has the property that $V \models \psi$ iff $\vdash_{L^\infty} \psi$.

Exercise 41. Repeat the analysis in Exercises 32 and 35 using resolution. Recall that deciding whether a formula is satisfiable or unsatisfiable requires rewriting the formula itself in CNF (rather than its negation).

Exercise 42. Find resolution proofs of $\vdash_R \varphi$ for each valid formula φ in Exercise 32 by applying the construction in the proof of the Completeness Theorem.

Chapter 3

Propositional modal logic

In spite of its multiple applications, the expressive power of propositional logic is quite limited, due both to its very simple structure and to its truth-functional semantics. In this chapter, we introduce a family of logics, called *modal logics*, that extend this expressive power by means of a different type of connectives, called *modalities*.

Modalities are unary connectives that qualify the truth value of a formula, in the spirit of expressions such as "I believe that", "I know that", "I think that", "I must" or "I may" in natural languages.[1] These connectives do not have a truth-functional semantics, and as such cannot be expressed in propositional logic.

Modalities come in pairs, one with a "universal" flavor (typically written as $\Box$, read "box" or "necessarily") and the other with an "existential" flavor (typically written $\Diamond$, read "diamond" or "possibly"). They are related by the duality axioms:

$$\Box\varphi \leftrightarrow \neg\Diamond\neg\varphi \qquad \text{and} \qquad \Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$$

that are equivalent to the two de Morgan laws

$$\neg\Box\varphi \leftrightarrow \Diamond\neg\varphi \qquad \text{and} \qquad \neg\Diamond\varphi \leftrightarrow \Box\neg\varphi$$

Modal logics originated in philosophy as a logic to talk about logic. It is possible, for example, to encode Gödel's proof of incompleteness of arithmetic as a (provable) formal statement in a particular modal logic. They are also widely used in knowledge representation and reasoning, as modalities can be used to express agents' beliefs about the world.

3.1 Language and axioms

In this chapter we focus on modal propositional logics, which are logics whose syntax is obtained by adding modalities to propositional logic. Thus we again assume a countable set $\mathcal{P} = \{p_1, \ldots, p_n, \ldots\}$ of propositional symbols. The connectives are all the propositional connectives plus the two (unary) modalities $\Box$ and $\Diamond$, and the set of well-formed formulas is defined inductively as follows.

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \rightarrow \varphi) \mid (\varphi \leftrightarrow \varphi) \mid (\Box\varphi) \mid (\Diamond\varphi)$$

Differently from propositional logic, the semantics of modalities is not uniquely specified. This happens because the precise interpretation of what "necessarily" should mean can vary

[1] In natural language, verbs such as 'can', 'must' or 'may' are called *modal verbs*.

according to the context, and therefore the properties that one wants $\Box$ to satisfy may change. Intuitively, modalities talk about possible realities, with $\Box\varphi$ meaning that φ should hold in all possible scenarios, and $\Diamond\varphi$ meaning that φ holds in some possible scenario.

The precise meaning of a "possible scenario" depends on the exact context that one is modeling. In order to formalize it, one typically specifies the axioms that the modalities should satisfy. Some of the most commonly used axioms in modal logic are the following.

$$K : \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi) \qquad T : \Box\varphi \to \varphi$$
$$D : \Box\varphi \to \Diamond\varphi \qquad 4 : \Box\varphi \to \Box(\Box\varphi)$$
$$B : \varphi \to \Box\Diamond\varphi \qquad 5 : \Diamond\varphi \to \Box(\Diamond\varphi)$$

Axiom K is known as *normality*, and is usually assumed to hold for any necessity operator. It states that, if φ necessarily implies ψ and φ is necessary, then ψ is also necessary. The semantics that we will see automatically enforces this axiom; there are some modal logics that do not assume it, but they require considering different types of semantics, and we will not discuss them.

Axiom T states that anything that is necessary also holds. Although this seems like a natural requirement at first glance, one typical interpretation where it fails is *deontic* logic – the logic of Law –, where $\Box\varphi$ is interpreted as "φ must hold" (in a legal sense). Since the real world does not necessarily follow the law, this axiom is not assumed in this setting.

Axiom D is the *deontic* axiom, and it is one that is typically associated with a legal interpretation of the modalities. Here $\Box\varphi$ denotes that φ is required by law; from the duality relations, we can see that $\Diamond\varphi$ translates to "φ is allowed" (since $\neg\Box\varphi$ reads "$\neg\varphi$ is not required by law", i.e., φ is not forbidden). The axiom can then be read as stating that if something is mandatory, then it must be allowed.

The remaining axioms involve nested modalities, and their intuitive analysis is a bit more complex. Axiom 4 is the axiom of *introspection*. If we read $\Box$ as "I know that", then this axiom states that if we know something, then we are aware of knowing it. This is a very reasonable acception in Mathematics, and this axiom is often assumed when using modal logic to reason about proof theory.

Axiom B states that anything that holds is necessarily possible. While this may at first sound quite reasonable, this axiom actually fails to hold in many settings: it is stating a form of consistency between the formulas that are true and the formulas that may be true (as dictated by the modality). However, one of the interests of modal logic is precisely its ability to model *in*consistencies between those things that should hold and those things that actually holds – such inconsistencies occur often in practice, e.g. in a legal context, or when reasoning about human beliefs.

Finally, axiom 5 states that anything that is possible is necessarily possible. It is related to axiom 4 (anything that is necessary is necessarily necessary) and B (axioms 5 and T jointly imply B).

Some typical interpretations of the modalities are the following.

Epistemic logic. Here, $\Box$ stands for "I know" and $\Diamond$ for "It might be the case that". This logic is usually taken to satisfy only axiom K.

Deontic logic. As mentioned above, here we read $\Box$ as "it is mandatory" and $\Diamond$ as "it is allowed", and this modality satisfies K and D

Doxastic logic. This is the logic of belief: $\Box$ reads "I believe" and $\Diamond$ reads "it is plausible that". Doxastic modalities usually satisfy K, D, 4 and (debatably) B.

Temporal logic. Here we read the modalities as statements about time. If we read $\Box$ as "always" and $\Diamond$ as "some time", then all the above axioms hold. A more standard interpretation is to read $\Box$ as "always in the future" and $\Diamond$ as "sometime in the future"; this is a very useful logic in which to express properties of programs, and these modalities satisfy K, T, 4 and D, but not B or 5.

Exercise 1. For each of the axioms above, find a reasonable interpretation of "necessarily" that makes the axiom true and another that makes it false.

Exercise 2. From the duality axiom $\Box\varphi \leftrightarrow \neg\Diamond\neg\varphi$ and the informal interpretation of negation, check that the informal meaning of $\Diamond$ in all the examples above is correct.

3.2 Semantics

The semantics of modal logic is fundamentally different from that of propositional logic, because it is not truth-functional. In order to understand why this is the case, consider the following simple example.

Example. Consider a temporal modal logic where the intended meaning of $\Box\varphi$ is "φ is true in all future instants". Let ψ and γ be two formulas representing the propositions $2 + 2 = 4$ and "it is raining", respectively.

Suppose that it is currently raining. Then both ψ and γ are true. However, $\Box\psi$ is true (since $2 + 2$ will always be equal to 4), but $\Box\gamma$ is not (since it will hopefully not always be raining). Therefore, the truth value of $\Box\varphi$ is not determined solely by the truth value of φ. $\triangleleft$

The semantics that we present for modal logic is based on what is sometimes called the *possible worlds interpretation* of modal logic. In this interpretation, we consider different alternative models of propositional logic, called "worlds", and use the modalities to switch between worlds – so $\Box\varphi$ informally stands for "φ is true in all possible worlds" and $\Diamond\varphi$ for "φ is true in some possible world".

In order to realize the full potential of modal logic, we need to refine this idea one step further. This is because the notion of "possible world" is itself not universal: the sets of conceivably possible worlds in the mind of a scientist, a flat-earther or a creationist are very different from each other. Therefore, our semantics will also allow for the possibility that the set of possible worlds used to give meaning to modalities can change within the same model.

This leads us to the notion of Kripke structure.

Definition. A *Kripke structure*, or *frame*, is a pair $\langle W, R\rangle$, where W is a set whose elements are called *worlds* and $R \subseteq W \times W$ is a binary relation called the *accessibility* or *visibility* relation.

There are two possible semantics for modal logic based on frames: one where we take as true formulas that are true in all worlds, another where we choose the view of a particular

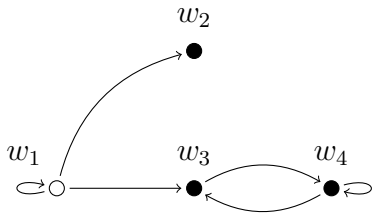
world. For applications to Computer Science, the latter semantics is often more useful, and this is the semantics that we will present.

Definition. A *directed Kripke structure*, or *directed frame*, is a triple $\langle W, R, w_0\rangle$ where $\langle W, R\rangle$ is a frame and $w_0 \in W$ is a world called the *initial world*.

Example. Consider a frame with four worlds, $W = \{w_1, w_2, w_3, w_4\}$, where w_1 is the initial world, and an accessibility relation defined by

$$R = \{\langle w_1, w_1\rangle, \langle w_1, w_2\rangle, \langle w_1, w_3\rangle, \langle w_3, w_4\rangle, \langle w_4, w_3\rangle, \langle w_4, w_4\rangle\}\,.$$

We can represent the frame $F_0 = \langle W, R, w_1\rangle$ graphically as follows.



Worlds are represented as vertices on the graph, with a circle representing the initial world, and the accessibility relation is modelled as edges (where an edge from w to w' means that wRw', or that w' is accessible from w). ◁

In order to obtain a model from a frame, we assign a propositional valuation V to each world of a frame.

Definition. A *(modal) valuation* over a frame $F = \langle W, R, w_0\rangle$ is a family $V : W \times \mathcal{P} \to \{\top, \bot\}$ of propositional valuations parameterized on W.

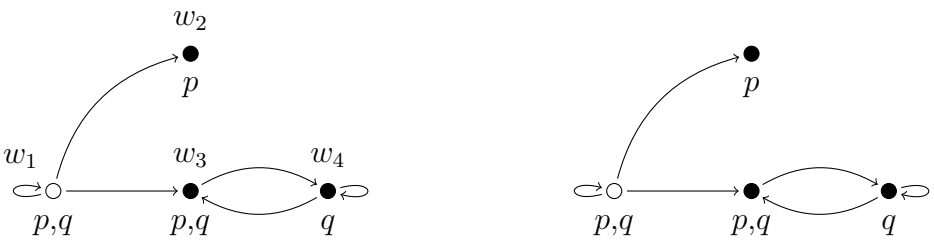
A model for modal logic is a pair $\langle F, V\rangle$, where F is a directed frame and V is a modal valuation over F.

Throughout this chapter, we use "valuation" to refer to a modal valuation, and explicitly write "propositional valuation" when referring to a valuation in the sense of the previous chapter. It is usual to represent valuations as functions $V : W \to 2^{\mathcal{P}}$, so that $V(w) = \{p \mid V(w, p) = \top\}$ is the set of propositional symbols satisfied in w.

Example. We can define a valuation V_0 over the frame F_0 in the previous example by $V_0(w_1) = V_0(w_3) = \{p, q\}$, $V_0(w_2) = \{p\}$ and $V_0(w_4) = \{q\}$. This is a shorthand way of writing the valuation defined by the following table.

	w_1	w_2	w_3	w_4
p	$\top$	$\top$	$\top$	$\bot$
q	$\top$	$\bot$	$\top$	$\top$

It is common to present a model (a frame together with a valuation) by attaching to each world the propositional symbols that are true at that world. If the names of the worlds are immaterial, they are sometimes left out altogether. The following are two possible ways of displaying valuation V_0 graphically.



The representation on the right is used mostly for exhibiting short examples that can be quickly analysed. If a discussion is desired, then it is best to include also the worlds' names. ◁

Given a model, we can define a notion of *local satisfaction*, essentially assigning a truth-value to every formula at every possible world, by treating propositional connectives in the usual way and using the accessibility relation to interpret the modalities.

Definition. Given a directed frame $F = \langle W, R, w_0 \rangle$, we define *local satisfaction* as a ternary relation $\Vdash$ over valuations over F, worlds (elements of W) and modal formulas as follows.

$$
\begin{aligned}
V \Vdash_w p &\text{ iff } V(w,p) = \top \\
V \Vdash_w \neg\varphi &\text{ iff } V \nVdash_w \varphi \\
V \Vdash_w \varphi \vee \psi &\text{ iff } V \Vdash_w \varphi \text{ or } V \Vdash_w \psi \\
V \Vdash_w \varphi \wedge \psi &\text{ iff } V \Vdash_w \varphi \text{ and } V \Vdash_w \psi \\
V \Vdash_w \varphi \to \psi &\text{ iff } V \nVdash_w \varphi \text{ or } V \Vdash_w \psi \\
V \Vdash_w \Box\varphi &\text{ iff } V \Vdash_{w'} \varphi \text{ for all } w' \text{ such that } wRw' \\
V \Vdash_w \Diamond\varphi &\text{ iff } V \Vdash_{w'} \varphi \text{ for some } w' \text{ such that } wRw'
\end{aligned}
$$

Example. We now show some examples of local satisfaction in the context of the frame F_0 and valuation V_0 presented earlier.

- $V_0 \Vdash_{w_1} p \wedge q$, since $V_0 \Vdash_{w_1} p$ and $V_0 \Vdash_{w_1} q$.
- $V_0 \Vdash_{w_1} \Box p$, since w_1 can see w_2 and w_3, and we have that both $V_0 \Vdash_{w_2} p$ and $V_0 \Vdash_{w_3} p$.
- $V_0 \Vdash_{w_1} \Diamond q$, since $w_1 R w_3$ and $V_0 \Vdash_{w_3} q$. The fact that $w_1 R w_2$ is irrelevant, since we only need *one* world accessible from w_1 to make q true.
- $V_0 \Vdash_{w_1} \Diamond\neg q$, since $w_1 R w_2$ and $V_0 \Vdash_{w_2} q$.
- As a consequence of the two previous results, $V_0 \Vdash_{w_1} (\Diamond q) \wedge (\Diamond\neg q)$: modalities allow for modeling contradictions, thanks to the possibility of viewing different worlds.
- $V_0 \Vdash_{w_3} \Box q$, since the only world accessible from w_3 is w_4 and $V_0 \Vdash_{w_4} q$.
- We also have that $V_0 \Vdash_{w_2} \Box q$: since there are no worlds visible from w_2, the condition for satisfying the universal modality is trivially satisfied. In fact, we even have that $V_0 \Vdash_{w_2} \Box\bot$.
- Since $V_0 \Vdash_{w_2} \Box q$ and $V_0 \Vdash_{w_3} \Box q$, we can conclude that $V_0 \Vdash_{w_1} \Box\Box q$.
- $V_0 \Vdash_{w_1} \Diamond\Box\neg p$, since $w_1 R w_2$ and $V_0 \Vdash_{w_2} \Box\neg p$ (by a similar argument to the one presented above).

- $V_0 \nVdash_{w_2} \Diamond\bot$: since there is no world accessible from w_2, the condition for $V_0 \Vdash_{w_2} \Diamond\bot$ necessarily fails.
- $V_0 \nVdash_{w_1} \Box\Diamond q$, since $w_1 R w_2$ and $V_0 \nVdash_{w_2} \Diamond q$.
- However, $V_0 \Vdash_{w_1} \Diamond\Diamond q$: since $w_1 R w_3$ and $w_3 R w_4$, we can successively infer that $V_0 \Vdash_{w_3} \Diamond q$ and $V_0 \Vdash_{w_1} \Diamond\Diamond q$. ◁

Exercise 3. Using the semantics for the modalities and negation, show that, for any world w in a directed frame $F = \langle W, R, w_0\rangle$, any valuation V over F and any formula φ, the following hold:

(a) $V \nVdash_w \Box\varphi$ iff $V \nVdash_{w'} \varphi$ for some w' such that wRw';

(b) $V \nVdash_w \Diamond\varphi$ iff $V \nVdash_{w'} \varphi$ for all w' such that wRw'.

Definition. Formula φ is said to be *true* in the directed frame $F = \langle W, R, w_0\rangle$ under V, written $F \models_V \varphi$, if $V \Vdash_{w_0} \varphi$.

We say that F satisfies φ, written $F \models \varphi$, if $F \models_V \varphi$ for every valuation V over F.

The condition $F \models_V \varphi$ is also called *global satisfaction* by contrast to the notion defined earlier. The dichotomy between local and global satisfaction is more clear in the undirected semantics, where global satisfaction is defined as $V \Vdash_w \varphi$ for all $w \in W$.

Example. We can rephrase some of the results in the previous example by writing e.g. $F_0 \models_{V_0} p \wedge q$, $F_0 \models_{V_0} \Box p$ or $F_0 \models_{V_0} \Diamond\Box\neg p$. ◁

Example. Formula $\Box p \to p$ is true in the frame F_0 from the previous example. Indeed, let V be any valuation. Then

$$\begin{aligned} F_0 \models_V \Box p \to p &\text{ iff } V \Vdash_{w_1} \Box p \to p \\ &\text{iff } V \nVdash_{w_1} \Box p \text{ or } V \Vdash_{w_1} p \\ &\text{iff } (V \nVdash_{w'} p \text{ for some } w' \text{ s.t. } w_1 R w') \text{ or } V \Vdash_{w_1} p. \end{aligned}$$

Since $w_1 R w_1$, one of these conditions must always hold: if $V \Vdash_{w_1} p$, then the second condition is immediately true, whereas if $V \nVdash_{w_1} p$ it is the first condition that holds. Therefore $F_0 \models \Box p \to p$. ◁

Exercise 4. Show that:

(a) $F_0 \models q \to \Diamond q$ (b) $F_0 \models \Diamond\Box\bot$ (c) $F_0 \models \Box\Box p \to \Box p$

As with propositional logic, we can use the semantics to inspect a formula and try to build a model that satisfies or falsifies it, or prove that no such model exists. This task is a bit more

challenging than in the propositional case, though, since we need to reason about different worlds, and our models may need to be somewhat complex. Two very useful models in this context are the very simple frames consisting of a single (initial) world $\star$, and one of the two possible accessibility relations – $\emptyset$ or $\{\langle\star,\star\rangle\}$. Hereafter we denote these two frames by $F_\bot$ and $F_\top$, i.e.,

$$F_\bot = \overset{\star}{\circ} \qquad\qquad F_\top = \overset{\star}{\circlearrowleft}$$

These frames are very simple to analyse, and have some useful properties.

- If the visibility relation is empty, then any formula of the form $\Box\varphi$ is true, and any formula of the form $\Diamond\varphi$ is false. This helps in simplifying the analysis of complex formulas, as we never need to look inside modalities.

- If the only world sees itself, then $\Box\varphi$, $\Diamond\varphi$ and φ are all equivalent in that world. This allows us to ignore all modalities when analysing the formula.

Example. We illustrate these properties with some examples.

- Every formula of the form $\Box\varphi$ is satisfiable and every formula of the form $\Diamond\varphi$ is falsifiable, as exhibited by frame $F_\bot$.

- Formulas $p \to \Box p$, $(\Diamond p \to \Box\neg q) \to (\Box q \to \Box\neg p)$ and $\neg(\Diamond p \wedge \Diamond\neg p)$ are all satisfiable: by removing modalities they become $p \to p$, $(p \to \neg q) \to (q \to \neg p)$ and $\neg(p \wedge \neg p)$, respectively, which are all propositional tautologies,[2] and therefore they are all true in the frame $F_\top$. ◁

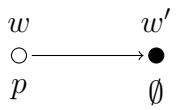
In general, we need to analyse the formula in more detail in order to get the necessary information to make a model that satisfies or falsifies it.

Example. Let us now try to falsify the three formulas given in the previous example.

- Let F be the directed frame we are trying to construct, with initial world w, and V be a valuation. Then

$$\begin{aligned} F \not\models_V p \to \Box p \text{ iff } & V \Vdash_w p \to \Box p \\ \text{iff } & V \Vdash_w p \text{ and } V \nVdash_w \Box p \\ \text{iff } & V \Vdash_w p \text{ and } V \nVdash_{w'} p \text{ for some } w' \text{ s.t. } wRw'. \end{aligned}$$

  Therefore, our model must have two worlds w and w', with wRw', and the valuation must be such that $V(p) = \{w\}$. (Note that w and w' must be distinct, since p is true at w and false at w'.) Graphically, we can represent this model as



  (We explicitly write $\emptyset$ to emphasize that no propositional symbol is true at w'.)

[2]Recall that *tautology* is another word for *valid formula*. When working in logics that extend propositional logic, it is customary to use the expression *propositional tautology* in order to avoid confusion with valid formulas in the extended logic.

- Consider now the formula $(\Diamond p \to \Box\neg q) \to (\neg\Box p \to \Box q)$. As before, we have that

$$\begin{aligned} F \not\models_V (\Diamond p \to \Box\neg q) \to (\Box q \to \neg\Box p) & \\ \text{iff } V \Vdash_w (\Diamond p \to \Box\neg q) \to (\Box q \to \neg\Box p) & \\ \text{iff } V \Vdash_w \Diamond p \to \Box\neg q \text{ and } V \nVdash_w \Box q \to \neg\Box p & \\ \text{iff } (V \nVdash_w \Diamond p \text{ or } V \Vdash_w \Box\neg q) \text{ and } V \Vdash_w \Box q \text{ and } V \nVdash_w \neg\Box p & \\ \text{iff } (\underbrace{V \nVdash_w \Diamond p}_{(1)} \text{ or } \underbrace{V \Vdash_w \Box\neg q}_{(2)}) \text{ and } \underbrace{V \Vdash_w \Box q}_{(3)} \text{ and } \underbrace{V \Vdash_w \Box p}_{(4)}. & \end{aligned}$$

  Since all these properties express conditions about all worlds visible from w, they are trivially satisfied if the visibility relation is empty. Therefore, the trivial model $F_\bot$ discussed above makes this formula false for every valuation V.

- Now we look at the formula $\neg(\Diamond p \wedge \Diamond\neg p)$. Reasoning as in the previous cases, we have that

$$\begin{aligned} F \not\models_V \neg(\Diamond p \wedge \Diamond\neg p) \text{ iff } V \nVdash_w \neg(\Diamond p \wedge \Diamond\neg p) & \\ \text{iff } V \Vdash_w \Diamond p \wedge \Diamond\neg p & \\ \text{iff } \underbrace{V \Vdash_w \Diamond p}_{(1)} \text{ and } \underbrace{V \Vdash_w \Diamond\neg p}_{(2)}. & \end{aligned}$$

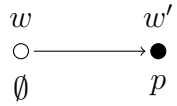
  Condition (1) requires that there be a world w_1 such that wRw_1 and $V \Vdash_{w_1} p$. Likewise, condition (2) demands a world w_2 such that wRw_2 and $V \nVdash_{w_2} p$. Worlds w_1 and w_2 must be distinct, since they differ on the semantics of p, but one of them can be the same as w. So both the following models make this formula false.

These examples show that the modalities *do* influence the semantics of the formulas: even though they all become propositional tautologies when the modalities are removed, they can be falsified by tweaking the accessibility relation in the right way. ◁

Example. Let us now consider the formula $\Diamond(p \wedge \Box\Box\neg p)$. Since this formula is of the form $\Diamond\varphi$, we know that it is false in frame $F_\bot$, so it is falsifiable. In order to decide whether it is also satisfiable, we consider a frame F with initial world w and a valuation V; then

$$\begin{aligned} F \models_V \Diamond(p \wedge \Box\Box\neg p) \text{ iff } V \Vdash_w \Diamond(p \wedge \Box\Box\neg p) & \\ \text{iff } V \Vdash_{w'} p \wedge \Box\Box\neg p \text{ for some } w' \text{ such that } wRw' & \\ \text{iff } V \Vdash_{w'} p \text{ and } V \Vdash_{w'} \Box\Box\neg p & \end{aligned}$$

The first condition tells us simply that p must hold at w'. In order for the second condition to hold, we must make sure that $\Box\neg p$ holds at every world accessible from w'; the easiest way to achieve this is to make sure that w' does not see any other world. We are thus led to the model



which indeed makes this formula true. ◁

Exercise 5. For each of the following formulas, find a model that satisfies them and a model that falsifies them, or prove that none exist.

(a) $\Box\Box p \to \Box\Diamond p$ (b) $(\Diamond p \wedge \Box q) \to \Diamond(p \to q)$ (c) $\Diamond(p \vee q) \wedge \Box p \wedge \Box(\neg q)$

As in any logic, in modal logic we can also say that a formula is valid if $F \models \varphi$ for every frame F. However, this turns out not to be a very useful notion of validity due to the flexibility that is needed when interpreting modalities in different contexts. Instead, we consider the notion of validity relative to a class of frames characterized by properties of the accessibility relation – as it is these properties that give the modalities their intended semantics.

Definition. A *regular class of frames* is a class of frames defined by a property P of the accessibility relation, i.e., the class of all frames $\langle W, R, w_0\rangle$ for which $P(R)$ holds.

Intuitively, $\mathcal{F}$ is a regular class of frames if changing the initial world does not change membership in $\mathcal{F}$.

Definition. Let $\mathcal{F}$ be a regular class of frames. A formula φ is *valid over* $\mathcal{F}$, written $\models_{\mathcal{F}} \varphi$ or $\mathcal{F} \models \varphi$, if $F \models \varphi$ for every $F \in \mathcal{F}$.

In particular, $\models \varphi$ if φ is valid over the class of all frames – but, as explained above, this notion is of limited interest.

Since changing the initial world of a frame does not affect membership in $\mathcal{F}$, validity of φ over $\mathcal{F}$ implies that φ holds at every world of every frame in $\mathcal{F}$.

Example. We claimed at the beginning of this section that our semantics makes axiom K true by construction. In other words, axiom K is valid in the class of all frames. We now show that this claim holds.

Let $F = \langle W, R, w_0\rangle$ be a frame and V be a valuation such that V falsifies some instance of K, i.e., $V \not\Vdash_{w_0} \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$ for some formulas φ and ψ.

From the semantics of modal logic,

$$V \not\Vdash_{w_0} \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi) \text{ iff } V \Vdash_{w_0} \Box(\varphi \to \psi) \text{ and } V \not\Vdash_{w_0} \Box\varphi \to \Box\psi$$
$$\text{iff } \underbrace{V \Vdash_{w_0} \Box(\varphi \to \psi)}_{(1)} \text{ and } \underbrace{V \Vdash_{w_0} \Box\varphi}_{(2)} \text{ and } \underbrace{V \not\Vdash_{w_0} \Box\psi}_{(3)}.$$

From (3), we know that there must exist a world $w \in W$ such that $w_0 R w$ and $V \not\Vdash_w \psi$. Since $w_0 R w$, we also obtain $V \Vdash_w \varphi \to \psi$ from (1) and $V \Vdash_w \varphi$ from (2). But these two conditions imply that $V \Vdash_w \psi$, yielding a contradiction. ◁

Exercise 6. Show that the duality rules and the de Morgan rules for the modalities hold, i.e., that the following formulas are valid for every formula φ.

(a) $\Box\varphi \leftrightarrow \neg\Diamond\neg\varphi$ (b) $\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$ (c) $\Box\neg\varphi \leftrightarrow \neg\Diamond\varphi$ (d) $\Diamond\neg\varphi \leftrightarrow \neg\Box\varphi$

We now show some examples of more restricted classes of frames.

Example. Axiom T is valid in the class of reflexive frames, i.e. the class of frames where wRw for all $w \in W$. Indeed, let $\langle W, R, w_0\rangle$ be a reflexive frame, V be a valuation, and φ be a formula. Suppose that $V \Vdash_{w_0} \Box\varphi$. Since w_0Rw_0, necessarily $V \Vdash_{w_0} \varphi$. Therefore $V \Vdash_{w_0} \Box\varphi \to \varphi$. Since φ is arbitrary, this means that T is valid in this class. ◁

These properties are usually phrased as (relative) soundness properties.

Definition. A regular class of frames $\mathcal{F}$ is said to be *sound* for an axiom A, written $\mathcal{F} \models A$, if all instances of A are valid in every frame in $\mathcal{F}$.

We can thus rephrase the previous results in the following lemma.

Lemma 17. The class of all frames is sound for axiom K, and the class of reflexive frames is sound for axiom T.

Exercise 7. Let $\mathcal{F}$ be the class of all symmetric frames (i.e., such that if wRw', then $w'Rw$). Show that $\mathcal{F} \models B$.

Exercise 8. Let $\mathcal{F}$ be the class of all transitive frames (i.e., such that if w_1Rw_2 and w_2Rw_3, then w_1Rw_3). Show that $\mathcal{F} \models 4$.

Conversely, we can define relative completeness of a regular class of frames with respect to modal axioms.

Definition. A regular class of frames $\mathcal{F}$ is said to be *complete* for an axiom A if there is no regular class of frames strictly larger than $\mathcal{F}$ where all instances of A are valid.

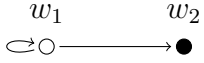
Regularity is of key importance in this definition: if $\mathcal{F}$ is complete for A, there may still be directed frames not in $\mathcal{F}$ where all instances of φ hold. This happens because validity in a frame only considers its initial world – so there will be at least one world in that frame where at least one instance of the axiom fails – while completeness of $\mathcal{F}$ implies that all instances of A are true in every world of every frame in $\mathcal{F}$.

Lemma 18. The class of reflexive frames is complete for axiom T.

Proof. Let $\langle W, R\rangle$ be a frame and assume that R is not reflexive. Then there exists a world $w \in W$ such that $w \not R w$.

Consider the directed frame $\langle W, R, w\rangle$, and define a valuation V such that $V \nVdash_w p$ and $V \Vdash_{w'} p$ for all $w' \neq w$. Since $w \not R w$, $V \Vdash_w \Box p$; but $V \nVdash_w p$, hence $V \nVdash_w \Box p \to p$. □

Example. Consider the following directed frame F.

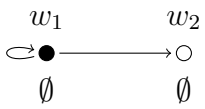


This frame is not reflexive, since $w_2 \not R w_2$. However, for any valuation V and formula φ it is the case that $V \Vdash_{w_1} \Box\varphi \to \varphi$:

- if $V \Vdash_{w_1} \varphi$, then the consequent of the implication immediately holds;
- if $V \nVdash_{w_1} \varphi$, then $V \nVdash_{w_1} \Box\varphi$ (since $w_1 R w_1$), and therefore the antecedent of the implication does not hold.

Therefore, $F \models T$.

However, any regular class of frames containing F must also contain the frame F' where the initial world is w_2 instead of w_1. The valuation V' defined by



shows that T is not valid in F': for the instance $\Box p \to p$ we have that $V' \Vdash_{w_2} \Box p$ vacuously (since w_2 does not see any world), while $V' \nVdash_{w_2} p$. Therefore $V' \nVdash_{w_2} \Box p \to p$, and thus $F' \not\models_{V'} \Box p \to p$, whence $F' \not\models T$. ◁

Exercise 9. Let $\mathcal{F}$ be the class of all symmetric frames. Show that $\mathcal{F}$ is also complete for B.

Exercise 10. Let $\mathcal{F}$ be the class of all transitive frames. Show that $\mathcal{F}$ is also complete for 4.

Exercise 11. Let $\mathcal{F}$ be the class of all Euclidean frames (i.e., such that if $w_1 R w_2$ and $w_1 R w_3$, then $w_2 R w_3$). Show that $\mathcal{F}$ is sound and complete for 5.

The last lemma in this section gives some insight on why we focus on regular classes of frames: in these classes, $\Box$ internalizes validity in the sense of the following result.

Lemma 19. Let $\mathcal{F}$ be a regular class of frames and φ be a formula. If $\mathcal{F} \models \varphi$, then $\mathcal{F} \models \Box\varphi$.

Proof. We prove this result by counterreciprocal: we assume that $\mathcal{F} \not\models \Box\varphi$, and show that then $\mathcal{F} \not\models \varphi$.

Let $F = \langle W, R, w_0 \rangle \in \mathcal{F}$ be a directed frame, and let V be a valuation such that $V \nVdash_{w_0} \Box\varphi$. Then there exists $w \in W$ such that $w_0 R w$ and $V \nVdash_w \varphi$.

Take F' to be the directed frame $\langle W, R, w \rangle$, differing from F only in the choice of initial world. Since $\mathcal{F}$ is a regular class of frames, $F' \in \mathcal{F}$ (since the accessibility relations in F and F' coincide). Furthermore, since w is the initial world in F' and $V \nVdash_w \varphi$, we conclude that $F' \not\models_V \varphi$, whence $F' \not\models \varphi$. This shows that $\mathcal{F} \not\models \varphi$. □

Exercise 12. As discussed in the text, an alternative way to define the global semantics of

modal logic is by considering only undirected frames, and saying that $\langle W, R\rangle \models_V \varphi$ iff $V \Vdash_w \varphi$ for every $w \in W$.

Using this alternative definition, the results proved in the text hold for arbitrary classes of frames (i.e., not only for regular classes of frames). In particular, show that:

(a) if F is not a reflexive frame, then $F \not\models T$;

(b) for any class of frames $\mathcal{F}$ and formula φ, if $\mathcal{F} \models \varphi$, then $\mathcal{F} \models \Box\varphi$.

We finish this section with a property that will be useful in later sections.

Lemma 20. Let φ be a propositional tautology, $p_1, \ldots, p_n$ be distinct propositional symbols and $\psi_1, \ldots, \psi_n$ be modal formulas (not necessarily distinct).

Let φ' be the formula obtained from φ by uniformly replacing each propositional symbol p_i with the formula ψ_i. Then $V \Vdash_w \varphi'$ for valuation V over a frame F and every world w of F.

Proof. Let F be a frame, V be a valuation over F, and w be a world in F. Define a propositional valuation V' as follows.

- $V'(p_i) = \top$ iff $V \Vdash_w \psi_i$
- $V'(p) = V(w, p)$ for $p \notin \{p_1, \ldots, p_n\}$

We prove that $V \Vdash_w \varphi'$ iff $V' \models \varphi$ by structural induction on φ.

If φ is a propositional symbol, then either it is one of the p_i, and then this property holds by construction; or it is not, and then φ' coincides with φ and the property also holds by definition of V'.

Suppose that φ is $\varphi_1 \vee \varphi_2$. Then φ' is also of the form $\varphi'_1 \vee \varphi'_2$, where each φ'_j is obtained from φ_j by the same uniform replacement. Thus, by induction hypothesis, $V \Vdash_w \varphi'_j$ iff $V' \models \varphi_j$; since the local semantics of disjunction at w coincides with the semantics of disjunction in propositional logic, this implies that $V \Vdash_w \varphi'$ iff $V' \models \varphi$. The case for the remaining propositional connectives is similar.

Since φ is valid, this implies that $V \Vdash_w \varphi'$. But V and w are arbitrary, therefore the thesis holds. □

3.3 Duality

Duality is an important concept in many logics, describing pairs of connectives that behave symmetrically in a precisely defined sense. The simplest example of dual connectives is that of $\vee$ and $\wedge$ in propositional logic. The semantics of these connectives is symmetric in the sense that the truth values are interchanged in their definition: $\varphi \wedge \psi$ is *true* if both φ and ψ are *true*, while $\varphi \vee \psi$ is *false* if both φ and ψ are *false*.

This translates into a number of symmetric rules in all deductive systems:

- In the tableaux calculus for propositional logic, rules ($\wedge$) and ($\neg\vee$) are very similar, as are ($\vee$) and ($\neg\wedge$) – the rules for the negated connectives are obtained from the rule for the dual connective by prepending a negation to every formula.

- In the sequent calculus for propositional logic, rules ($\wedge R$) and ($\vee L$) are very similar, as are $\vee\mathrm{R}_1$ and $\wedge\mathrm{L}_1$ and $\vee\mathrm{R}_2$ and $\wedge\mathrm{L}_2$ – the left rule for each connective is the same as the right rule for the dual connective with the two components of the sequent swapped.

  Likewise, the admissible rules $\vee$R and $\wedge$L are also related to each other in the same way – and the proofs of admissibility can also be transformed into each other by flipping all sequents around, see the proof of Lemma 6.

- If we extend the Hilbert calculus for propositional logic so that conjunction and disjuction are taken as primitive connectives (which we did not discuss), they can be axiomatized as follows.

$$\begin{array}{ll}
\varphi \to \psi \to (\varphi \wedge \psi) & (\mathrm{Ax.}\wedge\mathrm{I}) \\
(\varphi \wedge \psi) \to \varphi & (\mathrm{Ax.}\wedge\mathrm{E}_1) \\
(\varphi \wedge \psi) \to \psi & (\mathrm{Ax.}\wedge\mathrm{E}_2) \\
\varphi \to (\varphi \vee \psi) & (\mathrm{Ax.}\vee\mathrm{I}_1) \\
\psi \to (\varphi \vee \psi) & (\mathrm{Ax.}\vee\mathrm{I}_2) \\
(\varphi \to \gamma) \to ((\psi \to \gamma) \to ((\varphi \vee \psi) \to \gamma)) & (\mathrm{Ax.}\vee\mathrm{E})
\end{array}$$

  Here, the symmetry between ($\mathrm{Ax.}\wedge\mathrm{E}_1$) and ($\mathrm{Ax.}\vee\mathrm{I}_1$), as well as that between ($\mathrm{Ax.}\wedge\mathrm{E}_2$) ($\mathrm{Ax.}\vee\mathrm{I}_2$), is also clear. The relation between ($\mathrm{Ax.}\wedge\mathrm{I}$) and ($\mathrm{Ax.}\vee\mathrm{E}$) is not so obvious at first, but it is also there: axiom ($\mathrm{Ax.}\wedge\mathrm{I}$) is equivalent to the more complex formula

$$(\gamma \to \varphi) \to ((\gamma \to \psi) \to (\gamma \to (\gamma \wedge \psi)))\,,$$

  but the version presented is preferred since it does not involve any formulas other than the conjuncts. Such a simplification is not possible for axiom ($\mathrm{Ax.}\vee\mathrm{E}$).

Duality is closely related to negation, and the aspect of the duality between $\wedge$ and $\vee$ most commonly stated are the de Morgan laws

$$\neg(\varphi \vee \psi) \leftrightarrow \neg\varphi \wedge \neg\psi \text{ and } \neg(\varphi \wedge \psi) \leftrightarrow \neg\varphi \vee \neg\psi\,,$$

which are another consequence of the relationship between the semantics of these connectives. However, duality does not require the existence of negation – and indeed, one of the applications of duality is that it can be used to internalize some restricted forms of negation in logics that do not have this connective.

In modal logic, the modalities $\Box$ and $\Diamond$ are also dual. Indeed, the semantics for these connectives are symmetric in the sense stated above for $\wedge$ and $\vee$: $\Box\varphi$ is *true* at world w if φ is *true* at every world accessible from w, while $\Diamond\varphi$ is *false* at world w if φ is *false* at every world accessible from w.[3] As we saw in an earlier exercise, the de Morgan laws for these connectives also hold, in the sense that the formulas $\Box\neg\varphi \leftrightarrow \neg\Diamond\varphi$ and $\Diamond\neg\varphi \leftrightarrow \neg\Box\varphi$ are valid.

The notion of duality can be extended to the whole language of modal logic by defining it as an operator on formulas, motivated by the de Morgan laws. In order to treat the other connectives, we observe that the semantics of negation makes it its own dual, while implication can be seen as an abbreviation ($\varphi \to \psi$ is equivalent to $\neg\varphi \vee \psi$).

[3]Or, dually, $\Diamond\varphi$ is *true* at world w if φ is *true* at some world accessible from w, while $\Box\varphi$ is *false* at world w if φ is *false* at some world accessible from w: the notions of "every" and "some" are also dual at the metalevel on which we analyze modal logic.

Definition. The *dual* of a formula is defined inductively as follows.

$$p^* = \neg p \qquad (\varphi \vee \psi)^* = \varphi^* \wedge \psi^* \qquad (\Box\varphi)^* = \Diamond\varphi^* \qquad (\varphi \to \psi)^* = \neg\varphi^* \wedge \psi^*$$
$$(\neg\varphi)^* = \neg\varphi^* \qquad (\varphi \wedge \psi)^* = \varphi^* \vee \psi^* \qquad (\Diamond\varphi)^* = \Box\varphi^*$$

Example. We show how to compute the duals of some of the formulas we met earlier.

- Consider the formula $\neg(\Diamond p \wedge \Diamond\neg p)$. Its dual is

$$\begin{aligned}(\neg(\Diamond p \wedge \Diamond\neg p))^* &= \neg(\Diamond p \wedge \Diamond\neg p)^* \\ &= \neg((\Diamond p)^* \vee (\Diamond\neg p)^*) \\ &= \neg(\Box p^* \vee \Box(\neg p)^*) \\ &= \neg(\Box p^* \vee \Box\neg p^*) \\ &= \neg(\Box\neg p \vee \Box\neg\neg p)\end{aligned}$$

  It is common to simplify the last formula to $\neg(\Box\neg p \vee \Box p)$ by adding the rule $(\neg p)^* = p$ to the definition of modality.

- Similarly, if we consider the formula $\Diamond(p \wedge \Box\Box\neg p)$, we obtain

$$\begin{aligned}(\Diamond(p \wedge \Box\Box\neg p))^* &= \Box(p \wedge \Box\Box\neg p)^* \\ &= \Box\,(p^* \vee (\Box\Box\neg p)^*) \\ &= \Box\,(\neg p \vee \Diamond(\Box\neg p)^*) \\ &= \Box\,(\neg p \vee \Diamond\Diamond(\neg p)^*) \\ &= \Box\,(\neg p \vee \Diamond\Diamond p)\end{aligned}$$

  applying the same simplification as before in the last step.

- Finally, consider an arbitrary instance $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$ of the modal axiom K. Its dual is

$$\begin{aligned}(\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi))^* &= \neg(\Box(\varphi \to \psi))^* \wedge (\Box\varphi \to \Box\psi)^* \\ &= \neg\Diamond(\varphi \to \psi)^* \wedge (\neg(\Box\varphi)^* \wedge (\Box\psi)^*) \\ &= \neg\Diamond(\neg\varphi^* \wedge \psi^*) \wedge \neg\Diamond\varphi^* \wedge \Diamond\psi^*\,.\end{aligned}$$

  Observe that this formula is a contradiction: in order for it to be valid at a world w, the subformula $\Diamond\psi^*$ must be true at w, that is, there is a world w' accessible from w where ψ^* is true. Furthermore, since w also satisfies $\neg\Diamond\varphi^*$, we can also conclude that φ^* is false at w'. But then w' satisfies $\neg\varphi^* \wedge \psi^*$, and therefore w satisfies $\Diamond(\neg\varphi^* \wedge \psi^*)$ – contradicting the first conjunct. ◁

Exercise 13. Write the duals of the formulas in Exercise 5.

The correspondence between duality and negation is formalized in the following result, which states that φ^* is equivalent to the negation of φ.

Lemma 21. For any frame $\langle W, R\rangle$, world $w \in W$, valuation V and formula φ, $V \Vdash_w \varphi^*$ iff $V \nVdash_w \varphi$.

Proof. By structural induction on φ. We detail a few cases.

If φ is a propositional symbol p, then $\varphi^* = \neg p$ and the thesis becomes simpy $V \Vdash_w \neg p$ iff $V \nVdash_w p$.

Assume φ is $\neg\psi$, and therefore $\varphi^* = \neg\psi^*$. Then $V \Vdash_w \neg\psi^*$ iff $V \nVdash_w \psi^*$ iff $V \Vdash_w \psi$ (induction hypothesis) iff $V \nVdash_w \neg\psi$.

Assume now that φ is $\Box\psi$, so $\varphi^* = \Diamond\psi^*$. Then $V \Vdash_w \Diamond\psi^*$ iff there is a world w' such that wRw' and $V \Vdash_{w'} \psi^*$. By induction hypothesis, the latter condition holds iff $V \nVdash_{w'} \psi$. But the condition that there exists a world w' such that wRw' and $V \nVdash_{w'} \psi$ is equivalent to $V \nVdash_w \Box\psi$. □

We already observed this behaviour in the previous example: every instance of axiom K is valid in every world of every frame, and any instance of its dual is a contradiction.

Example. Consider the formula $\neg(\Diamond p \wedge \Diamond\neg p)$ and its dual $\neg(\Box\neg p \vee \Box p)$. We previously saw that $\neg(\Diamond p \wedge \Diamond\neg p)$ is both satisfiable and falsifiable; therefore, by Lemma 21, $\neg(\Box\neg p \vee \Box p)$ is both falsifiable and satisfiable.

We can refine this analysis a bit more. In order for $\neg(\Diamond p \wedge \Diamond\neg p)$ to be true at the initial world w of a directed frame $\langle W, R, w\rangle$ given a valuation V, it must be the case that either $V \nVdash_w \Diamond p$ or $V \nVdash_w \Diamond\neg p$; this means that $V(p)$ must be the same in all worlds accessible from p – and therefore there must be at most one such world. Indeed, this formula characterizes the class of frames whose accessibility relation is *functional* (if wRw_1 and wRw_2, then $w_1 = w_2$).

Lemma 21 then implies that the dual formula $\neg(\Box\neg p \vee \Box p)$ characterizes the class of frames where every world satisfies the dual condition, namely those frames where every world sees at least two different worlds.[4] We can check this directly: if wRw' for only one world w', then necessarily one of $\Box p$ or $\Box\neg p$ will be true at w (corresponding to whether p is true or false at w', respectively). ◁

Exercise 14. For each of the formulas in Exercise 5, directly check using the models that you build that the following all hold.

(a) If the original formula is satisfiable, then its dual is falsifiable.

(b) If the original formula is falsifiable, then its dual is satisfiable.

Duality has a number of interesting properties, too many to enumerate exhaustively in this setting. We do however present one more, because it is specific to modal logic and has some interesting applications.

Theorem 17 (Duality)**.** For any class of frames $\mathcal{F}$ and formulas φ and ψ, $\models_{\mathcal{F}} \varphi \to \psi$ iff $\models_{\mathcal{F}} \psi^* \to \varphi^*$.

Proof. The formulas $\varphi \to \psi$ and $\neg\psi \to \neg\varphi$ are equivalent. By Lemma 21, $\neg\psi$ is also equivalent to ψ^*, and $\neg\varphi$ is equivalent to φ^*. Therefore $\varphi \to \psi$ is logically equivalent to $\psi^* \to \varphi^*$, and the thesis follows. □

[4] Not the complementary class of frames to the previous one – this would be the class of frames where *some* world sees two worlds, but since Lemma 21 regards local satisfaction and not global satisfaction we need to negate the condition at each world.

Since all axioms of modal logic are phrased as implications and duality exchanges modalities, we can use this result to rewrite them in an equivalent form based on possibility, rather than necessity.

Example. Consider axiom T, which has the form $\varphi \to \Box\varphi$. By the Duality Theorem, this formula is equivalent to $(\Box\varphi)^* \to \varphi^*$ for any formula φ, or simply to $\Diamond\varphi^* \to \varphi^*$. Since φ can be instantiated to any formula, we can ignore the innermost occurrences of * in the last formula and simplify it to $\Diamond\varphi \to \varphi$. We call this axiom $T\Diamond$.

Note that this is *not* the dual of axiom T: that would be the formula $(\varphi \to \Box\varphi)^*$, which reduces to $\neg\varphi^* \wedge \Diamond\varphi*$; we call this formula T^*. By Lemma 21, this formula T^* is actually a contradiction in any model of T. ◁

Sometimes the "diamond" versions of modal axioms are easier to interpret, making it more clear whether a particular formalization of a modality should include that axiom or not.

Exercise 15. Show that the Duality Theorem yields the following "diamond" versions of axioms B, 4 and 5

$$B\Diamond:\ \Diamond\Box\varphi \to \varphi \qquad 4\Diamond:\ \Diamond\Diamond\varphi \to \Diamond\varphi \qquad 5\Diamond:\ \Diamond\Box\varphi \to \Box\varphi$$

and that $D\Diamond$ is exactly the same as D.

Exercise 16. Let $\mathcal{F}$ be the class of all symmetric frames. Show directly that $\mathcal{F}$ is sound and complete for $B\Diamond$.

Dualizing K yields directly the formula $\neg(\Diamond\psi \to \Diamond\varphi) \to \Diamond\neg(\psi \to \varphi)$, which is not very readable and not very interesting. By using propositional reasoning and the de Morgan laws for the modalities, it can be shown to be equivalent to the more meaningful

$$K\Diamond:\ \Box(\psi \to \varphi) \to (\Diamond\psi \to \Diamond\varphi) \qquad \text{and} \qquad K\wedge:\ \Box\varphi \wedge \Diamond\psi \to \Diamond(\varphi \wedge \psi)$$

while the original axiom can also be formulated as

$$K\vee:\ \Box(\varphi \vee \psi) \to \Diamond\varphi \vee \Box\psi\,.$$

Observe that $K\wedge$ and $K\vee$ are syntactically asymmetric, but since conjunction and disjunction are both symmetric one has the freedom to decide which subformula should be necessary and which should be possible.

Exercise 17.

(a) Check that $K\vee$ is equivalent to K.

(b) Check that both $K\Diamond$ and $K\wedge$ are equivalent to $\neg(\Diamond\psi \to \Diamond\varphi) \to \Diamond\neg(\psi \to \varphi)$.

3.4 Hilbert calculus

The first deductive system that we discuss for modal logic is the Hilbert calculus, which is a generalization of the Hilbert calculus for propositional logic. The results we previously showed for the Hilbert calculus for propositional logic – in particular, the completeness of that calculus – make its counterpart for modal logic much easier to use, as we will shortly see. Furthermore, the fact that this system is by nature based on axioms makes it much easier to work with different modalities and tune the particular axioms we are interested in for each situation without having to reanalyse the properties of the calculus.

We consider only negation ($\neg$), implication ($\to$) and necessity ($\Box$) as primitive connectives, and all remaining ones as defined by abbreviation. In particular, $\Diamond\varphi$ stands for $\neg\Box\neg\varphi$; we will use these two formulas interchangeably during proofs.

The Hilbert calculus for modal logic includes the following four axioms.

$$\varphi \to (\psi \to \varphi) \qquad \text{(Ax.1)}$$
$$(\varphi \to (\psi \to \gamma)) \to ((\varphi \to \psi) \to (\varphi \to \gamma)) \qquad \text{(Ax.2)}$$
$$(\neg\psi \to \neg\varphi) \to (\varphi \to \psi) \qquad \text{(Ax.3)}$$
$$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi) \qquad K$$

There are the axioms for propositional logic, together with the modal axiom K. Formulas φ, ψ and γ can be instantiated by any modal formula, meaning that there are now more instances of axioms (Ax.1–3) than before: for example, $\Box p \to (\Diamond q \to \Box p)$ is an instance of (Ax.1) in this calculus.

As for inference rules, we now have two rules:

$$\text{from } \varphi \text{ and } \varphi \to \psi \text{ infer } \psi \qquad \text{(MP)}$$
$$\text{from } \varphi \text{ infer } \Box\varphi \qquad \text{(Nec)}$$

where (MP) is Modus Ponens, as in propositional logic, and (Nec) is a new inference rule called *Necessitation*.

Derivations are defined as before, and we write $\Gamma \vdash_L \varphi$ to denote that there exists a derivation of φ using hypotheses from Γ.

Example. Using K and (Nec), we can "lift" all valid propositional implications to the modal level, by turning them into implications between necessary formulas. In other words, from $\varphi \to \psi$ we can derive $\Box\varphi \to \Box\psi$.

For example, the following derivation shows that $\vdash_L \Box p \to \Box(q \to p)$.

1. $p \to (q \to p)$ (Ax.1)
2. $\Box(p \to (q \to p))$ Nec(1)
3. $\Box(p \to (q \to p)) \to (\Box p \to \Box(q \to p))$ K
4. $\Box p \to \Box(q \to p)$ MP(3,2)

This technique is extremely useful to find proofs in the Hilbert calculus for modal logic, as we will see in later examples. ◁

Every formula provable in the Hilbert calculus with no hypotheses is valid, just as in the propositional case.

Theorem 18 (Weak soundness)**.** For any modal formula φ, if $\vdash_L \varphi$, then $\models \varphi$.

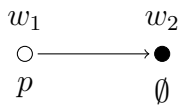
Proof. By induction on the proof of $\Gamma \vdash_L \varphi$. If φ is an instance of one of the propositional axioms, then this is a particular case of Lemma 20. If φ is an instance of K, then the thesis follows by Lemma 17. If φ is obtained by (MP), then the proof is analogous to the corresponding step of the proof for the propositional case. Finally, if φ is obtained by (Nec), then the thesis follows from Lemma 19, since the class of all frames is regular. □

However, when hypotheses are present the situation is different, and it is not the case that $\Gamma \vdash_L \varphi$ implies that $\Gamma \models \varphi$.

Example. Consider the judgement $\{p\} \vdash_L \Box p$. This judgement is valid, since from p we can infer $\Box p$ with a two-step derivation (where (Nec) is applied to the hypothesis p).

1. p	(Hyp)
2. $\Box p$	Nec(1)

However, it is not the case that $\{p\} \models \Box p$, as the following example shows.



This frame satisfies p (since it is true in the initial world), but not $\Box p$ (since $w_1 R w_2$ and p is false at w_2). Therefore it is not always the case that $\Gamma \vdash_L \varphi$ implies $\Gamma \models \varphi$. ◁

This example exhibits the fundamental problem with extending the proof of soundness to reasoning with hypotheses: the necessitation rule effectively propagates the hypotheses from the initial world to all worlds accessible from it. Inspection of the proof of weak soundness also confirms this analysis: in the case where (Nec) is applied in a derivation, the proof uses the fact that the class of all frames is regular. However, it does not even make sense to discuss the class of frames that satisfy p, as this is a property of the valuation and not of the frame.

Later in this section we will discuss a stronger soundness result for the Hilbert calculus that allows hypotheses to be used with some restrictions.

Many other properties of the Hilbert calculus for propositional logic carry over to modal logic – namely the ones whose proofs systematically transform the derivation in a way that is independent of the axioms and rules available, i.e., Hypothetical Reasoning and the Replacement Theorem. In the latter, the formulas being replaced can be modal formulas. This is not the case for the Deduction Theorem, though, since its proof uses a transformation that depends on the axiom or rule being applied at each step. Before discussing this result, we prove another very useful property that substantially simplifies the task of finding proofs in this calculus.

Theorem 19 (Propositional Reasoning)**.** Let φ be a propositional tautology, $p_1, \ldots, p_n$ be distinct propositional symbols and $\psi_1, \ldots, \psi_n$ be modal formulas (not necessarily distinct).

Let φ' be the formula obtained from φ by uniformly replacing each propositional symbol p_i with the formula ψ_i. Then $\vdash_L \varphi'$.

Proof. Let φ be a propositional tautology. Since the Hilbert calculus for propositional logic is complete, we know that φ is provable in that calculus; then also $\vdash_L \varphi$ in the calculus for

modal logic, since the latter includes all axioms and inference rules from the former. Applying the Replacement Theorem to this proof yields a proof of $\vdash_L \varphi'$. □

Thanks to this result, we can avoid using axioms (Ax.1–3) explicitly in proofs; more importantly, it makes it much easier to reason about derived propositional connectives, since we can write any propositional tautology involving them. We denote applications of this result in derivations by (Prop).

We illustrate how this technique works in practice with some examples.

Example. We start by showing that $\vdash_L \Box p \to \Box(p \vee q)$. The derivation is very similar to the previous example. We start by observing that $p \to p \vee q$ is a propositional tautology; afterwards, we lift this implication to the modal level using K and (Nec).

1. $p \to (p \vee q)$	Prop
2. $\Box(p \to (p \vee q))$	Nec(1)
3. $\Box(p \to (p \vee q)) \to (\Box p \to \Box(p \vee q))$	K
4. $\Box p \to \Box(p \vee q)$	MP(3,2)

This shows that indeed $\vdash_L \Box p \to \Box(p \vee q)$. ◁

Example. A similar example, but where we need to instantiate a propositional tautology with non-propositional formulas, is proving that $\vdash_L \Box(\Diamond p \wedge (q \to \Box q)) \to \Box\Diamond p$. This formula can be obtained by applying (Nec) to $(\Diamond p \wedge (q \to \Box q)) \to \Diamond p$ and combining the result with the appropriate instance of K. In turn, $(\Diamond p \wedge (q \to \Box q)) \to \Diamond p$ is an instance of the propositional tautology $p_1 \wedge p_2 \to p_1$, as can be seen by considering the mapping $p_1 \mapsto \Diamond p$, $p_2 \mapsto q \to \Box q$.

1. $(\Diamond p \wedge (q \to \Box q)) \to \Diamond p$	Prop
2. $\Box((\Diamond p \wedge (q \to \Box q)) \to \Diamond p)$	Nec(1)
3. $\Box((\Diamond p \wedge (q \to \Box q)) \to \Diamond p) \to (\Box(\Diamond p \wedge (q \to \Box q)) \to \Box\Diamond p)$	K
4. $\Box(\Diamond p \wedge (q \to \Box q)) \to \Box\Diamond p$	MP(3,2)

This illustrates that Propositional Reasoning can also be used with formulas that are not propositional formulas. ◁

Exercise 18. Find proofs of the following formulas in the Hilbert calculus for modal logic.

(a) $\Box(p \to \neg q) \to \Box(q \to \neg p)$

(b) $\Box(\Diamond p \wedge \Diamond q) \to \Box(\Diamond q \wedge \Diamond p)$

(c) $\Box(\neg\Diamond p) \to \Box(\Diamond p \to (q \to \Box p))$

(d) $\Box(q \to \Box\neg p) \to \Box(\Diamond p \to \neg q)$

Hint. In the last exercise, recall that $\Diamond p$ is an abbreviation for $\neg\Box\neg p$.

As in the previous two examples, when invoking Propositional Reasoning it is essential to indicate the underlying propositional tautology and instantiation: this information is necessary in order to validate the proof. Establishing that the given formula is a tautology can be done by means of any of the techniques discussed in the previous chapter.

These two examples illustrate a very simple use of Propositional Reasoning, namely when the target formula follows directly from a propositional tautology. In general, this lemma is also used to chain reasoning steps. The following is a non-exhaustive list of useful tautologies that often appear.

- Implications can be chained by using $(p_1 \to p_2) \to ((p_2 \to p_3) \to (p_1 \to p_3))$, which allows us to infer $\varphi \to \gamma$ from $\varphi \to \psi$ and $\psi \to \gamma$ with two applications of (MP), for any formulas φ, ψ and γ.
- Likewise, conjunctions can be constructed from the tautology $p_1 \to (p_2 \to (p_1 \wedge p_2))$: by applying (MP) twice, we can derive $\varphi \wedge \psi$ from any two formulas φ and ψ.
- Conversely, the tautologies $p_1 \wedge p_2 \to p_1$ and $p_1 \wedge p_2 \to p_2$ are useful to isolate the two conjuncts of a conjunction (as in the previous example).
- For introducing disjunctions, we use the dual tautologies $p_1 \to p_1 \vee p_2$ and $p_2 \to p_1 \vee p_2$.
- Finally, case analysis can be performed by invoking the tautology $(p_1 \to p_3) \to ((p_2 \to p_3) \to (p_1 \vee p_2 \to p_3))$. This allows us to combine a proof of $\varphi \to \gamma$ and a proof of $\psi \to \gamma$ into a proof of $\varphi \vee \psi \to \gamma$.

Of course, nothing prevents us from using other tautologies – and we often will. In particular, the de Morgan laws often help in proving formulas that use $\Diamond$. The above list is often useful when trying to figure out what the next step should be; but occasionally we can combine several steps with a more complex tautology.

Example. We now show that $\Box$ also distributes over conjunction in a way similar to implication. Let φ and ψ be arbitrary modal formulas; we prove that $\vdash_L \Box(\varphi \wedge \psi) \to \Box\varphi \wedge \Box\psi$.

The intuition to build the derivation is as follows. First, we use the tautologies $p_1 \wedge p_2 \to p_1$ and $p_1 \wedge p_2 \to p_2$ with the mapping $p_1 \mapsto \varphi$ and $p_2 \mapsto \psi$ in order to derive $\Box(\varphi \wedge \psi) \to \Box\varphi$ and $\Box(\varphi \wedge \psi) \to \Box\psi$. Then we combine the conclusions of these formulas in a single implication using the tautology $(p_1 \to p_2) \to ((p_1 \to p_3) \to (p_1 \to p_2 \wedge p_3))$ with instantiation $p_1 \mapsto \Box(\varphi \wedge \psi)$, $p_2 \mapsto \Box\varphi$ and $p_3 \mapsto \Box\psi$.

1. $\varphi \wedge \psi \to \varphi$	Prop
2. $\Box(\varphi \wedge \psi \to \varphi)$	Nec(1)
3. $\Box(\varphi \wedge \psi \to \varphi) \to (\Box(\varphi \wedge \psi) \to \Box\varphi)$	K
4. $\Box(\varphi \wedge \psi) \to \Box\varphi$	MP(3,2)
5. $\varphi \wedge \psi \to \psi$	Prop
6. $\Box(\varphi \wedge \psi \to \psi)$	Nec(5)
7. $\Box(\varphi \wedge \psi \to \psi) \to (\Box(\varphi \wedge \psi) \to \Box\psi)$	K
8. $\Box(\varphi \wedge \psi) \to \Box\psi$	MP(7,6)
9. $(\Box(\varphi \wedge \psi) \to \Box\varphi) \to ((\Box(\varphi \wedge \psi) \to \Box\psi) \to (\Box(\varphi \wedge \psi) \to \Box\varphi \wedge \Box\psi))$	Prop
10. $(\Box(\varphi \wedge \psi) \to \Box\psi) \to (\Box(\varphi \wedge \psi) \to \Box\varphi \wedge \Box\psi)$	MP(9,4)
11. $\Box(\varphi \wedge \psi) \to \Box\varphi \wedge \Box\psi$	MP(10,8)

The structure of this derivation reflects the informal argument given above: we first derive $\Box(\varphi \wedge \psi) \to \Box\varphi$ (steps 1–4) and $\Box(\varphi \wedge \psi) \to \Box\psi$ (steps 5–8), then combine them as described in steps 9–11. ◁

Exercise 19. Show that $\vdash_L \Box\varphi \vee \Box\psi \to \Box(\varphi \vee \psi)$.

As discussed in the previous section, we are often interested in relative validity rather than validity in the class of all frames. It thus makes sense to allow for other modal axioms to be used as axiom schemata in the Hilbert calculus for modal logic. We indicate that axioms $A_1, \ldots, A_k$ are being assumed by replacing L with $L + A_1 \ldots A_k$ in the superscript of the derivation symbol.

Thus, $\Gamma \vdash_{L+T} \varphi$ means that we can derive φ from Γ using axiom T, $\Gamma \vdash_{L+B45} \varphi$ means that we can derive φ from Γ using axioms B, 4 and 5, and so on. Note that we can use *any* instance of the additional axioms (or several different instances of them) in the derivation; this is why it is important to mark that they are being added as a schematic axiom, rather than as a hypothesis.

Example. Recall that T is the axiom $\Box\varphi \to \varphi$. The following derivation shows that $\vdash_{L+T} \Box\Box p \to p$.

1. $\Box p \to p$	T
2. $\Box\Box p \to \Box p$	T
3. $(\Box\Box p \to (\Box p \to p)) \to ((\Box\Box p \to \Box p) \to (\Box\Box p \to p))$	(Ax.2)
4. $(\Box p \to p) \to (\Box\Box p \to (\Box p \to p))$	(Ax.1)
5. $\Box\Box p \to (\Box p \to p)$	MP(4,1)
6. $(\Box\Box p \to \Box p) \to (\Box\Box p \to p)$	MP(3,5)
7. $\Box\Box p \to p$	MP(6,2)

In this derivation we used axiom T twice, once with φ instantiated as p (step 1), and another with φ instantiated as $\Box p$ (step 2).

Alternatively, we could invoke Propositional Reasoning to obtain the following, simpler, derivation.

1. $\Box p \to p$	T
2. $\Box\Box p \to \Box p$	T
3. $(\Box\Box p \to \Box p) \to ((\Box p \to p) \to (\Box\Box p \to p))$	Prop
4. $(\Box p \to p) \to (\Box\Box p \to p)$	MP(3,2)
5. $\Box\Box p \to p$	MP(4,1)

In step 3, we used the propositional tautology $(p_1 \to p_2) \to ((p_2 \to p_3) \to (p_1 \to p_3))$ with the instantiation $p_1 \mapsto \Box\Box p$, $p_2 \mapsto \Box p$ and $p_3 \mapsto p$. ◁

As before, any formula that can be derived in this extended calculus is valid in any regular class of frames satisfying the axioms being assumed.

Theorem 20 (Relative soundness)**.** Let $\mathcal{F}$ be a regular class of frames satisfying the axioms in $\mathcal{A}$ and φ be a formula in modal logic. If $\vdash_{L+\mathcal{A}} \varphi$, then $\models_{\mathcal{F}} \varphi$.

Proof. By induction on the proof of $\Gamma \vdash_L \varphi$. The cases where φ is an instance of a propositional axiom or an instance of K are handled as in the proof of Theorem 18. The cases where φ is

derived by (MP) or (Nec) are also similar, observing in the latter case that $\mathcal{F}$ is assumed to be regular, and therefore Lemma 19 still applies.

Finally, in the new case where φ is an instance of an axiom in $\mathcal{A}$, the thesis immediately follows by the hypothesis on $\mathcal{F}$. □

When reasoning with axioms, one often needs to use duality in order to prove formulas that involve $\Diamond$. For any axiom A, there is a standard strategy to derive $A\Diamond$ from A: instantiate A with the negation of the formulas that should appear in $A\Diamond$, and then use one of the tautologies that revert implications (e.g. $(p_1 \to p_2) \to (\neg p_2 \to \neg p_1)$). Then use de Morgan laws and the definition of $\Diamond$ to rewrite the result in the desired form.

Example. Recall that T is the axiom $\Box\varphi \to \varphi$, and that $T\Diamond$ is $\varphi \to \Diamond\varphi$. We show that $\vdash_{L+T} T\Diamond$.

1. $\Box\neg\varphi \to \neg\varphi$	T
2. $(\Box\neg\varphi \to \neg\varphi) \to (\varphi \to \underbrace{\neg\Box\neg\varphi}_{\Diamond\varphi})$	Prop
3. $\varphi \to \Diamond\varphi$	MP(2,1)

Here, we use the tautology $(p_1 \to \neg p_2) \to (p_2 \to \neg p_1)$ with the instantiation $p_1 \mapsto \Box\neg\varphi$, $p_2 \mapsto \varphi$. Observe that $\neg\Box\neg\varphi$ and $\Diamond\varphi$ are the same formula, since $\Diamond$ is defined as an abbreviation. ◁

Example. As a less trivial example, consider the following derivation of a generic instance of $4\Diamond$ from 4.

1. $\Box\neg\varphi \to \Box\Box\neg\varphi$	4
2. $(\Box\neg\varphi \to \Box\Box\neg\varphi) \to (\neg\Box\Box\neg\varphi \to \underbrace{\neg\Box\neg\varphi}_{\Diamond\varphi})$	Prop
3. $\neg\Box\Box\neg\varphi \to \Diamond\varphi$	MP(2,1)
4. $\Box\neg\varphi \to \neg\neg\Box\neg\varphi$	Prop
5. $\Box(\Box\neg\varphi \to \neg\neg\Box\neg\varphi)$	Nec(4)
6. $\Box(\Box\neg\varphi \to \neg\neg\Box\neg\varphi) \to (\Box\Box\neg\varphi \to \Box\neg\neg\Box\neg\varphi)$	K
7. $\Box\Box\neg\varphi \to \Box\neg\neg\Box\neg\varphi$	MP(6,5)
8. $(\Box\Box\neg\varphi \to \Box\neg\neg\Box\neg\varphi) \to (\underbrace{\neg\Box\neg\neg\Box\neg\varphi}_{\Diamond\Diamond\varphi} \to \neg\Box\Box\neg\varphi)$	Prop
9. $\Diamond\Diamond\varphi \to \neg\Box\Box\neg\varphi$	MP(8,7)
10. $(\Diamond\Diamond\varphi \to \neg\Box\Box\neg\varphi) \to ((\neg\Box\Box\neg\varphi \to \Diamond\varphi) \to (\Diamond\Diamond\varphi \to \Diamond\varphi))$	Prop
11. $(\neg\Box\Box\neg\varphi \to \Diamond\varphi) \to (\Diamond\Diamond\varphi \to \Diamond\varphi)$	MP(10,9)
12. $\Diamond\Diamond\varphi \to \Diamond\varphi$	MP(11,3)

Let us analyse this derivation in more detail. In step 2 we use the propositional tautology $(p_1 \to p_2) \to (\neg p_2 \to \neg p_1)$ with instantiation $p_1 \mapsto \Box\neg\varphi$ and $p_2 \mapsto \Box\Box\neg\varphi$. In step 3, we get a formula with the desired consequent, but whose antecedent is $\neg\Box\Box\neg\varphi$; this is not the same as $\Diamond\Diamond\varphi$, which expands to $\neg\Box\neg\neg\Box\neg\varphi$.

Therefore, we need to find a proof of $\neg\Box\neg\neg\Box\neg\varphi \to \neg\Box\Box\neg\varphi$, which we can combine with the previous formula by chaining implications (steps 10–12, where step 10 uses the propositional

tautology $(p_1 \to p_2) \to ((p_2 \to p_3) \to (p_1 \to p_3))$ with instantiation $p_1 \mapsto \Diamond\Diamond\varphi$, $p_2 \mapsto \neg\Box\Box\neg\varphi$ and $p_3 \mapsto \Diamond\varphi$).

In order to prove $\neg\Box\neg\neg\Box\neg\varphi \to \neg\Box\Box\neg\varphi$, we proceed by counterreciprocal – we first derive $\Box\Box\neg\varphi \to \Box\neg\neg\Box\neg\varphi$ (steps 4–7), and again invoke the tautology $(p_1 \to p_2) \to (\neg p_2 \to \neg p_1)$, this time mapping $p_1 \mapsto \Box\Box\neg\varphi$ and $p_2 \mapsto \Box\neg\neg\Box\neg\varphi$ (step 8). Deriving $\Box\Box\neg\varphi \to \Box\neg\neg\Box\neg\varphi$ is easy, as this follows from (Nec) and K applied to the tautology $p_1 \to \neg\neg p_1$ by using $p_1 \mapsto \Box\neg\varphi$ (step 4).

Therefore $\vdash_{L+4} 4\Diamond$. ◁

Exercise 20. Show that the following judgements are derivable.

(a) $\vdash_{L+B} B\Diamond$ (b) $\vdash_{L+5} 5\Diamond$ (c) $\vdash_L K\Diamond$ (d) $\vdash_L K\Diamond\wedge$

Using the Hilbert calculus for modal logic, we can also establish relationships between axioms – for example, T implies D, and B and T together imply 5. This can also be seen by looking at the corresponding semantic properties of the classes of frames these axioms characterize, but typically the Hilbert calculus proof is more concise and easier to check.

Exercise 21. Show that the following judgements are derivable.

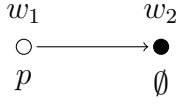
(a) $\vdash_{L+T} D$ (b) $\vdash_{L+BT} 5$ (c) $\vdash_{L+T5} B$

The Propositional Reasoning theorem makes proofs in the Hilbert calculus for modal logic much easier than we were used to from propositional logic, and so far we have not discussed the Deduction Theorem. Although it is not as useful as before to construct derivations, the Deduction Theorem states a result that is important in itself: implication internalizes logical consequence (in the same sense that the Duality Theorem states that duality internatizes logical negation). Regardless of its practical relevance for building derivations, it is always interesting to understand whether a logic enjoys this property.

The direct translation of the Deduction Theorem to modal logic does not hold, and for the same reason that we were only able to prove weak soundness of the Hilbert calculus.

Example. Consider the judgement $\{p\} \vdash_L \Box p$. We already pointed out that this judgement can be shown to be valid with a two-step derivation.

However, $\nvdash_L p \to \Box p$. By soundness of the Hilbert calculus, if this were the case then $p \to \Box p$ would be valid in every frame, and the same example as before shows that this is not true.



Therefore it is not always the case that $\Gamma \cup \{\varphi\} \vdash_L \psi$ implies $\Gamma \vdash_L \varphi \to \psi$. ◁

The reason why this example fails is again that necessitation is a global inference rule: soundness of necessitation (Lemma 19) has as hypothesis that the formula it is applied to

holds in all frames. However, in $\Gamma \models \varphi$ the hypotheses are local: they restrict to valuations that make Γ true on the initial world of a frame. This is the same issue encountered when discussing Theorem 18, and we now look into how to address it.

Definition. Let $\varphi_1, \ldots, \varphi_n$ be a derivation from a set of hypotheses Γ. Formula φ_i *depends* on Γ if $\varphi_i \in \Gamma$ or if φ_i is obtained by applying an inference rule to a formula that depends on Γ.

We also say that ψ depends on φ to make it explicit that φ is the only hypothesis used in one of the steps relevant for deriving ψ.

Example. Consider the following derivation of $\{p \to \Box p\} \vdash_L \Box p \to \Box\Box p$.

1. $p \to \Box p$	Hyp
2. $\Box(p \to \Box p)$	Nec(2)
3. $\Box(p \to \Box p) \to (\Box p \to \Box\Box p)$	K
4. $\Box p \to \Box\Box p$	MP(3,4)

In this derivation, the formulas in steps 1, 2 and 4 depend on the hypothesis, but the formula in step 3 does not. ◁

Note that it can be assumed that any formula introduced by Propositional Reasoning does not depend on any hypothesis.

Lemma 22. Suppose that $\Gamma \vdash_L \varphi$ and that φ does not depend on Γ. Then $\vdash_L \varphi$.

Proof. Let $\mathcal{D}$ be the derivation of $\Gamma \vdash_L \varphi$, and consider the derivation $\mathcal{D}'$ obtained from $\mathcal{D}$ by removing every formula that depends on Γ. The following two observations establish that $\vdash_L \varphi$.

- $\mathcal{D}'$ is a valid derivation: any formula ψ in $\mathcal{D}'$ is either an axiom or an application of an inference rule to formulas occurring previously in $\mathcal{D}$. But since ψ does not depend on Γ, the premises of the inference rule applied to derive ψ also do not depend on Γ, and therefore they are also in $\mathcal{D}'$.
- The last formula in $\mathcal{D}'$ is φ, since it does not depend on Γ. Therefore $\mathcal{D}'$ derives φ. □

Theorem 21 (Deduction Theorem)**.** If $\Gamma \cup \{\varphi\} \vdash_L \psi$ and the derivation of ψ does not apply (Nec) to any formula depending on Γ, then $\Gamma \vdash_L \varphi \to \psi$.

Proof. By induction on the derivation of ψ. The proof is similar to the corresponding result for propositional logic, with just one new case – the case when $\Box\psi$ follows by applying (Nec) to ψ.

By hypothesis, the derivation of $\Box\psi$ does not apply (Nec) to any formula depending on Γ. This means that, in particular, ψ does not depend on Γ. By the previous lemma, this implies that $\vdash_L \psi$, which allows us to produce the following derivation.

n. ψ	
$n+1$. $\Box\psi$	Nec(n)
$n+2$. $\Box\psi \to (\varphi \to \Box\psi)$	(Ax.1)
$n+3$. $\varphi \to \Box\psi$	MP($n+2$,$n+1$)

This derivation shows not only that $\vdash_L \varphi \to \Box\psi$, but also that $\Gamma \vdash_L \varphi \to \Box\psi$, establishing the thesis. □

Observe that the premise of the Deduction Theorem refers to the actual derivation, and not to the judgement. It may be the case that there are different derivations of $\Gamma \cup \{\varphi\} \vdash_L \psi$, and that the Deduction Theorem is applicable to some of them, but not all (see also Exercise 22 (d) and (e)).

Exercise 22. The following derivations all have the form $\{\varphi\} \vdash_L \psi$; all instances of (Prop) use a propositional tautology. In which cases can you invoke the Deduction Theorem to claim that $\vdash_L \varphi \to \psi$?

(a) $\{p\} \vdash_L \Box(p \vee q)$

1. p	(Hyp)
2. $p \to p \vee q$	Prop
3. $p \vee q$	MP(2,1)
4. $\Box(p \vee q)$	Nec(3)

(b) $\{\Box p\} \vdash_L \Box(q \to p)$

1. $\Box p$	(Hyp)
2. $p \to (q \to p)$	(Ax.1)
3. $\Box(p \to (q \to p))$	Nec(2)
4. $\Box(p \to (q \to p))$ $\to (\Box p \to \Box(q \to p))$	K
5. $\Box p \to \Box(q \to p)$	MP(4,3)
6. $\Box(q \to p)$	MP(5,1)

(c) $\{p \wedge q\} \vdash_L p \to \Box q$

1. $p \wedge q$	(Hyp)
2. $p \wedge q \to q$	Prop
3. q	MP(2,1)
4. $\Box q$	Nec(3)
5. $\Box q \to (p \to \Box q)$	(Ax.1)
6. $p \to \Box q$	MP(5,4)

(d) $\{p \wedge q, \Box p \to p\} \vdash_L p$

1. $p \wedge q$	(Hyp)
2. $p \wedge q \to p$	Prop
3. p	MP(2,1)

(e) $\{p \wedge q, \Box p \to p\} \vdash_L p$

1. $p \wedge q$	(Hyp)
2. $p \wedge q \to p$	Prop
3. $\Box(p \wedge q \to p)$	Nec(2)
4. $\Box(p \wedge q \to p)$ $\to (\Box(p \wedge q) \to \Box p)$	K
5. $\Box(p \wedge q) \to \Box p$	MP(4,3)
6. $\Box(p \wedge q)$	Nec(1)
7. $\Box p$	MP(5,6)
8. $\Box p \to p$	(Hyp)
9. p	MP(8,7)

Example. Using the Deduction Theorem, we show that $\vdash_{L+4} \Box p \to \Box\Box\Box p$. We start by proving that $\{\Box p\} \vdash_{L+4} \Box\Box\Box p$.

1. $\Box p$	(Hyp)
2. $\Box p \to \Box\Box p$	4
3. $\Box\Box p$	MP(2,1)
4. $\Box\Box p \to \Box\Box\Box p$	4
5. $\Box\Box\Box p$	MP(4,3)

This derivation does not use necessitation, therefore the Deduction Theorem is applicable, and we can conclude that $\vdash_{L+4} \Box p \to \Box\Box\Box p$.

It is also not hard to find a direct proof of this judgement.

1. $\Box p \to \Box\Box p$	4
2. $\Box\Box p \to \Box\Box\Box p$	4
3. $(\Box p \to \Box\Box p) \to ((\Box\Box p \to \Box\Box\Box p) \to (\Box p \to \Box\Box\Box p))$	Prop
4. $(\Box\Box p \to \Box\Box\Box p) \to (\Box p \to \Box\Box\Box p)$	MP(3,1)
5. $\Box p \to \Box\Box\Box p$	MP(4,2)

where the formula in step 3 is obtained by instantiating the propositional tautology $(p_1 \to p_2) \to ((p_2 \to p_3) \to (p_1 \to p_3))$ by using the mapping $p_1 \mapsto \Box p$, $p_2 \mapsto \Box\Box p$ and $p_3 \mapsto \Box\Box\Box p$.

However, using the Deduction Theorem yields a derivation that is arguably simpler: it uses shorter formulas, and does not require invoking Propositional Reasoning. ◁

Example. As a more complex example, we show that $\vdash_{L+D} \Box\Box p \to \Diamond\Diamond p$. We start by showing that $\{\Box\Box p\} \vdash_{L+D} \Diamond\Diamond p$.

1. $\Box\Box p$	(Hyp)
2. $\Box p \to \Diamond p$	D
3. $\Box(\Box p \to \Diamond p)$	Nec(2)
4. $\Box(\Box p \to \Diamond p) \to (\Box\Box p \to \Box\Diamond p)$	K
5. $\Box\Box p \to \Box\Diamond p$	MP(4,3)
6. $\Box\Diamond p$	MP(5,1)
7. $\Box\Diamond p \to \Diamond\Diamond p$	D
8. $\Diamond\Diamond p$	MP(7,6)

This derivation uses necessitation in step 3, but it is applied to an instance of axiom D, and thus not to a formula depending on the hypothesis. Therefore, the Deduction Theorem is applicable, and we can conclude that $\vdash_{L+D} \Box\Box p \to \Diamond\Diamond p$. ◁

Exercise 23. Using the Deduction Theorem, show that:

(a) $\vdash_{L+T} \Box\Box p \to p$ (b) $\vdash_{L+T} \Box(p \to q) \to ((p \vee q) \to q)$

The same restriction allows us to prove a more robust soundness result for this calculus.

Theorem 22 (Strong soundness)**.** Let Γ be a set of formulas in modal logic, φ be a formula, $\mathcal{A}$ be a (possibly empty) set of axioms, and $\mathcal{F}$ be a regular class of frames satisfying the axioms in $\mathcal{A}$.

Suppose that $\Gamma \vdash_{L+\mathcal{A}} \varphi$ and that there is a derivation of this fact where (Nec) is never applied to any formula depending on Γ. Then $\Gamma \models_{\mathcal{F}} \varphi$.

Proof. By induction on the derivation of $\Gamma \vdash_{L+\mathcal{A}} \varphi$. For all the cases except that of (Nec), the proof is as in that of Theorem 18. For the case of (Nec), we observe that the assumptions

of the theorem guarantee that the formula to which it is being applied is derivable from the empty set of hypotheses (as in the proof of Theorem 21), and therefore the argument from the proof of Theorem 18 still holds. □

Furthermore, we can generalize the construction in the completeness proof for the propositional Hilbert calculus to obtain a similar result for modal logic. We start by presenting the proof of weak completeness for K, and discuss how this construction can be generalized.

Lemma 23. For all n and formulas $\varphi_1, \ldots, \varphi_n$, $\vdash_L \Box\varphi_1 \wedge \ldots \wedge \Box\varphi_n \to \Box(\varphi_1 \wedge \ldots \wedge \varphi_n)$.

Proof. By induction on n. For $n = 1$ the thesis is simply $\vdash_L \Box\varphi_1 \to \Box\varphi_1$, which is a propositional tautology. Assume that $\vdash_L \Box\varphi_1 \wedge \ldots \wedge \Box\varphi_n \to \Box(\varphi_1 \wedge \ldots \wedge \varphi_n)$; we show that $\vdash_L \Box\varphi_1 \wedge \ldots \wedge \Box\varphi_{n+1} \to \Box(\varphi_1 \wedge \ldots \wedge \varphi_{n+1})$. For simplicity, we write $\vec{\varphi}$ for $\varphi_1 \wedge \ldots \varphi_n$ and $\vec{\Box\varphi}$ for $\Box\varphi_1 \wedge \ldots \wedge \Box\varphi_n$, and abbreviate some formulas throughout the derivation to make the proposional tautologies used more evident.

1. $\vec{\varphi} \to (\varphi_{n+1} \to (\vec{\varphi} \wedge \varphi_{n+1}))$	Prop
2. $\underbrace{\Box(\vec{\varphi} \to (\varphi_{n+1} \to (\vec{\varphi} \wedge \varphi_{n+1})))}_{\psi_1}$	Nec(1)
3. $\psi_1 \to (\Box\vec{\varphi} \to \underbrace{\Box(\varphi_{n+1} \to (\vec{\varphi} \wedge \varphi_{n+1}))}_{\psi_2})$	K
4. $\psi_2 \to \underbrace{(\Box\varphi_{n+1} \to \Box(\vec{\varphi} \wedge \varphi_{n+1}))}_{\psi_3}$	K
5. $(\psi_1 \to (\Box\vec{\varphi} \to \psi_2)) \to ((\psi_2 \to \psi_3) \to (\psi_1 \to (\Box\vec{\varphi} \to \psi_3)))$	Prop
6. $(\psi_2 \to \psi_3) \to (\psi_1 \to (\Box\vec{\varphi} \to \psi_3))$	MP(5,3)
7. $\psi_1 \to (\Box\vec{\varphi} \to \psi_3)$	MP(6,4)
8. $\Box\vec{\varphi} \to \psi_3$	MP(7,2)
9. $\vec{\Box\varphi} \to \Box\vec{\varphi}$	IH
10. $(\vec{\Box\varphi} \to \Box\vec{\varphi}) \to ((\Box\vec{\varphi} \to \psi_3) \to (\vec{\Box\varphi} \wedge \Box\varphi_{n+1} \to \Box(\vec{\varphi} \wedge \varphi_{n+1})))$	Prop
11. $(\Box\vec{\varphi} \to \psi_3) \to (\vec{\Box\varphi} \wedge \Box\varphi_{n+1} \to \Box(\vec{\varphi} \wedge \varphi_{n+1}))$	MP(10,9)
12. $\vec{\Box\varphi} \wedge \Box\varphi_{n+1} \to \Box(\vec{\varphi} \wedge \varphi_{n+1})$	MP(11,8)

This establishes the thesis. □

Theorem 23 (Weak completeness)**.** Let φ be a formula and $\mathcal{F}$ be the class of all frames. If $\models_{\mathcal{F}} \varphi$, then $\vdash_L \varphi$.

Proof. We rely heavily on the construction of maximally consistent sets of formulas in the proof of propositional completeness. There, we saw that every consistent set of formulas can be extended to a maximally consistent set of formulas; we can use the same construction with the more expressive language of modal logic, using the notion of derivation including axiom K, rule (Nec) and the restriction in the statement of the lemma.

We make a model whose set of worlds are all maximally consistent sets of formulas, with accessibility relation defined by

$$\Sigma R \Delta \text{ iff } \varphi \in \Delta \text{ whenever } \Box\varphi \in \Sigma$$

and $V \Vdash_\Sigma p$ iff $p \in \Sigma$. (Valuation V is defined for each maximal consistent set as in the case of propositional logic.)

By structural induction, we can now show that $V \Vdash_\Sigma \varphi$ iff $\varphi \in \Sigma$. The proof in the case of propositional connectives is exactly as in the propositional case.

The only new case is the case when φ is $\Box\psi$ for some formula ψ.

- If $\Box\psi \in \Sigma$ and $\Sigma R \Delta$, then by definition of R we immediately know that $\psi \in \Delta$, and by induction hypothesis $V \Vdash_\Delta \psi$. Since this holds for every possible Δ accessible from Σ, we conclude that $V \Vdash_\Sigma \Box\psi$.

- Now assume that $\Box\psi \notin \Sigma$ and consider the set $\Delta^- = \{\neg\psi\} \cup \{\gamma \mid \Box\gamma \in \Sigma\}$. We first show that Δ^- is consistent.

  If Δ^- is not consistent, then by propositional reasoning there exist a finite number of formulas $\{\delta_1, \ldots, \delta_n\} \in \Sigma$ such that $\delta_1 \wedge \ldots \wedge \delta_n \vdash_L \psi$, and by the Deduction Theorem we can conclude that $\vdash_L \delta_1 \wedge \ldots \wedge \delta_n \to \psi$. We can then build the following derivation.

1. $\delta_1 \wedge \ldots \wedge \delta_n \to \psi$	Lemma
2. $\Box(\delta_1 \wedge \ldots \wedge \delta_n \to \psi)$	Nec(1)
3. $\Box(\delta_1 \wedge \ldots \wedge \delta_n \to \psi) \to (\Box(\delta_1 \wedge \ldots \wedge \delta_n) \to \Box\psi)$	K
4. $\Box(\delta_1 \wedge \ldots \wedge \delta_n) \to \Box\psi)$	MP(3,2)
5. $\Box\delta_1 \wedge \ldots \wedge \Box\delta_n \to \Box(\delta_1 \wedge \ldots \wedge \delta_n)$	Lemma
6. $(\Box\delta_1 \wedge \ldots \wedge \Box\delta_n \to \Box(\delta_1 \wedge \ldots \wedge \delta_n)) \to$ $\to ((\Box(\delta_1 \wedge \ldots \wedge \delta_n) \to \Box\psi) \to (\Box\delta_1 \wedge \ldots \wedge \Box\delta_n \to \Box\psi))$	Prop
7. $(\Box(\delta_1 \wedge \ldots \wedge \delta_n) \to \Box\psi) \to (\Box\delta_1 \wedge \ldots \wedge \Box\delta_n \to \Box\psi)$	MP(6,5)
8. $\Box\delta_1 \wedge \ldots \wedge \Box\delta_n \to \Box\psi$	MP(7,4)

  But since Σ is closed under derivation, this would imply that $\Box\psi \in \Sigma$, contradicting our hypothesis. Therefore Δ^- is consistent.

  Since Δ^- is consistent, it can be extended to a maximally consistent set Δ, and by definition of R it follows that $\Sigma R \Delta$. Furthermore, $\neg\psi \in \Delta$, so $V \Vdash_\Delta \neg\psi$ by induction hypothesis. It thus follows that $V \nVdash_\Sigma \Box\psi$.

Suppose now that $\nvdash_L \varphi$. Then $\{\neg\varphi\}$ is consistent, hence there is a maximally consistent set Σ such that $\neg\varphi \in \Sigma$. By taking Σ as initial world, we have constructed a model that satisfies $\neg\varphi$, so φ is not valid. □

The general case is now easy.

Theorem 24 (Strong completeness)**.** Let Γ be a set of formulas in modal logic, φ be a formula, $\mathcal{A}$ be a set of axioms, and $\mathcal{F}$ be a regular class of frames that is complete for the axioms in $\mathcal{A}$.

Suppose that $\Gamma \models_{\mathcal{F}} \varphi$. Then $\Gamma \vdash_{L+\mathcal{A}} \varphi$, and there is a derivation of this fact where (Nec) is never applied to any formula depending on Γ.

Proof. This result is established by showing that the construction done earlier also works.

The hypotheses in Γ are simple to deal with: the only place where (Nec) was used in the previous proof was in the derivation in Lemma 23, where it was applied to a propositional tautology. Therefore the thesis automatically holds if we allow formulas from Γ to be used in derivations as allowed by the hypothesis.

In order to deal with axioms, we need to show that the model constructed is in $\mathcal{F}$. By construction, all worlds in this model satisfy all instances of all axioms in $\mathcal{A}$. Since $\mathcal{F}$ is complete for these axioms, it must contain this model. □

Exercise 24. Go through the construction in the proof of weak completeness of the Hilbert calculus for modal logic and verify that the two claims in the proof of strong completeness indeed hold.

3.5 Tableaux calculus

The other deductive system for modal logic that we discuss is the tableaux calculus. Tableaux calculi for modal logics differ from those for propositional logic in two aspects. First, they no longer work with sets of formulas, but rather with sets of judgements stating that some formulas are true or false at some worlds. Second, they syntactically embody K; but additional axioms must be incorporated by adding rules that encode their semantic impact on the accessibility relation.

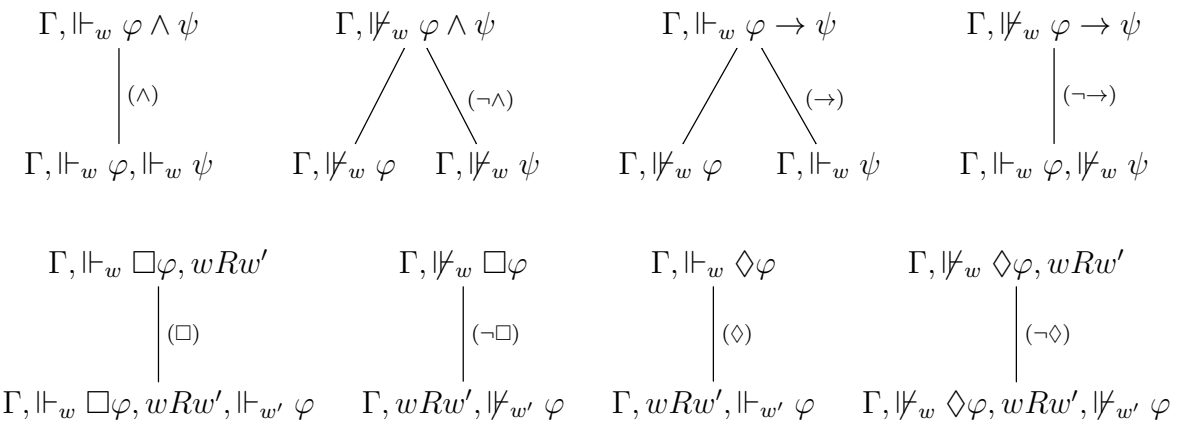
As in propositional logic, building a tableau for a set of formulas corresponds to trying to build a model that satisfies all the formulas in that set. The first difference mentioned above arises from the nature of the semantics of modal logic: formulas are no longer simply true or false, but rather true or false at each world – and the same formula may be true in one world and false in another. The nodes of tableaux for modal logic therefore store judgements of the form $\Vdash_w \varphi$ or $\nVdash_w \varphi$ (read "φ is true at w" and "φ is false at w", respectively) – since we are explicitly using judgements, we no longer need to overload the negation symbol with a semantic meaning. Furthermore, building a model also involves building an accessibility relation; as such, nodes also need to store information about which worlds see each other.

The second difference is for convenience: in principle we can use axioms just as in the Hilbert calculus – by adding the relevant instances explicitly to the judgement we want to prove. But, as we will see, adding the corresponding semantic rule it is much easier in practice, makes for shorter proofs, and removes the need to choose instantiations of axioms. This last point is especially relevant if one is aiming at automating proof search, as it makes it easier to constraint the number of possible proofs. In many cases, this number becomes finite, but of course this depends on the actual axioms that are being used.

Definition. Let W be a set of worlds. A *tableau* for a set Γ is a labeled tree whose nodes contain judgements of the form $\Vdash_w \varphi$, $\nVdash_w \varphi$ or $w_1 R w_2$, with $w, w_1, w_2 \in W$, whose root contains $\Vdash_{w_0} \varphi$ for all $\varphi \in \Gamma$ and some (fixed) $w_0 \in W$, and where every node has descendants generated by one of the following rules.

$$\frac{\Gamma, \Vdash_w \neg\varphi}{\Gamma, \nVdash_w \varphi}\,(\neg) \qquad \frac{\Gamma, \nVdash_w \neg\varphi}{\Gamma, \Vdash_w \varphi}\,(\neg\neg)$$

$$\frac{\Gamma, \Vdash_w \varphi \vee \psi}{\Gamma, \Vdash_w \varphi \quad \Gamma, \Vdash_w \psi}\,(\vee) \qquad \frac{\Gamma, \nVdash_w \varphi \vee \psi}{\Gamma, \nVdash_w \varphi, \nVdash_w \psi}\,(\neg\vee)$$



In rules (◇) and (¬□), w' is a variable that stands for an unknown world. This variable must be fresh, i.e. it may not occur anywhere in Γ.

We briefly discuss the intuition behind these rules.

The rules for the propositional connectives are simply the rules from the tableaux calculus for propositional logic reformulated as a judgement, where the world where the judgement holds is kept unchanged when the rule is applied.

Rules (□) and (¬◇) deal with the universal modalities. The semantics of modal logic states that: if $\Box\varphi$ holds at a world w, then φ holds at every world w' such that wRw'. This is translated as a rule that adds $\Vdash_{w'} \varphi$ whenever $\Vdash_w \Box\varphi$ and wRw' have already been established. (Dually, if $\nVdash_w \Diamond\varphi$ and wRw', then $\nVdash_{w'} \varphi$ can be added.) Since w may be connected to several different worlds (including some that are not yet known), and since it may satisfy several formulas of the form $\Box\varphi$, these rules may need to be applied several times in a proof with intersecting sets of premises. Therefore these rules do not remove their premises from the set of formulas.

Rules (□) and (¬◇) require the node to which they are applied to include judgements of the form wRw'. Initially, no such judgements are present, since the root only contains one node: this is to be expected, since this calculus extends the calculus for propositional logic. New worlds, and information about how they are connected to the existing ones, are added by the rules dealing with existential modalities – (◇) and (¬□).

If $\Vdash_w \Diamond\varphi$, then necessarily w can access a world where φ is true. Rule ◇ encodes this information by *adding* a new world w' satisfying φ, together with the information that wRw'. It is essential that this world be new: we can not assume any connection between it and any existing world. To understand this, consider the case where $\Vdash_w \Diamond p$ and $\Vdash_w \Diamond\neg p$: w must see a world where p holds and another where $\neg p$ holds, and if these worlds are assumed to be the same we wrongly conclude that there is no model where both conditions can hold.

As in the tableaux calculus for propositional logic, we take all connectives as primitive.

Definition. A leaf in a tableau is said to be *contradictory* if it contains two judgements $\Vdash_w \varphi$ and $\nVdash_w \varphi$ for some w and φ.

Definition. A tableau is *closed* if each of its leaves either is contradictory or cannot be expanded further by application of some rule.

As before, we write $\Gamma \vdash_S \varphi$ if there is a closed tableau for $\Gamma' \cup \{\neg\varphi\}$ with $\Gamma' \subseteq \Gamma$ whose leaves are all contradictory.

Example. We start by showing that $\vdash_S \Box p \to \Box(p \vee q)$.

$$\underline{\not\Vdash_w \Box p \to \Box(p \vee q)}$$
$$\Big| \ (\neg\to)$$
$$\Vdash_w \Box p, \underline{\not\Vdash_w \Box(p \vee q)}$$
$$\Big| \ (\neg\Box)$$
$$\Vdash_w \Box p, wRw', \underline{\not\Vdash_{w'} p \vee q}$$
$$\Big| \ (\neg\vee)$$
$$\underline{\Vdash_w \Box p}, \underline{wRw'}, \not\Vdash_{w'} p, \not\Vdash_{w'} q$$
$$\Big| \ (\Box)$$
$$\Vdash_w \Box p, wRw', \boxed{\Vdash_{w'} p}, \boxed{\not\Vdash_{w'} p}, \not\Vdash_{w'} q$$

This tableau is closed and its only leaf is a contradiction, allowing us to conclude that $\vdash_S \Box p \to \Box(p \vee q)$. ◁

Example. We now show that $\vdash_S \Box(p \to q) \to (\Box p \to \Box q)$.

$$\underline{\not\Vdash_w \Box(p \to q) \to (\Box p \to \Box q)}$$
$$\Big| \ (\neg\to)$$
$$\Vdash_w \Box(p \to q), \underline{\not\Vdash_w \Box p \to \Box q}$$
$$\Big| \ (\neg\to)$$
$$\Vdash_w \Box(p \to q), \Vdash_w \Box p, \underline{\not\Vdash_w \Box q}$$
$$\Big| \ (\neg\Box)$$
$$\Vdash_w \Box(p \to q), \underline{\Vdash_w \Box p}, \underline{wRw'}, \not\Vdash_{w'} \psi$$
$$\Big| \ (\Box)$$
$$\underline{\Vdash_w \Box(p \to q)}, \Vdash_w \Box p, \underline{wRw'}, \not\Vdash_{w'} \psi, \Vdash_{w'} p$$
$$\Big| \ (\Box)$$
$$\Vdash_w \Box(p \to q), \Vdash_w \Box p, wRw', \not\Vdash_{w'} q, \Vdash_{w'} p, \underline{\Vdash_{w'} p \to q}$$

$(\to)$

$$\Vdash_w \Box(p \to q), \Vdash_w \Box p \qquad \Vdash_w \Box(p \to q), \Vdash_w \Box p,$$
$$wRw', \not\Vdash_{w'} q, \boxed{\Vdash_{w'} p}, \boxed{\not\Vdash_{w'} p} \qquad wRw', \boxed{\not\Vdash_{w'} q}, \Vdash_{w'} p, \boxed{\Vdash_{w'} q}$$

Again this tableau is closed and all its leaves are contradictions. By replacing p and q with arbitrary formulas φ and ψ, we can prove any instance of K in this calculus. ◁

Example. Likewise, we can prove that $\vdash_S \Box(\Diamond p \wedge (q \to \Box q)) \to \Box\Diamond p$.

$$\underline{\nVdash_w \Box(\Diamond p \wedge (q \to \Box q)) \to \Box\Diamond p}$$
$$\big|\ (\neg\to)$$
$$\Vdash_w \Box(\Diamond p \wedge (q \to \Box q)), \underline{\nVdash_w \Box\Diamond p}$$
$$\big|\ (\neg\Box)$$
$$\underline{\Vdash_w \Box(\Diamond p \wedge (q \to \Box q))}, \underline{wRw'}, \nVdash_{w'} \Diamond p$$
$$\big|\ (\Box)$$
$$\Vdash_w \Box(\Diamond p \wedge (q \to \Box q)), wRw', \underline{\Vdash_{w'} \Diamond p \wedge (q \to \Box q)}, \nVdash_{w'} \Diamond p$$
$$\big|\ (\wedge)$$
$$\Vdash_w \Box(\Diamond p \wedge (q \to \Box q)), wRw', \boxed{\Vdash_{w'} \Diamond p}, \Vdash_{w'} q \to \Box q, \boxed{\nVdash_{w'} \Diamond p}$$

Again the only leaf of this tableau is a contradiction, showing that indeed we can derive the judgement $\vdash_S \Box(\Diamond p \wedge (q \to \Box q)) \to \Box\Diamond p$. ◁

Exercise 25. Find proofs of the following judgements in the tableaux calculus for modal logic.

(a) $\Box(p \to \neg q) \to \Box(q \to \neg p)$

(b) $\Box(\Diamond p \wedge \Diamond q) \to \Box(\Diamond q \wedge \Diamond p)$

(c) $\Box(\neg\Diamond p) \to \Box(\Diamond p \to (q \to \Box p))$

(d) $\Box(q \to \Box\neg p) \to \Box(\Diamond p \to \neg q)$

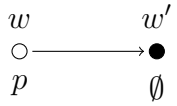
Example. The following tableaux show that $\nvdash_S p \to \Box p$ and that $\nvdash_S \Box p \to p$.

$$\underline{\nVdash_w p \to \Box p}$$
$$\big|\ (\neg\to)$$
$$\Vdash_w p, \underline{\nVdash_w \Box p}$$
$$\big|\ (\neg\Box)$$
$$\Vdash_w p, wRw', \nVdash_{w'} p$$

$$\underline{\nVdash_w \Box p \to p}$$
$$\big|\ (\neg\to)$$
$$\Vdash_w \Box p, \nVdash_w p$$

As in the propositional case, we can use a non-contradictory leaf of a closed tableau to build a model satisfying all judgements in its root node. We take as set of worlds the worlds occurring in the leaf, the accessibility relation consisting exactly of the judgements of the form wRw' in the leaf, and the valuation that assigns true to every pair $\langle w, p\rangle$ such that $\Vdash_w p$ appears in the leaf. The initial world is the world in the judgement at the root of the tree.

Applying this construction to the two tableaux above yields the models



and

and it is straightforward to check that they indeed falsify the formulas $p \to \Box p$ and $\Box p \to p$, respectively. ◁

Unfortunately, unlike in the propositional case, this does not automatically yield a decision procedure for modal logic: tableaux may be infinite.

Example. Consider the following tableau for the formula $(\Diamond p \to \Box\neg q) \to (\neg\Box p \to \Box q)$.

$$\underline{\Vdash\!\!\!\!/_w\ (\Diamond p \to \Box\neg q) \to (\neg\Box p \to \Box q)}$$
$$\Big|\ (\neg\to)$$
$$\Vdash_w \Diamond p \to \Box\neg q, \underline{\Vdash\!\!\!\!/_w\ \neg\Box p \to \Box q}$$
$$\Big|\ (\neg\to)$$
$$\Vdash_w \Diamond p \to \Box\neg q, \underline{\Vdash_w\ \neg\Box p}, \Vdash\!\!\!\!/_w\ \Box q$$
$$\Big|\ (\neg)$$
$$\Vdash_w \Diamond p \to \Box\neg q, \underline{\Vdash\!\!\!\!/_w\ \Box p}, \Vdash\!\!\!\!/_w\ \Box q$$
$$\Big|\ (\neg\Box)$$
$$\Vdash_w \Diamond p \to \Box\neg q, wRw_p, \Vdash\!\!\!\!/_{w_p}\ p, \underline{\Vdash\!\!\!\!/_w\ \Box q}$$
$$\Big|\ (\neg\Box)$$
$$\underline{\Vdash_w \Diamond p \to \Box\neg q}, wRw_p, \Vdash\!\!\!\!/_{w_p}\ p, wRw_q, \Vdash\!\!\!\!/_{w_q}\ q$$

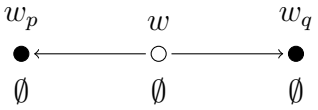
$(\to)$ left:
$$\underline{\Vdash\!\!\!\!/_w \Diamond p}, wRw_p, \Vdash\!\!\!\!/_{w_p}\ p, \underline{wRw_q}, \Vdash\!\!\!\!/_{w_q}\ q$$
$$\Big|\ (\neg\Diamond)$$
$$\Vdash\!\!\!\!/_w \Diamond p, wRw_p, \Vdash\!\!\!\!/_{w_p}\ p, wRw_q, \Vdash\!\!\!\!/_{w_q}\ p, \Vdash\!\!\!\!/_{w_q}\ q$$

$(\to)$ right:
$$\underline{\Box\neg q}, wRw_p, \Vdash\!\!\!\!/_{w_p}\ p, \underline{wRw_q}, \Vdash\!\!\!\!/_{w_q}\ q$$
$$\Big|\ (\Box)$$
$$\Box\neg q, wRw_p, \Vdash\!\!\!\!/_{w_p}\ p, wRw_q, \underline{\Vdash_{w_q}\ \neg q}, \Vdash\!\!\!\!/_{w_q}\ q$$
$$\Big|\ (\neg)$$
$$\underline{\Box\neg q}, \underline{wRw_p}, \Vdash\!\!\!\!/_{w_p}\ p, wRw_q, \Vdash\!\!\!\!/_{w_q}\ q$$
$$\Big|\ (\Box)$$
$$\Box\neg q, wRw_p, \underline{\Vdash_{w_p}\ \neg q}, \Vdash\!\!\!\!/_{w_p}\ p, wRw_q, \Vdash\!\!\!\!/_{w_q}\ q$$
$$\Big|\ (\neg)$$
$$\Box\neg q, wRw_p, \Vdash\!\!\!\!/_{w_p}\ q, \Vdash\!\!\!\!/_{w_p}\ p, wRw_q, \Vdash\!\!\!\!/_{w_q}\ q$$

This tableau is not closed. In the right branch we can still apply rule $(\Box)$; this will lead to an infinite branch whose nodes are all either the last one shown, or include also one or both of the judgements $\Vdash_{w_p} \neg q$, $\Vdash_{w_q} \neg q$. Likewise, we can apply rule $(\neg\Diamond)$ to the node on the left branch and generate a descendant that is equal to itself.

Nevertheless, we can still use this tableau to infer a model, for example by applying the construction described above to the node on the left branch (which in practice behaves as a

leaf). We obtain the model



which indeed falsifies $(\Diamond p \to \Box\neg q) \to (\neg\Box p \to \Box q)$. ◁

Exercise 26. Use the tableaux calculus for modal logic to find a model that falsifies each of the following formulas.

(a) $\Diamond(p \wedge \Box\Box\neg p)$ (b) $\neg(\Diamond p \wedge \Diamond\neg p)$ (c) $\Box(p \vee q) \to \Box p \vee \Box q$

Soundness of the tableaux calculus for propositional modal logic relies on the following lemma, similar to the one for propositional logic.

Definition. Let $F = \langle W, R, w_0\rangle$ be a directed frame and V be a valuation over F. Let Γ be a set of judgements in the tableaux calculus for modal logic, and let σ be a mapping from the set of worlds occurring in Γ to W. Then $\langle F, V\rangle$ *satisfies* Γ *under* σ if:

- $V \Vdash_{\sigma(w)} \varphi$ for every judgement of the form $\Vdash_w \varphi$ in Γ;
- $V \nVdash_{\sigma(w)} \varphi$ for every judgement of the form $\nVdash_w \varphi$ in Γ;
- $\sigma(w)R\sigma(w')$ for every judgement of the form wRw' in Γ.

Lemma 24. Let Γ be a set of judgements in the tableaux calculus for modal logic, r be a rule in the same calculus, $F = \langle W, R, w_0\rangle$ be a directed frame, V be a valuation over F, and σ be a mapping from the set of worlds in the premise of r to W. Then $\langle F, V\rangle$ satisfies the premise of r under σ iff for one of the nodes in the conclusion of r there is a mapping σ' from the set of worlds in that node to W such that $\langle F, V\rangle$ satisfies that node under σ' and $\sigma'(w) = \sigma(w)$ for each w occurring in the premise of r.

Proof. As in the propositional case, the result is proven by showing that it holds for every rule. For the propositional connectives, the proof is the same as before, taking $\sigma' = \sigma$ (since these rules do not add new worlds).

We illustrate the cases of the rules dealing with the necessity modality $\Box$.

For rule ($\Box$), assume that $\langle F, V\rangle$ satisfies $\Gamma, \Vdash_w \Box\varphi$, wRw' under σ. Then $\langle F, V\rangle$ satisfies Γ under σ, and furthermore $V \Vdash_{\sigma(w)} \Box\varphi$ and $\sigma(w)R\sigma(w')$. From the semantics of $\Box$, this implies that $V \Vdash_{\sigma(w')} \varphi$; hence the thesis holds by taking $\sigma = \sigma'$.

For rule ($\neg\Box$), assume that $\langle F, V\rangle$ satisfies $\Gamma, \nVdash_w \Box\varphi$, wRw' under σ. Then $\langle F, V\rangle$ satisfies Γ under σ, and furthermore $V \nVdash_{\sigma(w)} \Box\varphi$. From the semantics of $\Box$, this implies that there exists a world $\star \in W$ such that $\sigma(w)R\star$ and $V \nVdash_\star \varphi$. Define σ' by $\sigma'(w') = \star$ and $\sigma'(w) = \sigma(w)$ if $w \neq w'$. Since w' does not occur in the premise of the rule, this definition automatically ensures that $\langle F, V\rangle$ satisfies Γ under σ'; also $V \nVdash_{\sigma'(w')} \varphi$ and $\sigma(w)R\sigma(w')$ by construction. Therefore $\langle F, V\rangle$ satisfies $\Gamma, wRw', \nVdash_{w'} \varphi$ under σ'. □

Exercise 27. Prove Lemma 24 for the rules involving the possibility modality $\Diamond$.

As before, a straightforward induction establishes a similar result about tableaux.

Lemma 25. Let Γ be a set of judgements in the tableaux calculus for modal logic, T be a tableau with root labeled by Γ, $F = \langle W, R, w_0 \rangle$ be a directed frame, V be a valuation over F, and σ be a mapping from the set of worlds in the premise of r to W. Then $\langle F, V \rangle$ satisfies the premise of r under σ iff for one of the leaves of T there is a mapping σ' from the set of worlds in that node to W such that $\langle F, V \rangle$ satisfies that leaf under σ' and $\sigma'(w) = \sigma(w)$ for each w occurring in the premise of r.

Theorem 25 (Soundness)**.** Let Γ be a set of propositional formulas and φ be a formula. If $\Gamma \vdash_S \varphi$, then $\Gamma \models \varphi$.

Proof. Suppose that $\Gamma \vdash_S \varphi$. Then there is a tableau T whose root contains judgements of the form $\Vdash_w \psi$ where $\psi \in \Gamma, \neg\varphi$ and such that all branches of T are finite and end in contradictory leaves.

Let $F = \langle W, R, w_0 \rangle$ be a directed frame and V be a valuation over F. Let w be the only world occurring in the root of T and define $\sigma(w) = w_0$. Then $\langle F, V \rangle$ satisfies the root of T under σ iff $\langle F, V \rangle$ satisfies some leaf of T under some σ' such that $\sigma'(w) = w_0$. But no valuation can satisfy a contradictory leaf, hence $\langle F, V \rangle$ cannot satisfy the root of T under σ. Therefore $V \nVdash_{w_0} \psi$ for some $\psi \in \Gamma \cup \{\neg\varphi\}$.

As in the propositional case, this implies that $\Gamma \cup \{\neg\varphi\}$ is contradictory, and therefore $\Gamma \models \varphi$. □

Exercise 28. Construct a tableau for the negation of each of the following formulas, and use it either to find a model that falsifies them, or to argue that none exist.

(a) $\Box\Box p \to \Box\Diamond p$ (b) $(\Diamond p \wedge \Box q) \to \Diamond(p \to q)$ (c) $\Diamond(p \vee q) \wedge \Box p \wedge \Box(\neg q)$

So far we have only seen the calculus for the minimal modal logic whose only modal axiom is K. In order to deal with extra axioms, we extend this calculus with rules that allow us to derive the corresponding properties of the accessibility relation.

Example. Earlier we saw that the class of reflexive frames is complete for axiom T. A reflexive frame is characterized by the property that wRw for every world w, so this translates in to a tableaux rule that allows us to add this judgement at any point in a derivation. A possible such rule is

$$\begin{array}{c} \Gamma \\ \Big| \, (T) \\ \Gamma, wRw \end{array}$$

and the previous proof of soundness straightforwardly extends to the calculus with this additional rule and the class of reflexive frames.

As an example, let us prove that $\vdash_{S+T} \Box\Box p \to p$.

$$\dfrac{\underline{\Vdash_w \neg(\Box\Box p \to p)}}{\Vdash_w \Box\Box p, \Vdash_w \neg p}\,(\neg\to)$$

$$\dfrac{\Vdash_w \Box\Box p, \Vdash_w \neg p}{\underline{\Vdash_w \Box\Box p}, \Vdash_w \neg p, \underline{wRw}}\,(T)$$

$$\dfrac{\underline{\Vdash_w \Box\Box p}, \Vdash_w \neg p, \underline{wRw}}{\Vdash_w \Box\Box p, \Vdash_w \neg p, \underline{wRw}, \underline{\Vdash_w \Box p}}\,(\Box)$$

$$\dfrac{\Vdash_w \Box\Box p, \Vdash_w \neg p, \underline{wRw}, \underline{\Vdash_w \Box p}}{\Vdash_w \Box\Box p, \underline{\Vdash_w \neg p}, wRw, \Vdash_w \Box p, \Vdash_w p}\,(\Box)$$

$$\dfrac{\Vdash_w \Box\Box p, \underline{\Vdash_w \neg p}, wRw, \Vdash_w \Box p, \Vdash_w p}{\Vdash_w \Box\Box p, \boxed{\nVdash_w p}, wRw, \Vdash_w \Box p, \boxed{\Vdash_w p}}\,(\neg)$$

Comparing with the corresponding proof in the Hilbert calculus, we see that the tableaux calculus does not require us to instantiate the axiom – making the proof more mechanical and easier to find automatically. ◁

Example. Recall that axiom B is sound and complete for the class of symmetric frames. A possible rule for the tableaux calculus for modal logic that incorporates this axiom is

$$\dfrac{\Gamma, wRw'}{\Gamma, wRw', w'Rw}\,(B)$$

and as before the resulting calculus is sound and complete with respect to the class of symmetric frames.

We show that every instance of B is indeed derivable in this calculus.

$$\dfrac{\underline{\Vdash_w \neg(\varphi \to \Box\Diamond\varphi)}}{\Vdash_w \varphi, \Vdash_w \neg\Box\Diamond\varphi}\,(\neg\to)$$

$$\dfrac{\Vdash_w \varphi, \Vdash_w \neg\Box\Diamond\varphi}{\Vdash_w \varphi, \underline{wRw'}, \Vdash'_w \neg\Diamond\varphi}\,(\neg\Box)$$

$$\dfrac{\Vdash_w \varphi, \underline{wRw'}, \Vdash'_w \neg\Diamond\varphi}{\Vdash_w \varphi, wRw', \underline{w'Rw}, \underline{\Vdash'_w \neg\Diamond\varphi}}\,(B)$$

$$\dfrac{\Vdash_w \varphi, wRw', \underline{w'Rw}, \underline{\Vdash'_w \neg\Diamond\varphi}}{\Vdash_w \varphi, wRw', w'Rw, \Vdash'_w \neg\Diamond\varphi, \underline{\Vdash_w \neg\varphi}}\,(\neg\Diamond)$$

$$\dfrac{\Vdash_w \varphi, wRw', w'Rw, \Vdash'_w \neg\Diamond\varphi, \underline{\Vdash_w \neg\varphi}}{\boxed{\Vdash_w \varphi}, wRw', w'Rw, \Vdash'_w \neg\Diamond\varphi, \boxed{\nVdash_w \varphi}}\,(\neg)$$

Since φ is arbitrary, this proves that $\vdash_{S+B} B$. ◁

Exercise 29. Prove that the following judgements hold.

(a) $\vdash_{S+T} q \to \Diamond q$

(b) $\vdash_{S+B} \Diamond\Box p \to p$

(c) $\vdash_{S+T} \Box\Box p \to ((\Box p \to q) \to \Diamond q)$

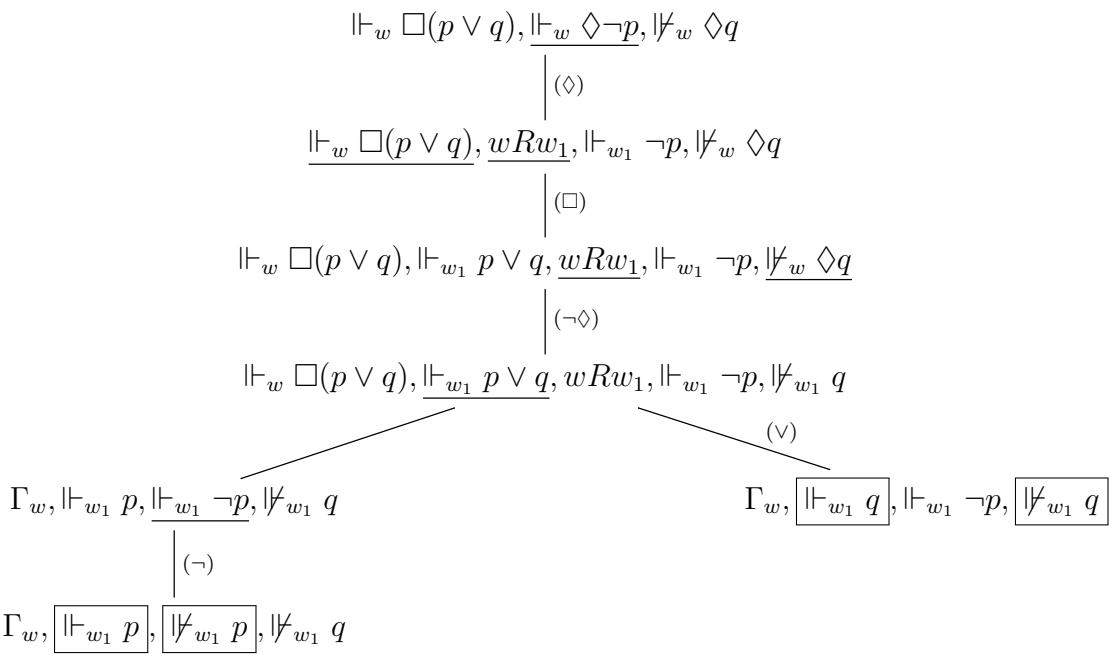
(d) $\vdash_{S+B} \Diamond\Box p \to (q \to \Box\Diamond(p \wedge q))$

Exercise 30. Recall that axiom 4 is sound and complete for the class of all transitive frames (i.e., such that if $w_1 R w_2$ and $w_2 R w_3$, then $w_1 R w_3$). Propose a rule to add to modal tableaux for a modality satisfying 4.

Exercise 31. Recall that axiom 5 is sound and complete for the class of all Euclidean frames (i.e., such that if $w_1 R w_2$ and $w_1 R w_3$, then $w_2 R w_3$). Propose a rule to add to modal tableaux for a modality satisfying 5.

We now show some more complex proofs in the tableaux calculus, involving entailments. These examples also show that finding derivations in this calculus is less mechanical than in the propositional case, as one has to think more carefully about how to use the accessibility relation in a useful way. As a general principle, one should try to apply first rules ($\neg\Box$) and ($\Diamond$), which generate new worlds.

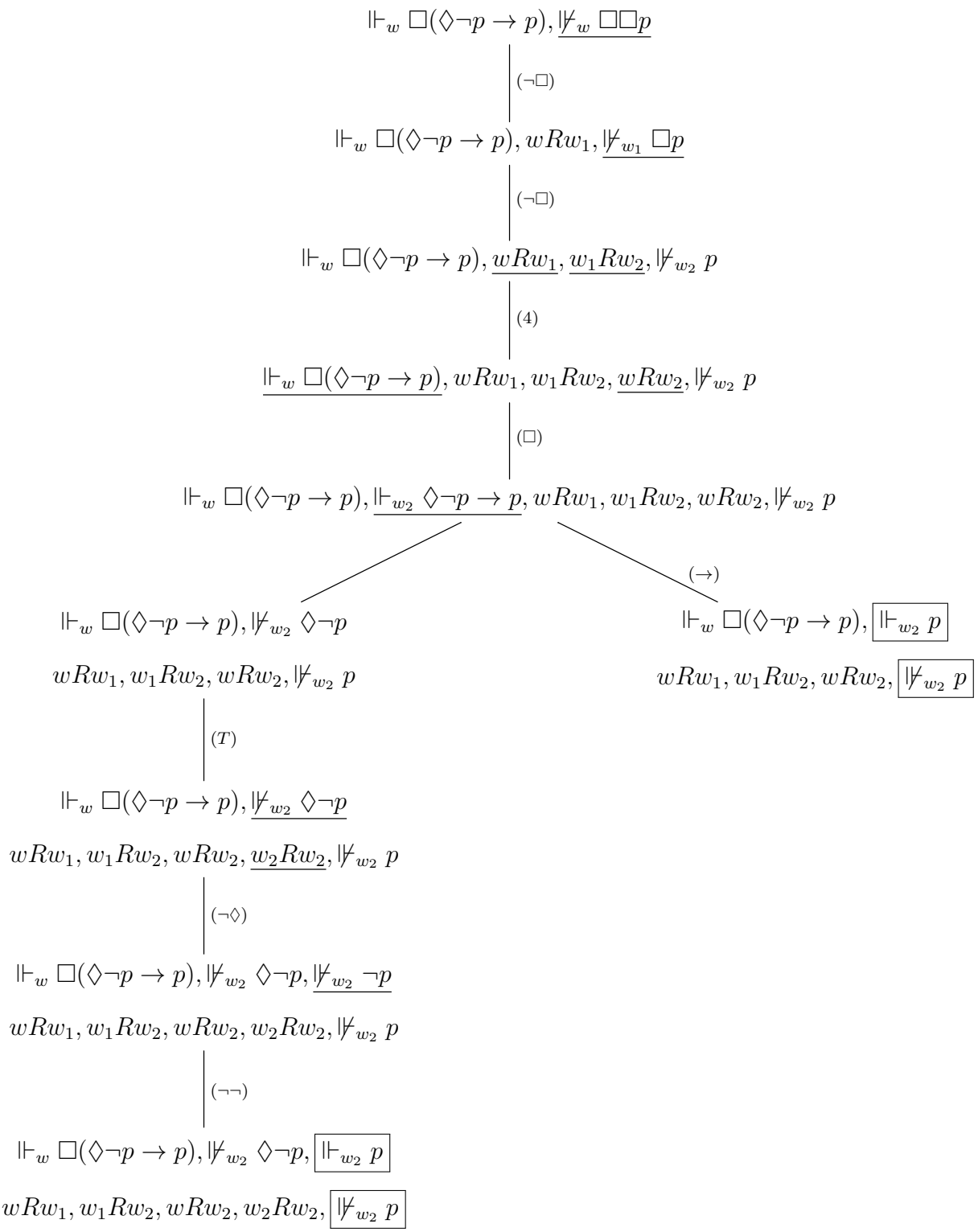
Example. Consider the following tableau for the judgement $\{\Box(p \vee q), \Diamond\neg p\} \vdash_S \Diamond q$. For conciseness, in the lower nodes we abbreviate $\Vdash_w \Box(p \vee q), wRw_1, \nVdash_w \Diamond q$ to Γ_w.



Since all leaves in this tableau are contradictions, we conclude that $\{\Box(p \vee q), \Diamond\neg p\} \vdash_S \Diamond q$. By invoking soundness, we can also conclude that $\{\Box(p \vee q), \Diamond\neg p\} \models \Diamond q$. ◁

Example. Consider the judgement $\{\Box(\Diamond\neg p \to p)\} \vdash_{S+T4} \Box\Box p$. We build a tableau proof for

it as follows.

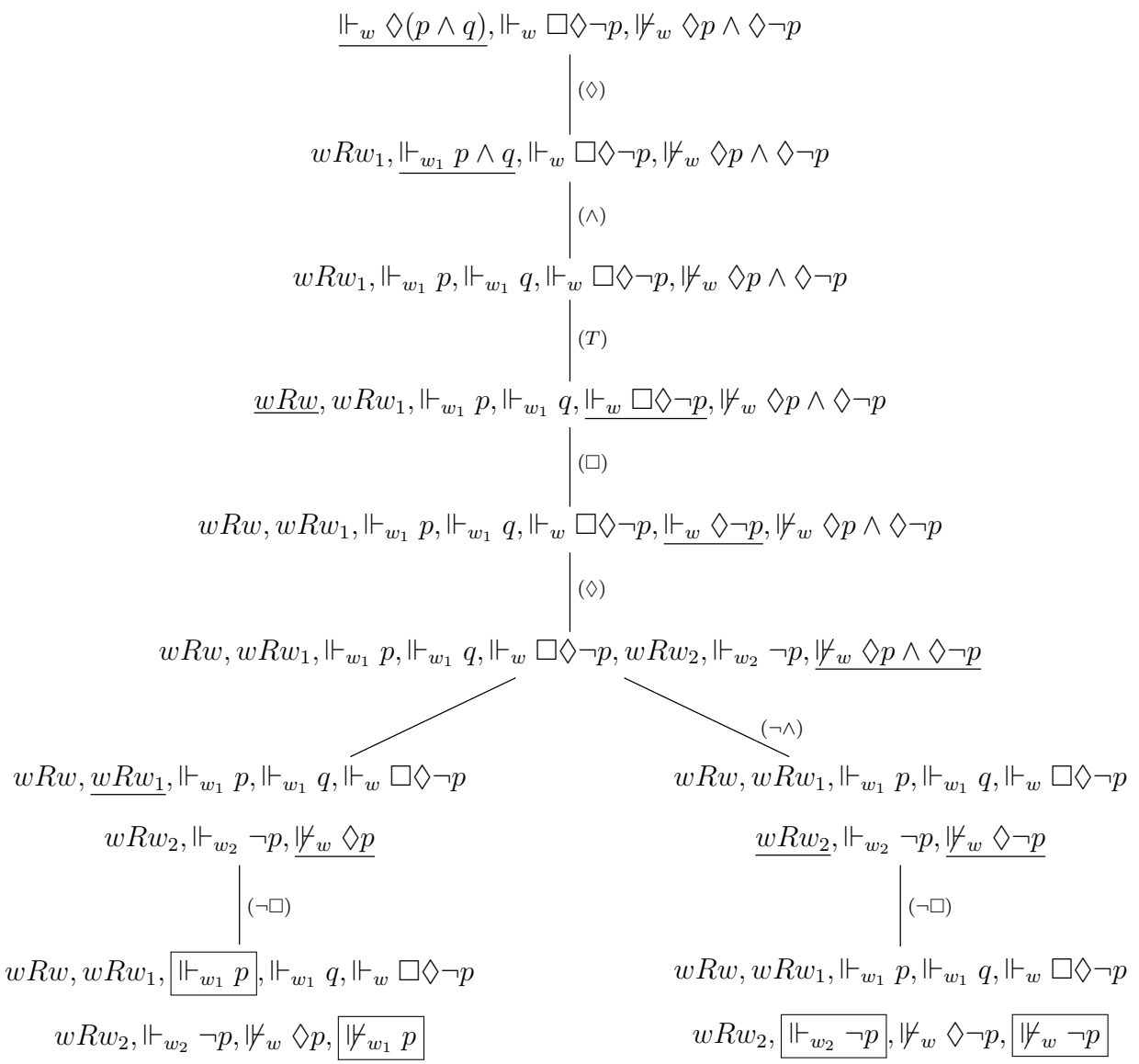


Again we observe that all the leaves in this tableau are contradictory, allowing us to conclude that $\{\Box(\Diamond\neg p \to p)\} \vdash_{S+T4} \Box\Box p$.

As in the previous example, we can also invoke soundness of the tableaux calculus for propositional modal logic to conclude that $\{\Box(\Diamond\neg p \to p)\} \models_{T4} \Box\Box p$. ◁

Example. In the next example we show that $\{\Diamond(p \wedge q), \Box\Diamond\neg p\} \vdash_{S+T} \Diamond p \wedge \Diamond\neg p$. We build the

following tableau proof.



As in the previous examples, all leaves in this tableau are contradictory, and therefore the initial judgement holds. ◁

We now move to situations where building a tableau fails, and examine the construction of a counter-example to the desired entailment.

To build counter-examples in the presence of additional axioms, we must check that the axioms encoding the accessibility relation have been fully applied in order to obtain a model in the appropriate class. This may however yield a non-terminating process, where new worlds are constantly being created. As a consequence, the tableaux calculus for modal logic does not automatically yield a decision procedure for entailment (or even validity) in the presence of axioms, and likewise we do not get a straightforward proof of completeness.

Example. Let us now consider the problem of building a tableau for the judgement $\{\Box(p \to \Diamond q), \Box\Diamond\neg q, \Diamond r\} \vdash_S \Diamond(\neg q \to \neg p)$. In order to save space, we make some abbreviations. We write φ_1 for $\Box(p \to \Diamond q)$, φ_2 for $\Box\Diamond\neg q$ and φ_3 for $\Diamond(\neg q \to \neg p)$. In the lowest nodes of the tree, where these formulas are no longer needed, we abbreviate the list of judgements

$\Vdash_w \varphi_1, \Vdash_w \varphi_2, \nVdash_w \varphi_3, wRw_1, w_1Rw_2$ to Γ.

$$\Vdash_w \varphi_1, \Vdash_w \varphi_2, \underline{\Vdash_w \Diamond r}, \nVdash_w \varphi_3$$
$$\downarrow (\Diamond)$$
$$\Vdash_w \varphi_1, \underline{\Vdash_w \varphi_2}, \underline{wRw_1}, \Vdash_{w_1} r, \nVdash_w \varphi_3$$
$$\downarrow (\Box)$$
$$\Vdash_w \varphi_1, \Vdash_w \varphi_2, \underline{\Vdash_{w_1} \Diamond\neg q}, wRw_1, \Vdash_{w_1} r, \nVdash_w \varphi_3$$
$$\downarrow (\Diamond)$$
$$\underline{\Vdash_w \varphi_1}, \Vdash_w \varphi_2, \Vdash_{w_2} \neg q, w_1Rw_2, \underline{wRw_1}, \Vdash_{w_1} r, \nVdash_w \varphi_3$$
$$\downarrow (\Box)$$
$$\Vdash_w \varphi_1, \Vdash_{w_1} p \to \Diamond q, \Vdash_w \varphi_2, \Vdash_{w_2} \neg q, w_1Rw_2, \underline{wRw_1}, \Vdash_{w_1} r, \underline{\nVdash_w \varphi_3}$$
$$\downarrow (\neg\Diamond)$$
$$\Vdash_w \varphi_1, \Vdash_{w_1} p \to \Diamond q, \Vdash_w \varphi_2, \Vdash_{w_2} \neg q, w_1Rw_2, wRw_1, \Vdash_{w_1} r, \nVdash_w \varphi_3, \underline{\nVdash_{w_1} \neg q \to \neg p}$$
$$\downarrow (\neg\to)$$
$$\Gamma, \underline{\Vdash_{w_1} p \to \Diamond q}, \Vdash_{w_2} \neg q, \Vdash_{w_1} r, \Vdash_{w_1} \neg q, \nVdash_{w_1} \neg p$$

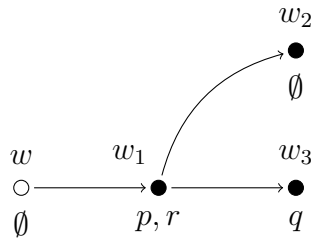
$(\to)$ branches to:

$$\Gamma, \nVdash_{w_1} p, \Vdash_{w_2} \neg q, \Vdash_{w_1} r, \Vdash_{w_1} \neg q, \underline{\nVdash_{w_1} \neg p}$$
$$\downarrow (\neg\neg)$$
$$\Gamma, \boxed{\nVdash_{w_1} p}, \Vdash_{w_2} \neg q, \Vdash_{w_1} r, \Vdash_{w_1} \neg q, \boxed{\Vdash_{w_1} p}$$

and

$$\Gamma, \underline{\Vdash_{w_1} \Diamond q}, \Vdash_{w_2} \neg q, \Vdash_{w_1} r, \Vdash_{w_1} \neg q, \nVdash_{w_1} \neg p$$
$$\downarrow (\Diamond)$$
$$\Gamma, w_1Rw_3, \Vdash_{w_3} q, \Vdash_{w_2} \neg q, \Vdash_{w_1} r, \Vdash_{w_1} \neg q, \underline{\nVdash_{w_1} \neg p}$$
$$\downarrow (\neg\neg)$$
$$\Gamma, w_1Rw_3, \Vdash_{w_3} q, \Vdash_{w_2} \neg q, \Vdash_{w_1} r, \Vdash_{w_1} \neg q, \Vdash_{w_1} p$$

The leaf on the right is not a contradiction, and it can only be expanded further by applying either rule $(\Box)$ to $\Vdash_w \varphi_1$ or $\Vdash_w \varphi_2$, or rule $(\neg\Diamond)$ to $\nVdash_w \varphi_3$, which will generate a descendant equal to itself. From the information in this leaf, we can build the following model.



It is simple to check that this model establishes that $\{\Box(p \to \Diamond q), \Box\Diamond\neg q, \Diamond r\} \not\models \Diamond(\neg q \to \neg p)$, as it satisfies all formulas in the set on the lefthandside of the entailment but falsifies its conclusion. $\triangleleft$

Exercise 32. Decide whether the following judgement holds, and in the negative case use

the tableau to construct a model that falsifies the corresponding entailment.

$$\{\Box(p \to \Box p), \Diamond(p \vee q), \Diamond q \to \Box\Diamond p\} \vdash_S \Diamond\Diamond p$$

Example. Consider now the similar judgement $\{\Box(p \to \Diamond q), \Box\Diamond\neg q, \Diamond r\} \vdash_{S+4} \Diamond(\neg q \to \neg p)$, where we are further requiring that the modality also satisfy axiom 4. The counter-example generated from the previous tableau no longer works, since its accessibility relation is not transitive. (It can be seen that, for example, this model falsifies the instance $\Box p \to \Box\Box p$ of axiom 4.)

Since 4 encodes transitivity of the accessibility relation, a simple rule that implements this axiom in the tableaux calculus is

$$\frac{\Gamma, wRw', w'Rw''}{\Gamma, wRw', w'Rw'', wRw''}\ (4)$$

In order to build a model from the node considered earlier, we need to continue expanding it by first applying rule (4), and then using ($\Box$) with the information newly inferred. We present the first steps in this subderivation; we denote by Γ_1 the set of judgements relating to w_1 and w_3 – $w_1Rw_3, \Vdash_{w_1} r, \Vdash_{w_1} \neg q, \Vdash_{w_1} p, \Vdash_{w_3} q$. We write formula φ_2 explicitly, since it plays a key role in the discussion later.

$$\frac{\Vdash_w \varphi_1, \Vdash_w \Box\Diamond\neg q, \nVdash_w \varphi_3, \underline{wRw_1}, \underline{w_1Rw_2}, \Vdash_{w_2} \neg q, \Gamma_1}{\Vdash_w \varphi_1, \underline{\Vdash_w \Box\Diamond\neg q}, \nVdash_w \varphi_3, wRw_1, w_1Rw_2, \underline{wRw_2}, \Vdash_{w_2} \neg q, \Gamma_1}\ (4)$$

$$\frac{\Vdash_w \varphi_1, \Vdash_w \Box\Diamond\neg q, \underline{\Vdash_{w_2} \Diamond\neg q}, \nVdash_w \varphi_3, wRw_1, w_1Rw_2, wRw_2, \Vdash_{w_2} \neg q, \Gamma_1}{\Vdash_w \varphi_1, \Vdash_w \Box\Diamond\neg q, w_2Rw_4, \Vdash_{w_4} \neg q, \nVdash_w \varphi_3, wRw_1, w_1Rw_2, wRw_2, \Vdash_{w_2} \neg q, \Gamma_1}\ (\Box)\quad (\Diamond)$$

At this stage, we can already see that a problem will arise. By applying (4), world w_2 became accessible from w; but since $\Box\Diamond\neg q$ holds at w, rule ($\Box$) implies that w_2 must satisfy $\Diamond\neg q$, from which we conclude that w_2 must be connected to some other world w_4. We can now repeat the same reasoning for w_4, which will generate a new world w_5, for which the same reasoning must be applied, etc. This leads to an infinite branch in the tableau, and the construction of a model will never terminate. ◁

This example shows that formulas of the type $\Box\Diamond\varphi$ can cause trouble when combined with axioms that allow for extending the accessibility relation to new worlds – notably, 4 and 5. Even more problematic are axioms that generate new worlds on their own, such as D or $4'$.

Example. Axiom D specifies seriality – every world sees at least another world – and we can

encode it with the rule

$$\dfrac{\Gamma}{\Gamma, wRw'}\,(D)$$

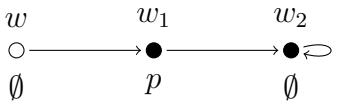
where w is any world already appearing in Γ, there is no judgement of the form wRw'' in Γ, and w' is fresh.

Consider the judgement $\{\Box p\} \vdash_D \Box\Box p$. We can build a tableau for this judgement as follows.

$$\dfrac{\Vdash_w \Box p, \underline{\nVdash_w \Box\Box p}}{\dfrac{\Vdash_w \Box p, wRw_1, \underline{\nVdash_{w_1} \Box p}}{\dfrac{\underline{\Vdash_w \Box p}, \underline{wRw_1}, w_1Rw_2, \nVdash_{w_2} p}{\Vdash_w \Box p, \Vdash_{w_1} p, wRw_1, w_1Rw_2, \nVdash_{w_2} p}\,(\Box)}\,(\neg\Box)}\,(\neg\Box)}$$

The leaf in this tableau does not immediately generate a counter-example for the judgement, because in the suggested model w_2 does not see any other worlds. We need to apply rule (D) to generate a successor for w_2 – but then this world does not see anyone, and we need to apply rule (D) again, and this construction will never finish. ◁

The last example can be easily fixed, though, by connecting w_2 to itself. This yields the model



which clearly falsifies the entailment $\{\Box p\} \models \Box\Box p$.

In general, modal logic has the *finite model property*: if a finite entailment does not hold, then we can build a finite model falsifying it. The proof of this result is beyond the scope of this course notes; in the case of the tableaux calculus, the observation is that if one is "far enough" from the initial world, then all worlds must satisfy the same set of formulas, and one can build a model from a non-contradictory leaf in the tableau by including some loops. Doing this in general is highly non-trivial, though.

Exercise 33. Extend the derivation of $\{\Box(p \to \Diamond q), \Box\Diamond\neg q, \Diamond r\} \vdash_{S+4} \Diamond(\neg q \to \neg p)$ until you can identify regularities in the worlds that are freshly generated, and use this information to build a model falsifying the corresponding entailment.

3.6 Exercises

Exercise 34. For each of the following modalities, discuss whether it should be represented

---

by $\Box$ or $\Diamond$, and which of the axioms K, T, D, B, 4 and 5 it should satisfy.

(a) There is a mathematical proof that...

(b) It is my opinion that...

(c) John believes that...

(d) It is reasonable to assume that...

(e) It is legal to...

(f) Executing this program leads to a state where...

(g) It is widely believed that...

(h) It is conceivable that...

(i) It is true that...

Exercise 35. For each of the following formulas, find a model that satisfies them and a model that falsifies them, or prove that none exist.

(a) $\Diamond(p \vee \Box q) \vee \Box(\Diamond p \wedge q)$

(b) $\Diamond(p \vee \Box q) \wedge \Box(\Diamond p \wedge q)$

(c) $\Diamond(p \vee \Box q) \rightarrow \Box(\Diamond p \wedge q)$

(d) $\Box\Box p \rightarrow \Box p$

Exercise 36. Write the duals of the formulas in the previous exercise and check whether they still hold in the models you constructed.

Exercise 37. Let $\mathcal{F}$ be the class of all serial frames (i.e., for every w there is w' such that wRw').

(a) Show that $\mathcal{F} \models D$.

(b) Show that $\mathcal{F}$ is complete for D.

(c) Show that rule (D) given in the text is an appropriate rule to add to modal tableaux for a modality satisfying D.

Exercise 38. Let $\mathcal{F}$ be the class of all convergent frames (i.e., such that if w_1Rw_2 and w_1Rw_3, then there exists w_4 such that w_2Rw_4 and w_3Rw_4).

(a) Show that $\mathcal{F} \models C$, where C is the axiom $\Diamond\Box\varphi \rightarrow \Box\Diamond\varphi$.

(b) Show that $\mathcal{F}$ is complete for C.

(c) Propose a rule to add to modal tableaux for a modality satisfying C.

Exercise 39. Let $\mathcal{F}$ be the class of all dense frames (i.e., such that if w_1Rw_2, then there exists w_3 such that w_1Rw_3 and w_3Rw_2).

(a) Show that $\mathcal{F} \models 4'$, where $4'$ is the axiom $\Box\Box\varphi \to \Box\varphi$.

(b) Show that $\mathcal{F}$ is complete for $4'$.

(c) Propose a rule to add to modal tableaux for a modality satisfying $4'$.

Exercise 40. Let $\mathcal{F}$ be the class of all functional frames (i.e., such that if w_1Rw_2 and w_1Rw_3, then $w_2 = w_3$).

(a) Show that $\mathcal{F} \models D'$, where D' is the axiom $\Diamond\varphi \to \Box\varphi$.

(b) Show that $\mathcal{F}$ is complete for D'.

(c) Propose a rule to add to modal tableaux for a modality satisfying D'.

Exercise 41. Prove the following judgments, if necessary using the Deduction Theorem.

(a) $\vdash_{L+D} \Box\Box p \to \Box\Diamond p$

(b) $\vdash_{L} (\Diamond p \wedge \Box q) \to \Diamond(p \to q)$

(c) $\vdash_{L+S5} \Diamond(p \vee \Box q) \to \Box(\Diamond p \wedge q)$

(d) $\vdash_{L} \Diamond(p \vee q) \to \Diamond p \vee \Diamond q$

(e) $\vdash_{L+4} \Diamond\Diamond p \to \Diamond p \vee \Diamond q$

(f) $\vdash_{L+D'} \Diamond p \to (\Diamond\neg p \to \Box\bot)$

Exercise 42. For each formula in Exercise 35, construct tableaux proving or disproving it in the basic system, in system KBD, and in $S5 = KT45$.

Exercise 43. Find tableaux proofs of all the judgements in Exercise 41.

Chapter 4

First-Order Logic

First-order logic is the widely used logic among non-logicians. Indeed, for many people, including mathematicians and computer scientists, the term "logic" simply refers to this logic, which is also known as predicate logic or predicate calculus.

This logic extends propositional logic by adding structure at the lowest level of formulas. Instead of being propositional symbols with no internal structure, the simplest formulas are now built from predicates. Furthermore, the presence of variables and constants yields the novel possibility of referring to individuals, both to particular instances or in general. As such, in first-order logic one can syntactically express connections between formulas that are impossible in propositional logic – for example, to say that different individuals share a common property, or that several different properties hold of a given individual.

First-order logic is a logic in which it is very natural to model the real world and reason about it, and, for this reason, it is historically the oldest logic that has been studied.

4.1 Language

Syntactically, first-order logic differs from propositional logic in two major aspects: basic formulas now have a non-trivial structure, and there is a new type of unary connective – the quantifiers, which allow parametric conjunction and disjunction over formulas.

4.1.1 Signatures, terms and formulas

The simplest formulas in first-order logic are called *atomic formulas*, and play the role of propositional symbols in propositional logic. These formulas allow us to express the simplest statements that can be true or false, for example "the car is red", or "the temperature is 13 ºC". The precise choice of what atomic formulas should look like depends on the particular domain we are interested in modeling, and needs to be agreed upon beforehand. This is the role of the *signature*.

Definition. A *first-order signature* is a pair $\langle \mathcal{F}, \mathcal{P} \rangle$, where:

- $\mathcal{F} = \{F_n \mid n \in \mathbb{N}\}$, where each F_n is a countable set whose elements are called *function symbols* with n arguments;

- $\mathcal{P} = \{P_n \mid n \in \mathbb{N}\}$, where each P_n is a countable set whose elements are called *predicate symbols* or *relation symbols* with n arguments.

The elements of F_0 are usually called *constants*.

We also need a countable set of *variables*, which is not part of the signature. We assume that this set is fixed, and refer to it as $\mathcal{X}$.

Function symbols correspond to operations on the objects that we want to work with, much in the same way that functions are used in mathematics. For example, if we were interested in modeling family relations, we might want to include a unary function symbol father, mapping each person to their father. In particular, 0-ary functions take no arguments: they simply represent elements of the domain.

Predicate symbols are used to state that relationships between elements hold, similar to relations in mathematics. For example, in the context of modeling family relations we could use a binary predicate symbol $\mathsf{sibling}$ to express that two people are siblings.

Example F.1. Suppose we want to model family relationships among people, where in particular we want to be able to talk about three individuals Alice, Bob and Carol. The following signature Σ_F covers some of the properties that we might want to model.

$$\begin{array}{lll} F_0 = \{\mathsf{Alice}, \mathsf{Bob}, \mathsf{Carol}\} & F_1 = \{\mathsf{father}, \mathsf{mother}\} & F_n = \emptyset, \quad n \geq 2 \\ P_1 = \{\mathsf{male}, \mathsf{female}\} & P_2 = \{\mathsf{sibling}, \mathsf{married}\} & P_0 = P_n = \emptyset, \quad n \geq 3 \end{array}$$

Using this signature, we can express that someone is a man or a woman, map people to their parents, or state that two people are married. ◁

The choice of signature is a design option: we could as well have used binary relations father or mother instead of function symbols. The advantage of using function symbols is that they immediately guarantee that everyone has a father, and that that father is unique.

Example M.1. A *monoid* is an algebraic structure consisting of an associative operation with a special element, known as *identity* or *unit*. Monoids are used in many different contexts, both in Mathematics and Computer Science. A possible signature Σ_M for working with monoids is

$$\begin{array}{llll} F_0 = \{\star\} & F_1 = \emptyset & F_2 = \{\cdot\} & F_n = \emptyset, n \geq 3 \\ P_0 = \emptyset & P_1 = \emptyset & P_2 = \{\mathsf{eq}\} & P_n = \emptyset, n \geq 3 \end{array}$$

In this signature, we can write expressions involving the monoid operation $\cdot$ and its identity $\star$, as well as equalities between such expressions. ◁

Example N.1. Suppose we want to write formulas about mathematics on natural numbers. The following signature Σ_N allows us to express some of the concepts we might want to model.

$$\begin{array}{llll} F_0 = \mathbb{N} & F_1 = \{\mathsf{succ}\} & F_2 = \{+, \times\} & F_n = \emptyset, \quad n \geq 3 \\ P_0 = \emptyset & P_1 = \{\mathsf{even}, \mathsf{odd}\} & P_2 = \{=, \leq\} & P_n = \emptyset, \quad n \geq 3 \end{array}$$

In this language we can represent every natural number directly, but we can also write down operations on them. The intended meaning of the unary operation succ is "successor" (the operation of adding one). ◁

These signatures also illustrate the difference between function and predicate symbols: $\mathsf{father}(\mathsf{Carol})$ denotes Carol's father (an individual), while $\mathsf{sibling}(\mathsf{Alice}, \mathsf{Bob})$ states that Alice

and Bob are siblings (a statement that may be true or false). Likewise, $\mathsf{succ}(3)$ denotes an object we can talk about (presumably the natural number 4), while $\mathsf{even}(3)$ intuitively states that 3 is even (which we expect to be false, but from a syntactic point of view could also be true).

As we will see below, function symbols allow us in general to construct objects from other objects, while predicate symbols create statements that have a truth value from such objects.

Exercise 1. Consider the following problem domains. For each of them, propose a relevant first-order signature. Pay special attention to concepts that can be represented both by function symbols or predicate symbols.

1. A signature Σ_C for the domain of cars, where we want to reason about make, color, production year, and license plates.
2. A signature Σ_S for the domain of set theory, where we want to talk about the empty set, unions, intersections, complements and equality of sets, and membership in a set.
3. A signature Σ_G for the domain of graphs, where we want to have elements that can be vertices or edges, and talk about paths between vertices, or a set of vertices being connected.

As in the previous examples, the sets F_n and P_n are all assumed to be mutually disjoint, and disjoint from $\mathcal{X}$. Some authors also require $P_0 = \emptyset$: since predicate symbols are used to express properties of objects, it may seem strange to have predicate symbols that do not take any arguments, and as such have a fixed truth value. However, by allowing 0-ary predicate symbols we obtain propositional logic as a particular case of first-order logic – which is convenient for some applications.

We make some syntactic conventions to alleviate the presentation. Unless otherwise stated, we use lowercase letters from the beginning of the alphabet (a, b, c, a_1, c', etc.) to denote constants, and lowercase letters from the end of the alphabet (x, y, z, w, x_1, y', etc.) to denote variables. For function symbols we use f, g, h, f', g_1, etc., and for predicate symbols we use p, q, r, p', q_1, etc.. When we want to make the arity of a function of predicate symbol explicit, we write it as a superscript, as in f^2 or p^5.

The separation between function symbols and predicate symbols is reflected in the layered-way that the language of first-order logic is defined. Constants, variables and function symbols are used to build *terms* – syntactic elements that refer to the objects in the domain, as $\mathsf{father}(\mathsf{Carol})$. Predicate symbols are used to build *atomic formulas* – syntactic elements that refer to properties of those objects, such as $\mathsf{female}(\mathsf{Carol})$.

Definition. Let Σ be a first-order signature. Terms t and formulas φ over Σ are inductively defined by the following grammars.

$$t ::= c \mid x \mid f^n(t_1, \ldots, t_n)$$
$$\varphi ::= p^n(t_1, \ldots, t_n) \mid \neg\varphi \mid \varphi \to \varphi \mid \forall x \varphi$$

In other words: every constant $c \in F_0$ is a term; every variable $x \in \mathcal{X}$ is a term; and if $t_1, \ldots, t_n$ are terms and $f \in F_n$, then $f(t_1, \ldots, t_n)$ is a term. Atomic formulas are built from terms: if $t_1, \ldots, t_n$ are terms and $p \in P_n$, then $p(t_1, \ldots, t_n)$ is an atomic formula. For simplicity, we again assume that the propositional connectives $\wedge$, $\vee$ and $\leftrightarrow$ are defined as abbreviations.

As a new ingredient, we have *quantified* formulas: if φ is a formula and $x \in X$, then $(\forall x\varphi)$ is a formula, read "for all x, φ". The symbol $\forall$ is also called the *universal quantifier*. The *existential quantifier* $\exists$ is defined as an abbreviation: $(\exists x\varphi)$, read "there exists an x such that φ", stands for $\neg(\forall x\neg\varphi)$. For readability, we often add a dot separating the quantified variable from the remainder of the formula, writing $\forall x.\varphi$ and $\exists x.\varphi$. It is also customary to merge identically quantified variables, writing e.g. $\forall xyz.\varphi$ instead of $\forall x\forall y\forall z\varphi$. The concrete choice of notation depends on the context: for abstract reasoning over formulas in proofs, the precise notation is typically used, but in examples these two alternatives are often preferred.

Example F. 2. Recall the signature Σ_F for family relations. Examples of terms over this signature, together with their intuitive interpretation, are:

- Alice
- $\mathsf{father}(\mathsf{Alice})$, corresponding to Alice's father;
- $\mathsf{mother}(\mathsf{father}(\mathsf{Alice}))$, corresponding to Alice's father's mother – i.e., Alice's paternal grandmother;
- x, corresponding to an unspecified element of the domain;
- $\mathsf{mother}(\mathsf{mother}(x))$, corresponding to x's maternal grandmother.

Possible formulas over this signature are:

- $\mathsf{female}(\mathsf{Alice})$, stating that Alice is a woman;
- $\mathsf{male}(\mathsf{father}(\mathsf{Alice}))$, stating that Alice's father is a man;
- $\mathsf{male}(\mathsf{father}(x))$, stating that x's father is a man (for some unknown x);
- $\forall x.\mathsf{male}(\mathsf{father}(x))$, stating that x's father is a man, independent of which individual x represents;
- $\forall x.(\mathsf{married}(\mathsf{Bob}, x) \to \mathsf{married}(x, \mathsf{Bob}))$, stating that if Bob is married to someone, then that person is also married to Bob. ◁

Example M.2. Using now the signature Σ_M, the following are examples of terms that we can write:

- $\star$, corresponding to the identity of the monoid;
- $\cdot(x, y)$, the result of applying the monoid operation to the elements denoted by x and y, which we will write more suggestively as $x \cdot y$;
- $\cdot(x, \cdot(\star, x))$, or simply $x \cdot (\star \cdot x)$.

Examples of formulas are:

- $\mathsf{eq}(x, x)$, which we write simply as $x \;\mathsf{eq}\; x$, stating that x is equal to itself;
- $\forall xy.\; \mathsf{eq}\; (\cdot(x, y), \cdot(y, x))$, or simply $\forall x.(x \cdot y \;\mathsf{eq}\; y \cdot x)$, stating that $\cdot$ is commutative;
- $\forall xyz.(x \cdot (y \cdot z) \;\mathsf{eq}\; (x \cdot y) \cdot z)$, stating that $\cdot$ is associative;

- $\forall x.(x \cdot \star \;\mathsf{eq}\; x)$, stating that $\star$ is a right identity with respect to $\cdot$;
- $\forall x \exists y.(x \cdot y \;\mathsf{eq}\; \star)$, stating that, given an element of the monoid, we can find another element such that when we apply $\cdot$ to them we get $\star$;
- $\exists y \forall x.(x \cdot y \;\mathsf{eq}\; \star)$, stating that there exists an element of the monoid that, when combined with any other element via $\cdot$, results in $\star$. ◁

It is instructive to understand the difference between the last two formulas.

Example N.2. We now show some examples of terms and formulas over signature Σ_N. As usual when working with mathematics, we often write binary functions and predicates in infix notation:

- $\mathsf{succ}(\mathsf{0})$, the successor of 0;
- $+(\mathsf{3}, \mathsf{5})$, the result of adding 3 and 5 – we will usually write this term as $\mathsf{3} + \mathsf{5}$;
- $\mathsf{8}$, the number 8: this is *not* the same term as the previous one, since terms are syntactic entities;
- $\times(+(\mathsf{x}, \mathsf{4}), \mathsf{3})$, or $(\mathsf{x} + \mathsf{4}) \times \mathsf{3}$;
- $\times(\mathsf{3}, +(\mathsf{x}, \mathsf{4}))$, or $\mathsf{3} \times (\mathsf{x} + \mathsf{4})$ – again, this term is *not* the same as the previous one, although we possibly expect it to represent the same object.

Examples of formulas we could write are:

- $\mathsf{even}(\mathsf{2})$, stating that the number 2 is even;
- $=(+(\mathsf{3}, \mathsf{5}), \mathsf{8})$, or more suggestively $\mathsf{3} + \mathsf{5} = \mathsf{8}$;
- $\forall n. \leq (n, \mathsf{succ}(n))$ – which we write as $\forall n.(n \leq \mathsf{succ}(n))$;
- $\forall y.(\mathsf{3} + y = y + \mathsf{3})$, stating that adding any quantity to 3 is the same as adding 3 to that same quantity;
- $\forall xyz.(x + (y \times z) = (x + y) \times z)$, stating a novel relationship between addition and multiplication that does not usually hold. ◁

Being able to express and understand a formula's meaning using only natural language (as in these examples: without explicitly mentioning variable names) is also an important skill that helps immensely in reasoning about first-order logic.

Exercise 2. Using the signature Σ_C that you defined in Exercise 1, write formulas stating the following properties.

(a) All red cars have number plates.

(b) If two cars have the same number plate, then they must be from the same year.

(c) There exists a blue car from 1993.

(d) Every car model produced in red has also been produced in green.

(e) No green cars were produced in 1995.

Exercise 3. Using the signature Σ_S that you defined in Exercise 1, write formulas stating the following properties.

(a) The empty set has no elements.

(b) The union of any set with the empty set is itself.

(c) An element in the intersection of two sets is also in both original sets.

(d) Set intersection distributes over set union.

(e) There is a set whose elements are elements of all other sets.

Exercise 4. Using the signature Σ_G that you defined in Exercise 1, write formulas stating the following properties.

(a) v is a vertex.

(b) There exists an edge between v_1 and v_2.

(c) The graph is undirected, i.e., if there is an edge between v_1 and v_2, then there is an edge between v_2 and v_1.

(d) The graph is complete, i.e., there exists an edge between any pair of vertices.

(e) There is a path between two vertices.

4.1.2 Variables

The use of variables and quantifiers requires some care. The syntax of first-order logic allows us to write formulas such as $\forall x \exists x.p(x, x)$, or $p(x) \rightarrow \forall x.q(x)$, which are very confusing to read: in the first formula, are the two occurrences of x existentially or universally quantified? In the second formula, is there any connection between the two xs? Although the language is precisely defined and we can answer these questions unequivocally based on the semantics, it is convenient to avoid writing such unintuitive formulas. In the next paragraphs, we define some concepts related to variables, and introduce some useful conventions.

Definition. The set of variables $V(t)$ in a term t is defined as follows.

$$\begin{aligned} V(c) &= \emptyset && c \in F_0 \\ V(x) &= \{x\} && x \in \mathcal{X} \\ V(f(t_1, \ldots, t_n)) &= V(t_1) \cup \ldots \cup V(t_n) && f \in F_n \end{aligned}$$

Definition. The set of *free variables* in a formula is also inductively defined.

$$\begin{aligned} FV(p(t_1,\ldots,t_n)) &= V(t_1)\cup\ldots\cup V(t_n), \quad p\in P_n \\ FV(\neg\varphi) &= FV(\varphi) \\ FV(\varphi\to\psi) &= FV(\varphi)\cup FV(\psi) \\ FV(\forall x\varphi) &= FV(\varphi)\setminus\{x\} \end{aligned}$$

Variables that occur in a formula and are not free are called *bound* variables. A formula with no free variables is also called a *closed* formula or a *sentence*.

The notions of free and bound variable are also useful when considering distinct occurrences of the same variable in a formula, so that we can say, for example, that a particular occurrence of x is free (see the next example). The *range* of quantifier $\forall x$ in $\forall x\varphi$ (or of $\exists x$ in $\exists x\varphi$) is φ, and the occurrences of x occurring in φ are said to be bound *by* $\forall x$ (or $\exists x$). Intuitively, each variable is in the range of the quantifier that is "closest" to it.

Example. In the formulas below, all free occurrences of variables are underlined.

$$p(\underline{x},\underline{y}) \qquad \forall x.p(x,\underline{y}) \qquad \forall xy.p(x,y) \qquad p(\underline{x},\underline{y})\to\forall x.p(x,\underline{y})$$

In the last formula, the first occurrence of x is free, and the second occurrence of x is bound; so variable x occurs both free and bound in that formula.

In the formula $\forall x.(p(x)\to\exists x.q(x))$, the first occurrence of x is in the range of the universal quantifier, whereas the second occurrence is in the range of the existential quantifier (even though it occurs in a subformula of the universal quantified formula):

$$\forall\ \underbrace{x}\ .\ (\ p\ (\ \underbrace{x}\ )\ \to\ \exists\ \underbrace{x}\ .\ q\ (\ \underbrace{x}\ )\ )$$

Thus, both occurrences of x are bound, but they are unrelated. ◁

Exercise 5. For each of the following formulas, indicate which variables are free and which variables are bound. For the latter, indicate which quantifier is binding them.

(a) $p(x)$

(b) $\forall x.q(x,y)$

(c) $\exists y.(p(x)\wedge q(x,y))\to\forall z.q(z,y)$

(d) $(\forall x.p(x))\vee(\exists y.q(x,y))$

(e) $\forall x.(p(x)\vee\exists y.q(x,y))$

There is a standard convention, known as the *variable convention*, which we will adhere to when writing first-order formulas: no variable occurs simultaneously free and bound in the same formula or in any of its subformulas. This disallows not only writing $p(x,y)\to\forall x.p(x,y)$, where x occurs both free and bound, but also $\forall x.(p(x)\to\exists x.q(x))$, since x occurs free and bound in the subformula $p(x)\to\exists x.q(x)$. In general, we are not allowed to quantify over a

variable that is already bound in the range of the quantifier. However, this convention does not exclude e.g. $(\forall x.p(x)) \to (\forall x.q(x))$, since the two quantifications on x are on non-overlapping subformulas; and indeed, such formulas often occur in practice.

Given any formula, it is always possible to find an equivalent one that satisfies the variable convention.

Example. The formula $p(x,y) \to \forall x.p(x,y)$ is intuitively equivalent to $p(x,y) \to \forall z.p(z,y)$, which satisfies the variable convention. Likewise, $\forall x.(p(x) \to \exists x.q(x))$ is intuitively equivalent to $\forall x.(p(x) \to \exists y.q(y))$.

When we have formally defined the semantics of first-order logic, we will be able to show that these pairs of formulas are indeed logically equivalent, and thus it is possible to work with the versions that satisfy the variable convention. ◁

Exercise 6. Which of the formulas in previous exercise do not respect the variable convention? For each of those formulas, find an equivalent one that does respect that convention.

The final notion related to variables is one that is central to first-order logic: capture-avoiding substitution.

Definition. The *substitution* of variable x by term t in formula φ, denoted $\varphi[x/t]$, is defined inductively for terms by

$$x[x/t] = t,\ x \in \mathcal{X} \qquad c[x/t] = c,\ c \in F_0$$
$$y[x/t] = y,\ y \in \mathcal{X},\ y \neq x \qquad f(t_1,\ldots,t_n)[x/t] = f\,(t_1[x/t],\ldots,t_n[x/t])$$

and for formulas by

$$p(t_1,\ldots,t_n)[x/t] = p\,(t_1[x/t],\ldots,t_n[x/t]) \qquad (\neg\varphi)[x/t] = \neg\varphi[x/t]$$
$$(\varphi \to \psi)[x/t] = \varphi[x/t] \to \psi[x/t] \qquad (\forall x\varphi)[x/t] = \forall x\varphi$$
$$(\forall y\varphi)[x/t] = \forall y(\varphi[x/t]),\ y \neq x$$

Intuitively, $\varphi[x/t]$ is obtained by simultaneously replacing all free occurrences of x by t. This is made explicit in the last two cases of the definition, where a substitution on x is not applied inside any quantifier whose variable is x.

Example. Let φ be the formula $p(x,y) \to \forall z.p(z,y)$. The result of replacing x by b, $\varphi[x/b]$, is $p(b,y) \to \forall z.p(z,y)$. Likewise, $\varphi[y/f(a)]$ is $p(x,f(a)) \to \forall z.p(z,f(a))$. The result of replacing z by anything is still φ, since z does not occur free in φ. ◁

Exercise 7. Compute $\varphi[x/a]$ and $\varphi[y/g(x,x)]$, where φ is each of the formulas in Exercise 5.

The previous definition does not exclude that, if t contains free variables, some of their occurrences may become bound by quantifiers in φ. In the previous example, this would happen if we considered $\varphi[y/z]$: we would obtain $p(x,z) \to \forall z.p(z,z)$. The fact that this formula no longer satisfies the variable convention is a hint that something is wrong – but the same problem would occur in e.g. $(\forall x.p(x,y))[y/x]$.

In order to characterize when a substitution is safe, in the sense that no occurrence of the variable we are substituting is in the range of a quantifier over a variable that occurs in the term we are substituting it with, we introduce another notion.

Definition. A term t is *free* for a variable x in φ, denoted $t \triangleright x : \varphi$, if:

- φ is an atomic formula;
- φ is $\neg\psi$ and $t \triangleright x : \psi$;
- φ is $\psi \to \theta$, $t \triangleright x : \psi$ and $t \triangleright x : \theta$;
- φ is $\forall x\psi$;
- φ is $\forall y\psi$ with $y \neq x$, $y \notin V(t)$, and $t \triangleright x : \psi$.

If $t \triangleright x : \varphi$, then the substitution of x by t in φ is said to be *capture-avoiding*.

Example. Letting φ be the formula $p(x, y) \to \forall z.p(z, y)$ as in the previous example, we have that $b \triangleright x : \varphi$, and indeed that $t \triangleright x : \varphi$ for any t, since x does not occur in the range of any quantifier in φ. Also $t \triangleright z : \varphi$ for any t, this time because z does not occur free in φ.

However, $f(a) \triangleright y : \varphi$, but $z \not\triangleright y : \varphi$, since y occurs in the range of a quantifier on z. ◁

Exercise 8. Decide whether $a \triangleright x : \varphi$, $z \triangleright x : \varphi$ and $g(x, x) \triangleright y : \varphi$, where φ is each of the formulas in Exercise 5.

Exercise 9. Show that $x \triangleright x : \varphi$ for any formula φ.

4.2 Semantics

Following the syntactic structure of first-order logic, its semantics is a two-layered semantics, where a first layer is concerned with the interpretation of terms (which requires making decisions about constants, variables and function symbols), and the second layer with the interpretation of atomic formulas. Thereafter the semantics of more complex formulas is defined recursively, as usual, with the propositional connectives being treated as in the propositional case. The semantics of quantifiers bears some similarity to that of modalities in modal logic, which we discuss below.

4.2.1 Interpretations, assignments and satisfaction

In order to be able to give meaning to atomic formulas, we need two ingredients: an *interpretation*, which provides the semantics to the symbols defined by the signature, and an *assignment*, which deals similarly with the variables. Intuitively, the interpretation specifies the semantics of the parts of the language that are fixed, while the assignment is used for the parts whose

meaning can change. Using these two components, we first define the semantics of terms, and thereafter assign a truth value to every formula.

Every interpretation includes a domain, which is the set of objects we want to talk about. Constants are then mapped to particular elements of the domain, function symbols to functions (of the expected arity) over the domain, and predicate symbols to sets of tuples of the domain (or, equivalently, to relations over the domain).

Definition. A *first-order interpretation structure*, or simply *interpretation*, over a signature Σ is a pair $I = \langle D, \cdot^I \rangle$ where:

- D is a non-empty set, called the *domain*;
- for each constant c, $c^I \in D$;
- for each function symbol $f \in F_n$, $f^I : D^n \to D$;
- for each predicate symbol $p \in P_n$, $p^I \subseteq D^n$.

Example F.3. A possible interpretation I_F over the signature Σ_F for our example of family relations, following the intuition that we gave earlier, could be:

- the domain D_F is the set of all humans who have ever lived;
- Alice^{I_F}, Bob^{I_F} and Carol^{I_F} are three concrete individuals;
- father^{I_F} and mother^{I_F} are the functions mapping each individual to their father or mother, respectively;
- male^{I_F} and female^{I_F} are the sets of all men and women, respectively;
- $\mathsf{sibling}^{I_F}$ is the set of all pairs of humans who are each other's siblings;
- $\mathsf{married}^{I_F}$ is the set of all pairs of humans who are married to each other. ◁

Example M.3. A possible interpretation I_M over the signature of monoids is to take as domain the set of all strings over a fixed alphabet – for example, $\{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$:

- the domain is $D_M = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}^*$, the set of all finite sequences of characters from the alphabet;
- $\star^{I_M} = \epsilon$ is the empty string;
- $\cdot^{I_M}$ is the concatenation operation, $\omega \cdot^{I_M} \omega' = \omega\omega'$;
- $\mathsf{eq}^{I_M} = \{\langle \omega, \omega \rangle \mid \omega \in D_M\}$ is the diagonal relation on D_M. ◁

Example M.4. Another possible interpretation J_M over the signature of monoids is the structure often known as *modular arithmetic* (modulo 7):

- the domain is the set $\mathbb{Z}_7 = \{0, 1, 2, 3, 4, 5, 6\}$ of natural numbers smaller than 7;
- $\star^{J_M} = 0$;

- $\cdot^{J_M} = \lambda jk.(j+k) \bmod 7$ is addition modulo 7, i.e., $j \cdot^{J_M} k = \begin{cases} j+k & \text{if } j+k < 7 \\ j+k-7 & \text{otherwise.} \end{cases}$
- $\mathsf{eq}^{J_M} = \{\langle k, k\rangle \mid 0 \leq k < 7\}$. ◁

The number 7 plays no special role in the above example – we could just as well have used 25, 42, or any other natural number. The set of all natural numbers with addition would also be a possible interpretation over this signature.

Example N.3. Likewise, we can define the expected interpretation I_N over the signature Σ_N for our example of mathematical formulas as:

- the domain D_N is the set of natural numbers;
- each constant $n \in F_0$ is interpreted to itself, i.e., $n^{I_N} = n$;
- succ^{I_N} is the successor function, assigning x to $x+1$;
- $+^{I_N}$ and $\times^{I_N}$ are interpreted as the usual sum and product of natural numbers;
- even^{I_N} and odd^{I_N} are the sets of even and odd numbers, respectively;
- $=^{I_N}$ is the set of all pairs $\langle n, n\rangle$, with n a natural number, and $\leq^{I_N}$ is the set of all pairs $\langle m, n\rangle$ where $m \leq n$. ◁

Exercise 10. Write down example interpretations for your signatures from Exercise 1.

We will represent functions in λ-notation, as is standard in several areas of Computer Science. In the context of the last example, we would write for example $\mathsf{succ}^{I_N} = \lambda x.x+1$, where the expression $\lambda x.x+1$ should be read as "the function that maps each x to $x+1$". λ-notation is a convenient way to talk about functions defined by their computational behaviour; in particular, it abstracts from the variables used: the function $\lambda x.x+1$ is the same as the function $\lambda y.y+1$.

When working with first-order logic, it is common to choose intuitive names for function and predicate symbols – as we have done in our examples so far. However, it is important to remember that these names are just part of the syntax, and we can define interpretations that completely violate this intuition.

Example F.4. The following interpretation J_F is also an interpretation over Σ_F.

- $D'_F = \mathbb{Z}$;
- $\mathsf{Alice}^{J_F} = 1$, $\mathsf{Bob}^{J_F} = 25$ and $\mathsf{Carol}^{J_F} = 42$;
- $\mathsf{father}^{J_F} = \lambda x.x+1$ and $\mathsf{mother}^{J_F} = \lambda x.x-1$ are the successor and predecessor functions;
- $\mathsf{male}^{J_F} = \{0, 1\}$ and $\mathsf{female}^{J_F} = \{1, 2, 3\}$;
- $\mathsf{sibling}^{J_F} = \{\langle m, n\rangle \mid m+n = 50\}$ and $\mathsf{married}^{J_F} = \emptyset$. ◁

Example N.4. Another possible interpretation over Σ_N is the following interpretation J_N:

- $D'_N = \mathbb{N}$;
- $n^{J_N} = 2n + 3$;
- $\mathsf{succ}^{J_N} = \lambda x.0$ is the constant function mapping every number to 0;
- $+^{J_N} = \lambda xy.y + 5$ is the function adding 5 to its second argument, and $\times^{J_N} = \lambda xy.x + y$ returns the sum of its two arguments;
- $\mathsf{even}^{J_N} = \{0\}$ and $\mathsf{odd}^{J_N} = \mathbb{N}$;
- $=^{J_N} = \{\langle n, n\rangle \mid n \in \mathbb{N}\}$ and $\leq^{J_N} = \{\langle m, n\rangle \mid m \neq n\}$. ◁

Exercise 11. Try to read the formulas that were earlier given as example formulas over Σ_F and Σ_N in light of the interpretations J_F and J_N. Which ones do you think should hold?

Exercise 12. Write down an interpretation over signature Σ_M that does not match the intended usage of that signature (reasoning about monoids).

Exercise 13. Write down an interpretation over signature Σ_C from Exercise 1 with domain $D_C = \mathbb{N}$.

The semantics of variables is defined via an *assignment*, which tells us which element of the domain should correspond to each variable.

Definition. An *assignment* relative to an interpretation I is a function $\rho : X \to D$.

In order to deal with quantifiers, we need to be able to consider alternative assignments – in particular, those assignments that differ exactly on the value assigned to a particular variable.

Definition. Given a set $Y \subseteq X$, two assignments ρ and σ are said to be *equivalent up to* Y (or Y-equivalent) if $\rho(x) = \sigma(x)$ for every $x \notin Y$. This is abbreviated as $\rho \equiv_Y \sigma$.

In particular, $\rho \equiv_{\{x\}} \sigma$, or simply $\rho \equiv_x \sigma$, if ρ and σ agree on the values of every variable except for x. We write $\sigma = \rho[x/t]$ to denote the substitution σ such that $\sigma(x) = t$ and $\sigma(y) = \rho(y)$ for any $y \neq x$; by construction, it is always the case that $\rho[x/t] \equiv_x \rho$.

Exercise 14. Show that, for any set Y, the relation $\equiv_Y$ is an equivalence relation, i.e.:

- for every ρ, $\rho \equiv_Y \rho$;
- for every ρ and σ, if $\rho \equiv_Y \sigma$, then $\sigma \equiv_Y \rho$;
- for every ρ, σ and θ, if $\rho \equiv_Y \sigma$ and $\sigma \equiv_Y \theta$, then $\rho \equiv_Y \theta$.

Having both an interpretation and an assignment, we can map any term in the syntax to an element of the interpretation's domain.

Definition. Given an interpretation I over Σ and an assignment ρ relative to I, we define the denotation of terms inductively as follows.

$$\begin{aligned} [\![c]\!]^I_\rho &= c^I && \text{for } c \in F_0 \\ [\![x]\!]^I_\rho &= \rho(x) && \text{for } x \in \mathcal{X} \\ [\![f\,(t_1, \ldots, t_n)]\!]^I_\rho &= f^I\left([\![t_1]\!]^I_\rho, \ldots, [\![t_n]\!]^I_\rho\right) && \text{for } f \in F_n \end{aligned}$$

Before we discuss satisfaction of formulas, we illustrate how interpretations and assignments are used to evaluate terms.

Example F.5. Consider the signature Σ_F for our example on family relations and the intuitive interpretation I_F, together with an assignment ρ that we leave unspecified. We show that the intuitive meaning of the terms we discussed above corresponds to their formal semantics under this interpretation.

- $[\![\mathsf{Alice}]\!]^{I_F}_\rho = \mathsf{Alice}^{I_F}$;
- $[\![\mathsf{father}(\mathsf{Alice})]\!]^{I_F}_\rho = \mathsf{father}^{I_F}\left([\![\mathsf{Alice}]\!]^{I_F}_\rho\right) = \mathsf{father}^{I_F}\left(\mathsf{Alice}^{I_F}\right)$, which according to the definition of father^{I_F} is precisely Alice^{I_F}'s father;
- similarly, $[\![\mathsf{mother}(\mathsf{father}(\mathsf{Alice}))]\!]^{I_F}_\rho = \mathsf{mother}^{I_F}\left([\![\mathsf{father}(\mathsf{Alice})]\!]^{I_F}_\rho\right)$, and using the previous result this is $\mathsf{mother}^{I_F}\left(\mathsf{father}^{I_F}\left(\mathsf{Alice}^{I_F}\right)\right)$, which indeed corresponds to Alice^{I_F}'s paternal grandmother;
- $[\![\mathsf{mother}(\mathsf{mother}(x))]\!]^{I_F}_\rho = \mathsf{mother}^{I_F}\left([\![\mathsf{mother}(x)]\!]^{I_F}_\rho\right) = \mathsf{mother}^{I_F}\left(\mathsf{mother}^{I_F}\left([\![x]\!]^{I_F}_\rho\right)\right) = \mathsf{mother}^{I_F}\left(\mathsf{mother}^{I_F}\left(\rho(x)\right)\right)$; as expected, this term corresponds to $\rho(x)$'s maternal grandmother. ◁

Example M.5. Under the interpretation I_M for the signature Σ_M and the assignment ρ such that $\rho(x) = \mathtt{abc}$ and $\rho(y) = \mathtt{bb}$, the terms that we previously showed as example are interpreted as follows.

- $[\![\star]\!]^{I_M}_\rho = \star^{I_M} = \epsilon$, corresponding to the empty string;
- $[\![\cdot(x, y)]\!]^{I_M}_\rho = \cdot^{I_M}\left([\![x]\!]^{I_M}_\rho, [\![y]\!]^{I_M}_\rho\right) = \cdot^{I_M}(\rho(x), \rho(y)) = \rho(x)\rho(y) = \mathtt{abcbb}$ is the concatenation of $\mathtt{abc}$ and $\mathtt{bb}$;
- Finally, $[\![\cdot(x, \cdot(\star, x))]\!]^{I_M}_\rho = \cdot^{I_M}\left([\![x]\!]^{I_M}_\rho, [\![\cdot(\star, x)]\!]^{I_M}_\rho\right) = \cdot^{I_M}\left([\![x]\!]^{I_M}_\rho, \cdot^{I_M}\left([\![\star]\!]^{I_M}_\rho, [\![x]\!]^{I_M}_\rho\right)\right) = \rho(x)(\star^{I_M}\rho(x)) = \mathtt{abcabc}$. ◁

Example M.6. If, instead, we consider interpretation J_M, together with ρ such that $\rho(x) = 2$ and $\rho(y) = 6$, we obtain:

- $[\![\star]\!]^{J_M}_\rho = \star^{J_M} = 0$;
- $[\![\cdot(x, y)]\!]^{J_M}_\rho = \cdot^{J_M}\left([\![x]\!]^{J_M}_\rho, [\![y]\!]^{J_M}_\rho\right) = (\lambda jk.(j + k) \bmod 7)(\rho(x), \rho(y)) = (2 + 6) \bmod 7 = 1$;

- Finally, $[\![\cdot(x,\cdot(\star,x))]\!]_\rho^{J_M} = \cdot^{J_M}\left([\![x]\!]_\rho^{J_M}, [\![\cdot(\star,x)]\!]_\rho^{J_M}\right)$. We have that

$$[\![\cdot(\star,x)]\!]_\rho^{J_M} = \cdot^{J_M}\left([\![\star]\!]_\rho^{J_M}, [\![x]\!]_\rho^{J_M})\right) \\ = (\lambda jk.(j+k) \bmod 7)\left(\star^{J_M}, \rho(x)\right) = (0+2) \bmod 7 = 2\,,$$

and therefore

$$[\![\cdot(x,\cdot(\star,x))]\!]_\rho^{J_M} = \cdot^{J_M}\left([\![x]\!]_\rho^{J_M}, 2\right) = (\lambda jk.(j+k) \bmod 7)(\rho(x),2) = (2+2) \bmod 7 = 4\,.$$

◁

Example N.5. We now look at the terms given earlier as examples in the context of signature Σ_N, and their interpretation under interpretation I_N and an assignment ρ such that $\rho(x) = 2$.

- $[\![\mathsf{succ}(0)]\!]_\rho^{I_N} = \mathsf{succ}^{I_N}\left([\![\mathsf{0}]\!]_\rho^{I_N}\right) = \mathsf{succ}^{I_N}\left(\mathsf{0}^{I_N}\right) = (\lambda x.x+1)(0) = 0+1 = 1$, as expected.
- $[\![+(\mathsf{3},\mathsf{5})]\!]_\rho^{I_N} = +^{I_N}\left([\![\mathsf{3}]\!]_\rho^{I_N}, [\![\mathsf{5}]\!]_\rho^{I_N}\right) = +^{I_N}\left(\mathsf{3}^{I_N}, \mathsf{5}^{I_N}\right) = (\lambda xy.x+y)(3,5) = 3+5 = 8$, also as expected.
- $[\![\mathsf{8}]\!]_\rho^{I_N} = \mathsf{8}^{I_N} = 8$, which coincides with the previous result: indeed, we expect that under this interpretation both $\mathsf{3}+\mathsf{5}$ and $\mathsf{8}$ denote the same value.
- $[\![\times(+(x,\mathsf{4}),\mathsf{3})]\!]_\rho^{I_N} = \times^{I_N}\left([\![+(x,\mathsf{4})]\!]_\rho^{I_N}, [\![\mathsf{3}]\!]_\rho^{I_N}\right)$. For simplicity of presentation, we compute the arguments of $\times^{I_N}$ separately.

  For the first argument, we have $[\![+(x,\mathsf{4})]\!]_\rho^{I_N} = +^{I_N}\left([\![x]\!]_\rho^{I_N}, [\![\mathsf{4}]\!]_\rho^{I_N}\right) = +^{I_N}\left(\rho(x), \mathsf{4}^{I_N}\right) = (\lambda xy.x+y)(2,4) = 2+4 = 6$.

  For the second argument, $[\![\mathsf{3}]\!]_\rho^{I_N} = \mathsf{3}^{I_N} = 3$.

  Therefore $[\![\times(+(x,\mathsf{4}),\mathsf{3})]\!]_\rho^{I_N} = \times^{I_N}(6,3) = (\lambda xy.x\times y)(6,3) = 6\times 3 = 18$.
- $[\![\times(\mathsf{3},+(x,\mathsf{4}))]\!]_\rho^{I_N} = \times^{I_N}\left([\![\mathsf{3}]\!]_\rho^{I_N}, [\![+(x,\mathsf{4})]\!]_\rho^{I_N}\right) = (\lambda xy.x\times y)(3,6) = 3\times 6 = 18$ using the auxiliary results computed above. ◁

The semantics of atomic formulas is derived by treating predicate symbols in a similar way as function symbols. Applying the usual inductive construction, we obtain semantics for all first-order formulas. The rule for dealing with the universal quantifier uses the notion of x-equivalent assignment.

Definition. Satisfaction of formulas is defined inductively as follows.

$$I \models_\rho p(t_1,\ldots,t_n) \text{ iff } \langle [\![t_1]\!]_\rho^I, \ldots, [\![t_n]\!]_\rho^I\rangle \in p^I \qquad I \models_\rho \neg\varphi \text{ iff } I \not\models_\rho \varphi$$
$$I \models_\rho \forall x\varphi \text{ iff } I \models_\sigma \varphi \text{ whenever } \sigma \equiv_x \rho \qquad I \models_\rho \varphi\to\psi \text{ iff } I \not\models_\rho \varphi \text{ or } I \models_\rho \psi$$

We say that φ is true in I, written $I \models \varphi$, if $I \models_\rho \varphi$ for every assignment ρ relative to I.

From these definitions, we can derive the semantics of the remaining connectives that are defined as abbreviations. For the propositional connectives $\vee$, $\wedge$ and $\leftrightarrow$, we obtain the same behaviour as in propositional logic. For the existential quantifier, we have the following result.

Lemma 26. Let I be an interpretation, ρ be an assignment over I, x be a variable and φ be a formula. Then $I \models_\rho \exists x\varphi$ iff $I \models_\sigma \varphi$ for some $\sigma \equiv_x \rho$.

Proof. Recall that $\exists x\varphi$ is an abbreviation of $\neg\forall x\neg\varphi$. Therefore, $I \models_\rho \exists x\varphi$ iff $I \models_\rho \neg\forall x\neg\varphi$ iff $I \not\models_\rho \forall x\neg\varphi$.

Now, we have that $I \models_\rho \forall x\neg\varphi$ iff $I \models_\sigma \neg\varphi$ for any $\sigma \equiv_x \rho$, i.e., iff $I \not\models_\sigma \varphi$ for every such σ. So $I \models_\rho \exists x\varphi$ iff it is not the case that ($I \not\models_\sigma \varphi$ for every $\sigma \equiv_x \rho$). But stating that not every σ satisfies $I \not\models_\sigma \varphi$ is the same as stating that $I \models_\sigma \varphi$ for at least one such σ. □

Validity, satisfiability, and related concepts are defined as usual. Entailment is also defined in a similar way, but it is important to note that the role of model is played by I (not by I and ρ together).

Definition. Let Γ be a set of formulas and φ be a formula. We say that an interpretation I *satisfies* Γ, or that I is a *model* of Γ, if $I \models \gamma$ for every $\gamma \in \Gamma$. We write $I \models \Gamma$ when this is the case.

We say that Γ *entails* φ, written $\Gamma \models \varphi$, if $I \models \varphi$ for every interpretation I such that $I \models \Gamma$.

We now illustrate how the semantics of formulas works in the context of the two examples we introduced earlier.

Example F.6. Consider again the signature Σ_F for our example on family relations and the intuitive interpretation I_F, together with an assignment ρ that we leave unspecified. We now look at the formulas presented in an earlier example.

- $I_F \models_\rho \mathsf{female}(\mathsf{Alice})$ iff $[\![\mathsf{Alice}]\!]^{I_F}_\rho \in \mathsf{female}^{I_F}$. Since $[\![\mathsf{Alice}]\!]^{I_F}_\rho = \mathsf{Alice}^{I_F}$, this formula holds iff Alice^{I_F} is a woman.

- Similarly, $I_F \models_\rho \mathsf{male}(\mathsf{father}(\mathsf{Alice}))$ iff $[\![\mathsf{father}(\mathsf{Alice})]\!]^{I_F}_\rho \in \mathsf{male}^{I_F}$. We already know from earlier that $[\![\mathsf{father}(\mathsf{Alice})]\!]^{I_F}_\rho$ is Alice^{I_F}'s father, and the formula is therefore true in I_F under ρ since Alice^{I_F}'s father is a man.

- Likewise, $I_F \models_\rho \mathsf{male}(\mathsf{father}(x))$ iff $[\![\mathsf{father}(x)]\!]^{I_F}_\rho \in \mathsf{male}^{I_F}$. We know that $[\![\mathsf{father}(x)]\!]^{I_F}_\rho = \mathsf{father}^{I_F}\left([\![x]\!]^{I_F}_\rho\right) = \mathsf{father}^{I_F}(\rho(x))$. Even though we do not know who $\rho(x)$ refers to, we can still be sure that their father is a man, so this formula is true in I_F under ρ.

- The rule for the universal quantifier states that $I_F \models_\rho \forall x.\mathsf{male}(\mathsf{father}(x))$ iff $I_F \models_\sigma \mathsf{male}(\mathsf{father}(x))$ for every $\sigma \equiv_x \rho$. Let us consider such an assignment σ. Reasoning as in the previous case, $I_F \models_\sigma \mathsf{male}(\mathsf{father}(x))$ iff $\mathsf{father}^{I_F}(\sigma(x)) \in \mathsf{male}^{I_F}$, which is certainly the case. Since this holds for every σ, we conclude that $I_F \models_\rho \forall x.\mathsf{male}(\mathsf{father}(x))$.

- Finally, $I_F \models_\rho \forall x.(\mathsf{married}(\mathsf{Bob}, x) \to \mathsf{married}(x, \mathsf{Bob}))$ iff for any $\sigma \equiv_x \rho$ it is the case that $I_F \models_\sigma \mathsf{married}(\mathsf{Bob}, x) \to \mathsf{married}(x, \mathsf{Bob})$. Again, let us consider an arbitrary such assignment σ. Then:

$$
\begin{aligned}
& I_F \models_\sigma \mathsf{married}(\mathsf{Bob}, x) \to \mathsf{married}(x, \mathsf{Bob}) \\
\text{iff } & I_F \not\models_\sigma \mathsf{married}(\mathsf{Bob}, x) \text{ or } I_F \models_\sigma \mathsf{married}(x, \mathsf{Bob}) \\
\text{iff } & \left\langle [\![\mathsf{Bob}]\!]^{I_F}_\sigma, [\![x]\!]^{I_F}_\sigma \right\rangle \notin \mathsf{married}^{I_F} \text{ or } \left\langle [\![x]\!]^{I_F}_\sigma, [\![\mathsf{Bob}]\!]^{I_F}_\sigma \right\rangle \in \mathsf{married}^{I_F} \\
\text{iff } & \left\langle \mathsf{Bob}^{I_F}, \sigma(x) \right\rangle \notin \mathsf{married}^{I_F} \text{ or } \left\langle \sigma(x), \mathsf{Bob}^{I_F} \right\rangle \in \mathsf{married}^{I_F}
\end{aligned}
$$

Now, regardless of who $\sigma(x)$ and Bob^{I_F} actually are, one of these conditions necessarily holds – otherwise Bob^{I_F} would be married to $\sigma(x)$, but not conversely.[1] Since σ is arbitrary, we conclude that $I_F \models_\rho \forall x.(\mathsf{married}(\mathsf{Bob}, x) \to \mathsf{married}(x, \mathsf{Bob}))$. ◁

Before presenting a similar analysis for our mathematical example, we state and prove a result to deal with repeated quantifiers of the same type.

Lemma 27. Let I be an interpretation, ρ be an assignment over I, $x_1, \ldots, x_n$ be variables and φ be a formula. Then $I \models_\rho \forall x_1 \ldots \forall x_n \varphi$ iff $I \models_\sigma \varphi$ for every $\sigma \equiv_{\{x_1,\ldots,x_n\}} \rho$.

Proof. Intuitively, by repeatedly applying the definition of the semantics of the universal quantifier, we quantify over all assignments that differ from ρ in the variables $x_1, \ldots, x_n$ sequentially; this is equivalent to allowing new values on all those variables at the same time.

The formal proof is by induction on n. The base case $n = 1$ is trivial, since it coincides with the way the semantics of $\forall x_1 \varphi$ is defined.

For the inductive step, consider the formula $\forall y \forall x_1 \ldots \forall x_n \varphi$. From the semantics, $I \models_\rho \forall y \forall x_1 \ldots \forall x_n \varphi$ iff $I \models_\theta \forall x_1 \ldots \forall x_n \varphi$ for every $\theta \equiv_y \rho$. By induction hypothesis, for each such θ, this is the case iff $I \models_\sigma \varphi$ for any $\sigma \equiv_{\{x_1,\ldots,x_n\}} \theta$. Therefore $I \models_\rho \forall y \forall x_1 \ldots \forall x_n \varphi$ iff: for every $\theta \equiv_y \rho$ and every $\sigma \equiv_{\{x_1,\ldots,x_n\}} \theta$, it is the case that $I \models_\sigma \varphi$. We now show that the last condition quantifies exactly over the substitutions that are $\{y, x_1, \ldots, x_n\}$-equivalent to ρ.

Assume that $\theta \equiv_y \rho$ and that $\sigma \equiv_{\{x_1,\ldots,x_n\}} \theta$, and let $z \in \mathcal{X} \setminus \{y, x_1, \ldots, x_n\}$. Then $\rho(z) = \theta(z) = \sigma(z)$, where the first equality holds because $z \neq y$ and the second equality holds because $z \notin \{x_1, \ldots, x_n\}$. This shows that $\sigma \equiv_{\{y,x_1,\ldots,x_n\}} \rho$.

Conversely, if $\sigma \equiv_{\{y,x_1,\ldots,x_n\}} \rho$, let $\theta = \rho[y/\sigma(y)]$. Then $\theta \equiv_y \rho$ by construction, and $\sigma \equiv_{\{x_1,\ldots,x_n\}} \theta$, since ρ and σ agree on all variables different from y not in $\{x_1, \ldots, x_n\}$. Therefore σ is $\{x_1, \ldots, x_n\}$-equivalent to some substitution that is y-equivalent to ρ. □

Exercise 15. Prove that: for all sets $X, Y \subseteq \mathcal{X}$ and substitutions ρ, σ and θ, if $\rho \equiv_X \sigma$ and $\sigma \equiv_Y \theta$, then $\rho \equiv_{X \cup Y} \theta$. Do you think it is necessary that $X \cap Y = \emptyset$? Why?

We use this result in some of the examples below. Observe that a similar result can be proved for repeated existential quantifiers. However, when dealing with alternating quantifiers (e.g., $\forall x \exists y \varphi$) we need to treat them one type at a time to get the correct dependencies between assignments, as in the last formulas in the next example.

Example M.7. Continuing with our monoid example, we look at the formulas we wrote down before and analyse their semantics under interpretation I_M and the assignment ρ given earlier.

- $I_M \models_\rho x \;\mathsf{eq}\; x$ iff $\langle [\![x]\!]^{I_M}_\rho, [\![x]\!]^{I_M}_\rho \rangle \in \mathsf{eq}^{I_M}$, which holds since eq^{I_M} contains all pairs of equal elements.

- To understand whether $I_M \models_\rho \forall xy.(x \cdot y \;\mathsf{eq}\; y \cdot x)$ we recall that, by Lemma 27, this is equivalent to $I_M \models_\sigma x \cdot y \;\mathsf{eq}\; y \cdot x$ for all $\sigma \equiv_{\{x,y\}} \rho$.

[1] The authors of these notes are actually aware of at least such a case, but we will disregard such legal anomalies in this example.

Consider such an assignment σ. Then $I_M \models_\sigma x \cdot y \;\mathsf{eq}\; y \cdot x$ iff $\left\langle [\![x \cdot y]\!]^{I_M}_\sigma, [\![y \cdot x]\!]^{I_M}_\sigma \right\rangle \in \mathsf{eq}^{I_M}$, i.e., iff $[\![x \cdot y]\!]^{I_M}_\sigma = [\![y \cdot x]\!]^{I_M}_\sigma$. Now we have that:

$$[\![x \cdot y]\!]^{I_M}_\sigma = \cdot^{I_M} \left([\![x]\!]^{I_M}_\sigma, [\![y]\!]^{I_M}_\sigma \right) = \cdot^{I_M} (\sigma(x), \sigma(y)) = \sigma(x)\sigma(y)$$
$$[\![y \cdot x]\!]^{I_M}_\sigma = \cdot^{I_M} \left([\![y]\!]^{I_M}_\sigma, [\![x]\!]^{I_M}_\sigma \right) = \cdot^{I_M} (\sigma(y), \sigma(x)) = \sigma(y)\sigma(x)$$

which does not necessarily hold: taking $\sigma = \rho$ we have that $\sigma(x)\sigma(y) = \texttt{abcbb}$, whereas $\sigma(y)\sigma(x) = \texttt{bbabc}$. Therefore we conclude that $I_M \not\models_\rho \forall xy.(x \cdot y \;\mathsf{eq}\; y \cdot x)$.

- To decide whether $I_M \models_\rho \forall xyz.(x \cdot (y \cdot z) \;\mathsf{eq}\; (x \cdot y) \cdot z)$, we make a similar analysis. We start by observing that this is the case iff $I_M \models_\sigma x \cdot (y \cdot z) \;\mathsf{eq}\; (x \cdot y) \cdot z$ for every assignment $\sigma \equiv_{\{x,y,z\}} \rho$, which again holds iff $\left\langle [\![x \cdot (y \cdot z)]\!]^{I_M}_\sigma, [\![(x \cdot y) \cdot z]\!]^{I_M}_\sigma \right\rangle \in \mathsf{eq}^{I_M}$ iff $[\![x \cdot (y \cdot z)]\!]^{I_M}_\sigma = [\![(x \cdot y) \cdot z]\!]^{I_M}_\sigma$, and that we have

$$\begin{aligned}
[\![x \cdot (y \cdot z)]\!]^{I_M}_\sigma &= \cdot^{I_M} \left([\![x]\!]^{I_M}_\sigma, [\![y \cdot z]\!]^{I_M}_\sigma \right) = \cdot^{I_M} \left([\![x]\!]^{I_M}_\sigma, \cdot^{I_M} \left([\![y]\!]^{I_M}_\sigma, [\![z]\!]^{I_M}_\sigma \right) \right) \\
&= \cdot^{I_M} \left(\sigma(x), \cdot^{I_M} (\sigma(y), \sigma(z)) \right) = \sigma(x)(\sigma(y)\sigma(z)) \\
[\![(x \cdot y) \cdot z]\!]^{I_M}_\sigma &= \cdot^{I_M} \left([\![x \cdot y]\!]^{I_M}_\sigma, [\![z]\!]^{I_M}_\sigma \right) = \cdot^{I_M} \left(\cdot^{I_M} \left([\![x]\!]^{I_M}_\sigma, [\![y]\!]^{I_M}_\sigma \right), [\![z]\!]^{I_M}_\sigma \right) \\
&= \cdot^{I_M} \left(\cdot^{I_M} (\sigma(x), \sigma(y)), \sigma(x) \right) = (\sigma(x)\sigma(y))\sigma(z)
\end{aligned}$$

Since the last two expressions are identical (they both contain the symbols in $\sigma(x)$, $\sigma(y)$ and $\sigma(z)$ in the same order), we conclude that the original formula holds.

- Similarly, $I_M \models_\rho \forall x.(x \cdot \star \;\mathsf{eq}\; x)$ iff $I_M \models_\sigma x \cdot \star \;\mathsf{eq}\; x$ for any $\sigma \equiv_x \rho$, which holds iff $\left\langle [\![x \cdot \star]\!]^{I_M}_\sigma, [\![x]\!]^{I_M}_\sigma = \sigma(x) \right\rangle \in \mathsf{eq}^{I_M}$ iff $[\![x \cdot \star]\!]^{I_M}_\sigma = [\![x]\!]^{I_M}_\sigma = \sigma(x)$. Since we have that $[\![x \cdot \star]\!]^{I_M}_\sigma = \cdot^{I_M} \left([\![x]\!]^{I_M}_\sigma, [\![\star]\!]^{I_M}_\sigma \right) = \sigma(x)\epsilon = \sigma(x)$, we conclude that this formula is also true in I_M under ρ.

- The next formula has alternating quantifiers. First, observe that $I_M \models_\rho \forall x \exists y.(x \cdot y \;\mathsf{eq}\; \star)$ iff $I_M \models_\sigma \exists y.(x \cdot y \;\mathsf{eq}\; \star)$ for any $\sigma \equiv_x \rho$. Given such σ, this is in turn equivalent to requiring that $I_M \models_\theta x \cdot y \;\mathsf{eq}\; \star$ for some $\theta \equiv_y \sigma$.

  By definition, $I_M \models_\theta x \cdot y \;\mathsf{eq}\; \star$ iff $\left\langle [\![x \cdot y]\!]^{I_M}_\theta, [\![\star]\!]^{I_M}_\theta \right\rangle \in \mathsf{eq}^{I_M}$ iff $[\![x \cdot y]\!]^{I_M}_\theta = [\![\star]\!]^{I_M}_\theta$. Now

$$[\![x \cdot y]\!]^{I_M}_\theta = \cdot^{I_M} \left([\![x]\!]^{I_M}_\theta, [\![y]\!]^{I_M}_\theta \right) = \cdot^{I_M} (\theta(x), \theta(y)) = \theta(x)\theta(y) = \sigma(x)\theta(y)$$

  since $\theta \equiv_y \sigma$; on the other hand, $[\![\star]\!]^{I_M}_\theta = \star^{I_M} = \epsilon$.

  So the question is: given the value of $\sigma(x)$, can we always find a value for $\theta(y)$ such that $\sigma(x)\theta(y) = \epsilon$? This is not the case, since adding symbols to $\sigma(x)$ can never result in the empty sequence unless $\sigma(x) = \epsilon$. Therefore $I_M \not\models_\rho \forall x \exists y.(x \cdot y \;\mathsf{eq}\; \star)$.

- We now discuss whether $I_M \models_\rho \exists y \forall x.(x \cdot y \;\mathsf{eq}\; \star)$. This happens if there is an assignment $\sigma \equiv_y \rho$ such that $I_M \models_\sigma \forall x.(x \cdot y \;\mathsf{eq}\; \star)$. In turn, this is the case if $I_M \models_\theta x \cdot y \;\mathsf{eq}\; \star$ for any $\theta \equiv_x \sigma$.

  From the previous example, we already know that $I_M \models_\theta x \cdot y \;\mathsf{eq}\; \star$ iff $\theta(x)\theta(y) = \epsilon$, and since $\theta(y) = \sigma(y)$ this simplifies to $\theta(x)\sigma(y) = \epsilon$. So the question is: can we choose $\sigma(y)$ such that $\theta(x)\sigma(y) = \epsilon$ regardless of the value of $\theta(x)$? As in the previous case, this is not true, so $I_M \not\models_\rho \exists y \forall x.(x \cdot y \;\mathsf{eq}\; \star)$. ◁

Example M.8. We now analyze the same formulas in interpretation J_M with the assignment ρ given in Example 6.

- $J_M \models_\rho x \;\mathsf{eq}\; x$ holds for the same reason as above – namely, that eq^{J_M} contains all pairs of equal elements.

- As before, $J_M \models_\rho \forall xy.(x \cdot y \;\mathsf{eq}\; y \cdot x)$ iff $J_M \models_\sigma x \cdot y \;\mathsf{eq}\; y \cdot x$ for all $\sigma \equiv_{\{x,y\}} \rho$. As before, given such an assignment σ, this holds iff $\cdot^{J_M}(\sigma(x), \sigma(y)) = \cdot^{J_M}(\sigma(x), \sigma(y))$ always holds, which in this interpretation reduces to $(\sigma(x) + \sigma(y)) \bmod 7 = (\sigma(y) + \sigma(x)) \bmod 7$. The latter is necessarily true, since addition is commutative.

  Therefore we conclude that $J_M \models_\rho \forall xy.x \cdot y \;\mathsf{eq}\; y \cdot x$.

- The analysis of whether $J_M \models_\rho \forall xyz.(x \cdot (y \cdot z) \;\mathsf{eq}\; (x \cdot y) \cdot z)$ is also similar to that in I_M. This condition holds iff $J_M \models_\sigma x \cdot (y \cdot z) \;\mathsf{eq}\; (x \cdot y) \cdot z$ for every assignment $\sigma \equiv_{\{x,y,z\}} \rho$, which again holds iff $[\![x \cdot (y \cdot z)]\!]_\sigma^{J_M} = [\![(x \cdot y) \cdot z]\!]_\sigma^{J_M}$, and using the results from before we have that

$$[\![x \cdot (y \cdot z)]\!]_\sigma^{J_M} = \cdot^{J_M}\left(\sigma(x), \cdot^{J_M}(\sigma(y), \sigma(z))\right) = (\sigma(x) + ((\sigma(y) + \sigma(z)) \bmod 7)) \bmod 7$$
$$[\![(x \cdot y) \cdot z]\!]_\sigma^{J_M} = \cdot^{J_M}\left(\cdot^{J_M}(\sigma(x), \sigma(y)), \sigma(x)\right) = (((\sigma(x) + \sigma(y)) \bmod 7) + \sigma(z)) \bmod 7\,.$$

  These two expressions coincide, since addition modulo 7 is an associative operation.

- For $J_M \models_\rho \forall x.(x \cdot \star \;\mathsf{eq}\; x)$, we know that it holds iff $[\![x \cdot \star]\!]_\sigma^{J_M} = [\![x]\!]_\sigma^{J_M} = \sigma(x)$ for any $\sigma \equiv_x \rho$. As before, we can easily see that

$$[\![x \cdot \star]\!]_\sigma^{J_M} = \cdot^{I_M}\left([\![x]\!]_\sigma^{I_M}, [\![\star]\!]_\sigma^{I_M}\right) = (\sigma(x) + 0) \bmod 7 = \sigma(x)\,,$$

  and therefore the original formula is also true in J_M under ρ.

- Again reusing previous results, we see that $J_M \models_\rho \forall x \exists y.(x \cdot y \;\mathsf{eq}\; \star)$ iff for any $\sigma \equiv_x \rho$ we can find $\theta \equiv_y \sigma$ such that $J_M \models_\theta x \cdot y \;\mathsf{eq}\; \star$. In turn, this last condition is equivalent to $[\![x \cdot y]\!]_\theta^{J_M} = [\![\star]\!]_\theta^{J_M}$, and

$$[\![x \cdot y]\!]_\theta^{J_M} = \cdot^{J_M}(\theta(x), \theta(y)) = (\theta(x) + \theta(y)) \bmod 7 = (\sigma(x) + \theta(y)) \bmod 7\,,$$

  while $[\![\star]\!]_\theta^{J_M} = \star^{J_M} = 0$.

  So the question is: given the value of $\sigma(x)$, can we always find a value for $\theta(y)$ such that $(\sigma(x) + \theta(y)) \bmod 7 = 0$?

  The answer is now affirmative: if we take $\theta(y) = (7 - \sigma(x)) \bmod 7$ (i.e., $\theta(y) = 0$ if $\sigma(x) = 0$ and $\theta(y) = 7 - \sigma(x)$ otherwise) we obtain that $\theta(y) + \sigma(x)$ is either 0 (if $\sigma(x) = 0$) or 7, and in both cases the modulo operation returns 0. Therefore $J_M \models_\rho \forall x \exists y.(x \cdot y \;\mathsf{eq}\; \star)$.

- However, it is still not the case that $J_M \models_\rho \exists y \forall x.(x \cdot y \;\mathsf{eq}\; \star)$. Indeed, this formula requires that we can choose $\sigma(y)$ such that $(\theta(x) + \sigma(y)) \bmod 7 = 0$ regardless of the value of $\theta(x)$, and this is not possible: if $\theta(x) = 0$ then the expression $\theta(x) + \sigma(y)$ evaluates to $\sigma(y)$, so $\sigma(y)$ would need to be 0, but if $\theta(x) = 1$ then $\sigma(y)$ would need to be 6. ◁

The last two formulas in the last example illustrate that the order of quantifiers is very relevant in the semantics of first-order logic, and it is necessary to be careful when manipulating assignments to ensure that the right dependencies are kept. As with other logics, we will later see that deductive systems can take care of these concerns automatically, simplifying our reasoning task.

Example N.6. We now look at the formulas introduced earlier in the context of signature Σ_N, and whether they are satisfied under interpretation I_N and an assignment ρ such that $\rho(x) = 2$.

- $I_N \models_\rho \mathsf{even}(2)$ iff $[\![2]\!]^{I_N}_\rho \in \mathsf{even}^{I_N}$, and since $[\![2]\!]^{I_N}_\rho = 2^{I_N} = 2$ is an even number, this formula is true in I_N under ρ.

- $I_N \models_\rho =(+(\mathsf{3},\mathsf{5}),\mathsf{8})$ iff $\left\langle [\![+(\mathsf{3},\mathsf{5})]\!]^{I_N}_\rho, [\![\mathsf{8}]\!]^{I_N}_\rho \right\rangle \in =^{I_N}$, i.e., iff $[\![+(\mathsf{3},\mathsf{5})]\!]^{I_N}_\rho = [\![\mathsf{8}]\!]^{I_N}_\rho$. Since we already saw that $[\![+(\mathsf{3},\mathsf{5})]\!]^{I_N}_\rho = 8 = [\![\mathsf{8}]\!]^{I_N}_\rho$, this is the case.

- $I_N \models_\rho \forall n.n \leq \mathsf{succ}(n)$ iff $I_N \models_\sigma n \leq \mathsf{succ}(n)$ for every σ that is n-equivalent to ρ. Let us consider such an assignment σ. Then $I_N \models_\sigma n \leq \mathsf{succ}(n)$ iff $\left\langle [\![n]\!]^{I_N}_\sigma, [\![\mathsf{succ}(n)]\!]^{I_N}_\sigma \right\rangle \in \leq^{I_N}$ iff $[\![n]\!]^{I_N}_\sigma \leq [\![\mathsf{succ}(n)]\!]^{I_N}_\sigma$. Since

$$\begin{aligned} [\![n]\!]^{I_N}_\sigma &= \sigma(n) \\ \text{and } [\![\mathsf{succ}(n)]\!]^{I_N}_\sigma &= \mathsf{succ}^{I_N}\left([\![n]\!]^{I_N}_\sigma\right) = (\lambda x.x+1)(\sigma(n)) = \sigma(n)+1\,, \end{aligned}$$

  this is also the case. Therefore $I_N \models_\rho \forall n.n \leq \mathsf{succ}(n)$. Note that the actual value of $\rho(n)$ is immaterial for this example.

- $I_N \models_\rho \forall y.\mathsf{3}+y = y+\mathsf{3}$ iff $I_N \models_\sigma \mathsf{3}+y = y+\mathsf{3}$ for any $\sigma \equiv_y \rho$.

  Let σ be such an assignment. Then $I_N \models_\sigma \mathsf{3}+y = y+\mathsf{3}$ iff $\left\langle [\![\mathsf{3}+y]\!]^{I_N}_\sigma, [\![y+\mathsf{3}]\!]^{I_N}_\sigma \right\rangle \in =^{I_N}$ iff $[\![\mathsf{3}+y]\!]^{I_N}_\sigma = [\![y+\mathsf{3}]\!]^{I_N}_\sigma$. Now

$$\begin{aligned} [\![\mathsf{3}+y]\!]^{I_N}_\sigma &= +^{I_N}\left([\![\mathsf{3}]\!]^{I_N}_\sigma, [\![y]\!]^{I_N}_\sigma\right) = +^{I_N}\left(\mathsf{3}^{I_N}, \sigma(y)\right) = (\lambda xy.x+y)(3,\sigma(y)) = 3+\sigma(y) \\ [\![y+\mathsf{3}]\!]^{I_N}_\sigma &= +^{I_N}\left([\![y]\!]^{I_N}_\sigma, [\![\mathsf{3}]\!]^{I_N}_\sigma\right) = +^{I_N}\left(\sigma(y), \mathsf{3}^{I_N}\right) = (\lambda xy.x+y)(\sigma(y),3) = \sigma(y)+3 \end{aligned}$$

  and since $3+\sigma(y) = \sigma(y)+3$, the original formula is true in I_N under ρ.

- $I_N \models_\rho \forall xyz.x+(y\times z) = (x+y)\times z$ iff $I_N \models_\sigma x+(y\times z) = (x+y)\times z$ for every $\sigma \equiv_{\{x,y,z\}} \rho$. As before, this is the case iff $[\![x+(y\times z)]\!]^{I_N}_\sigma = [\![(x+y)\times z]\!]^{I_N}_\sigma$.

  We have that

$$\begin{aligned} [\![x+(y\times z)]\!]^{I_N}_\sigma &= +^{I_N}\left([\![x]\!]^{I_N}_\sigma, [\![y\times z]\!]^{I_N}_\sigma\right) \\ &= +^{I_N}\left([\![x]\!]^{I_N}_\sigma, \times^{I_N}\left([\![y]\!]^{I_N}_\sigma, [\![z]\!]^{I_N}_\sigma\right)\right) \\ &= (\lambda xy.x+y)\left(\sigma(x), (\lambda xy.x\times y)\left(\sigma(y),\sigma(z)\right)\right) \\ &= (\lambda xy.x+y)\left(\sigma(x), \sigma(y)\times\sigma(z)\right) = \sigma(x)+\sigma(y)\times\sigma(z) \\ [\![(x+y)\times z]\!]^{I_N}_\sigma &= \times^{I_N}\left([\![x+y]\!]^{I_N}_\sigma, [\![z]\!]^{I_N}_\sigma\right) \\ &= \times^{I_N}\left(+^{I_N}\left([\![x]\!]^{I_N}_\sigma, [\![y]\!]^{I_N}_\sigma\right), [\![z]\!]^{I_N}_\sigma\right) \\ &= (\lambda xy.x\times y)\left((\lambda xy.x+y)\left(\sigma(x),\sigma(y)\right),\sigma(z)\right) \\ &= (\lambda xy.x\times y)\left(\sigma(x)+\sigma(y),\sigma(z)\right) = (\sigma(x)+\sigma(y))\times\sigma(z) \end{aligned}$$

  and the two resulting expressions are not always equal: if we choose σ such that $\sigma(x) = 1$, $\sigma(y) = 2$ and $\sigma(z) = 0$, then $\sigma(x)+\sigma(y)\times\sigma(z) = 1$ and $(\sigma(x)+\sigma(y))\times\sigma(z) = 0$. ◁

The previous examples also show that, although an assignment must give values to all variables, in practice we only care about its values on variables that occur in the formula we are evaluating. Even stronger: the only relevant values are those assigned to the variables that occur free in the formula. The next results formally state this observation.

Lemma 28. Let t be a term, I be an interpretation, and ρ and σ be assignments such that $\rho(x) = \sigma(x)$ for all $x \in V(t)$. Then $[\![t]\!]^I_\rho = [\![t]\!]^I_\sigma$.

Proof. By structural induction on t: if t is a constant c, then $[\![t]\!]^I_\rho = c^I = [\![t]\!]^I_\sigma$; if t is a variable x, then by hypothesis $\rho(x) = \sigma(x)$ and therefore $[\![t]\!]^I_\rho = \rho(x) = \sigma(x) = [\![t]\!]^I_\sigma$; and if t is a function symbol applied to terms, then using the induction hypothesis we obtain $[\![f(t_1, \ldots, t_n)]\!]^I_\rho = f^I([\![t_1]\!]^I_\rho, \ldots, [\![t_n]\!]^I_\rho) = f^I([\![t_1]\!]^I_\sigma, \ldots, [\![t_n]\!]^I_\sigma) = [\![f(t_1, \ldots, t_n)]\!]^I_\sigma$. □

Lemma 29. Let φ be a formula, I be an interpretation structure, and ρ and σ be assignments such that $\rho(x) = \sigma(x)$ for all $x \in FV(\varphi)$. Then $I \models_\rho \varphi$ iff $I \models_\sigma \varphi$.

Proof. By structural induction on φ.

For atomic formulas, the previous lemma guarantees that $[\![t_i]\!]^I_\rho = [\![t_i]\!]^I_\sigma$ for each $i = 1, \ldots, n$. Therefore $\langle [\![t_1]\!]^I_\rho, \ldots, [\![t_n]\!]^I_\rho \rangle \in p^I$ iff $\langle [\![t_1]\!]^I_\sigma, \ldots, [\![t_n]\!]^I_\sigma \rangle \in p^I$, and thus $I \models_\rho p(t_1, \ldots, t_n)$ iff $I \models_\sigma p(t_1, \ldots, t_n)$.

For negation, $I \models_\rho \neg\psi$ iff $I \not\models_\rho \psi$ iff $I \not\models_\sigma \psi$ (by induction hypothesis) iff $I \models_\sigma \neg\psi$.

For implication, $I \models_\rho \psi \to \gamma$ iff ($I \not\models_\rho \psi$ or $I \models_\rho \gamma$) iff ($I \not\models_\sigma \psi$ or $I \models_\sigma \gamma$) (by induction hypothesis) iff $I \models_\sigma \psi \to \gamma$.

Finally, we consider the case where φ is a universally quantified formula $\forall x\psi$. By definition, $I \models_\rho \forall x\psi$ iff $I \models_\theta \psi$ for every $\theta \equiv_x \rho$. For each such θ we can define θ' by $\theta'(y) = \theta(y)$ if $y \in FV(\psi)$ and $\theta'(y) = \sigma(y)$ otherwise. By construction θ' agrees with θ on every free variable in ψ, so by induction hypothesis $I \models_\theta \psi$ iff $I \models_{\theta'} \psi$. Furthermore:

- each $\theta' \equiv_x \sigma$, since for each $y \in FV(\psi) \setminus \{x\}$ we have that $\theta'(y) = \theta(y) = \rho(y) = \sigma(y)$, and by construction $\theta'(y) = \sigma(y)$ for $y \notin FV(\psi)$;

- every substitution x-equivalent to σ can be constructed in this way from some $\theta \equiv_x \rho$.

Therefore $I \models_\rho \forall x\psi$ iff $I \models_\theta \psi$ for every $\theta \equiv_x \rho$ iff $I \models_{\theta'} \psi$ for every $\theta \equiv_x \sigma$ iff $I \models_\sigma \forall x\psi$. □

In particular, if φ is closed, then $I \models \varphi$ iff $I \models_\rho \varphi$ for some ρ iff $I \models_\rho \varphi$ for any ρ.

Exercise 16. Using this observation, go through the last examples and decide for which formulas you can conclude that they are true in I_F, I_M, J_M or I_N, respectively.

We now present a result that we will use in several places: it gives a semantic description of the ternary relation $t \triangleright x : \varphi$ that we defined earlier, and allows us to relate syntactic substitutions with operations on assignments.

Lemma 30. Let x be a variable and t, t_x be terms. For every interpretation I and assignment ρ, $[\![t\,[x/t_x]]\!]^I_\rho = [\![t]\!]^I_{\rho[x/[\![t_x]\!]^I_\rho]}$.

Proof. By structural induction on t. For conciseness, let $\sigma = \rho\left[x/[\![t_x]\!]^I_\rho\right]$. We have:

$$
\begin{aligned}
&[\![c\,[x/t_x]]\!]^I_\rho = [\![c]\!]^I_\rho = c^I = [\![c]\!]^I_\sigma && \text{for constants}\\
&[\![x\,[x/t_x]]\!]^I_\rho = [\![t_x]\!]^I_\rho = \sigma(x) = [\![x]\!]^I_\sigma &&\\
&[\![y\,[x/t_x]]\!]^I_\rho = [\![y]\!]^I_\rho = \rho(y) = \sigma(y) = [\![y]\!]^I_\sigma && \text{for } y \in \mathcal{X} \setminus \{x\}
\end{aligned}
$$

If $f \in F_n$ is a function symbol and $t_1, \ldots, t_n$ are terms, by induction hypothesis we know that $[\![t_i\,[x/t_x]]\!]^I_\rho = [\![t_i]\!]^I_\sigma$ for each $i = 1, \ldots, n$. Then we also have

$$\begin{aligned}
[\![f\,(t_1, \ldots, t_n)\,[x/t_x]]\!]^I_\rho &= [\![f\,(t_1\,[x/t_x], \ldots, t_n\,[x/t_x])]\!]^I_\rho \\
&= f^I\left([\![t_1\,[x/t_x]]\!]^I_\rho, \ldots, [\![t_n\,[x/t_x]]\!]^I_\rho\right) \\
&= f^I\left([\![t_1]\!]^I_\sigma, \ldots, [\![t_n]\!]^I_\sigma\right) \\
&= [\![f\,(t_1, \ldots, t_n)]\!]^I_\sigma
\end{aligned}$$

□

Lemma 31. Let φ be a formula, x be a variable and t be a term such that $t \triangleright x : \varphi$. For every interpretation I and assignment ρ, $I \models_\rho \varphi[x/t]$ iff $I \models_{\rho[x/[\![t]\!]^I_\rho]} \varphi$.

Proof. By structural induction on φ. Again we write σ for $\rho\left[x/[\![t]\!]^I_\rho\right]$.

For atomic formulas $p\,(t_1, \ldots, t_n)$, we use the previous lemma.

$$\begin{aligned}
I \models_\rho p\,(t_1, \ldots, t_n)\,[x/t] &\text{ iff } I \models_\rho p\,(t_1[x/t], \ldots, t_n[x/t]) \\
&\text{iff } \left\langle [\![t_1[x/t]]\!]^I_\rho, \ldots, [\![t_n[x/t]]\!]^I_\rho \right\rangle \in p^I \\
&\text{iff } \left\langle [\![t_1]\!]^I_\sigma, \ldots, [\![t_n]\!]^I_\sigma \right\rangle \in p^I \\
&\text{iff } I \models_\sigma p\,(t_1, \ldots, t_n)
\end{aligned}$$

For the propositional connectives the induction hypothesis directly gives the result:

$$\begin{aligned}
&I \models_\rho (\neg\psi)[x/t] \\
&\quad\text{iff } I \models_\rho \neg(\psi[x/t]) \\
&\quad\text{iff } I \not\models_\rho \psi[x/t] \\
&\quad\text{iff } I \not\models_\sigma \psi \\
&\quad\text{iff } I \models_\sigma \neg\psi
\end{aligned}
\qquad\qquad
\begin{aligned}
&I \models_\rho (\psi \to \gamma)[x/t] \\
&\quad\text{iff } I \models_\rho (\psi[x/t]) \to (\gamma[x/t]) \\
&\quad\text{iff } I \not\models_\rho \psi[x/t] \text{ or } I \models_\rho \gamma[x/t] \\
&\quad\text{iff } I \not\models_\sigma \psi \text{ or } I \models_\sigma \gamma \\
&\quad\text{iff } I \models_\sigma \psi \to \gamma
\end{aligned}$$

Finally we consider the case of universally quantified formulas. The case of $\forall x\psi$ is trivial, since $(\forall x\psi)[x/t] = \forall x\psi$ and any assignment that is x-equivalent to ρ is also x-equivalent to σ. So we consider only the case of $\forall y\psi$, where we additionally know that $y \notin V(t)$. In this case, we can identify assignments that are y-equivalent to ρ with assignments that are y-equivalent to σ by means of the bijection $\theta \mapsto \theta\left[x/[\![t]\!]^I_\rho\right]$. Since $y \notin V(t)$, $[\![t]\!]^I_\rho = [\![t]\!]^I_\theta$, and therefore $\theta\left[x/[\![t]\!]^I_\rho\right] = \theta\left[x/[\![t]\!]^I_\theta\right]$. Thus the induction hypothesis applies, i.e., $I \models_\theta \psi[x/t]$ iff $I \models_{\theta[x/[\![t]\!]^I_\theta]} \psi$, and therefore $I \models_\rho (\forall y\psi)[x/t]$ iff $I \models_\sigma \forall y\psi$. □

Example. Earlier we saw that $f(a) \triangleright y : p(x, y) \to \forall z.p(z, y)$. Let I be an interpretation and ρ be an assignment. Then $[\![f(a)]\!]^I_\rho = f^I\left(a^I\right)$, and so $\rho\left[y/[\![f(a)]\!]^I_\rho\right] = \rho\left[y/f^I\left(a^I\right)\right]$. We will call this substitution σ. Also, $(p(x, y) \to \forall z.p(z, y))\,[y/f(a)]$ is the formula $p(x, f(a)) \to \forall z.p(z, f(a))$.

We have that:

$$\begin{aligned}
I \models_\rho p(x, f(a)) \to \forall z.p(z, f(a)) &\text{ iff } I \not\models_\rho p(x, f(a)) \text{ or } I \models_\rho \forall z.p(z, f(a)) \\
&\text{iff } I \not\models_\rho p(x, f(a)) \text{ or } I \models_\theta p(z, f(a)) \qquad \text{for any } \theta \equiv_z \rho \\
&\text{iff } \left\langle [\![x]\!]^I_\rho, [\![f(a)]\!]^I_\rho \right\rangle \notin p^I \text{ or } \left\langle [\![z]\!]^I_\theta, [\![f(a)]\!]^I_\theta \right\rangle \in p^I \\
&\text{iff } \left\langle \rho(x), f^I\,(a) \right\rangle \notin p^I \text{ or } \left\langle \theta(z), f^I\,(a) \right\rangle \in p^I
\end{aligned}$$

whereas $I \models_\sigma p(x,y) \to \forall z.p(z,y)$ iff $I \not\models_\sigma p(x,y)$ or $I \models_\sigma \forall z.p(z,y)$
iff $I \not\models_\sigma p(x,y)$ or $I \models_{\theta'} p(z,y)$ for any $\theta' \equiv_z \sigma$
iff $\langle [\![x]\!]^I_\sigma, [\![y]\!]^I_\sigma \rangle \notin p^I$ or $\langle [\![z]\!]^I_{\theta'}, [\![y]\!]^I_{\theta'} \rangle \notin p^I$
iff $\langle \sigma(x), \sigma(y) \rangle \notin p^I$ or $\langle \theta'(z), \theta'(y) \rangle \notin p^I$

Since $\theta' \equiv_z \sigma$, we have that $\theta'(y) = \sigma(y) = f^I\left(a^I\right)$. Also by construction $\sigma(x) = \rho(x)$. Therefore the last condition is equivalent to $(\langle \rho(x), f^I\left(a^I\right)\rangle \in p^I$ or $\langle \theta'(z), f^I\left(a^I\right)\rangle \notin p^I$. Since the conditions have to hold for any values of both $\theta(z)$ and $\theta'(z)$, they are equivalent, and therefore $I \models_\rho p(x, f(a)) \to \forall z.p(z, f(a))$ iff $I \models_\sigma p(x,y) \to \forall z.p(z,y)$. ◁

Exercise 17. Repeat the analysis in the previous example using the term $g(x,y)$; observe that $g(x,y) \triangleright y : p(x,y) \to \forall z.p(z,y)$.

Exercise 18. We also saw that $z \not\Vdash y : p(x,y) \to \forall z.p(z,y)$. Consider an arbitrary interpretation I and an assignment ρ and use the definition of the semantics of first-order logic to simplify the conditions $I \models_\rho (p(x,y) \to \forall z.p(z,y))[y/z]$ and $I \models_{\rho[y/[\![z]\!]^I_\rho]} p(x,y) \to \forall z.p(z,y)$.

Why are these two statements not equivalent? What step of the proof of Lemma 31 fails?

Find an example of concrete I and ρ for which $I \models_\rho (p(x,y) \to \forall z.p(z,y))[y/z]$ but $I \not\models_{\rho[y/[\![z]\!]^I_\rho]} p(x,y) \to \forall z.p(z,y)$.

The examples in this section may seem like extremely complicated ways of solving very simple problems. This is because we chose interpretations that are very close to their suggested meaning, and therefore they are easy to understand intuitively. However, several different situations arise in practice. First, we may be working with an interpretation that we do not intuitively understand, and for which we need to be able to evaluate terms and decide on formulas' satisfiability in a systematic way. Second, we need to avoid being misled by our intuition and remember that, when we write formulas with an interpretation in mind, we still need to remember that there may be other valid interpretations that can have surprising consequences.

Therefore, we conclude this section by again looking at our example terms and formulas, but now using the strange alternative interpretations J_F and J_N. Note that we can reuse a lot of the work done above, since the recursive processing of terms and formulas is independent of the actual interpretation under consideration.

Example F.7. We now revisit our example on family relations using the unintuitive interpretation J_F. We consider an assignment ρ such that $\rho(x) = 12$.

For the terms, we have:

- $[\![\mathsf{Alice}]\!]^{J_F}_\rho = \mathsf{Alice}^{J_F} = 1$;
- $[\![\mathsf{father}(\mathsf{Alice})]\!]^{J_F}_\rho = \mathsf{father}^{J_F}\left(\mathsf{Alice}^{J_F}\right)$, and from the definition of J_F it now follows that $\mathsf{father}^{J_F}\left(\mathsf{Alice}^{J_F}\right) = (\lambda x.x+1)(1) = 2$.
- Likewise, $[\![\mathsf{mother}(\mathsf{father}(\mathsf{Alice}))]\!]^{J_F}_\rho = \mathsf{mother}^{J_F}\left(\mathsf{father}^{J_F}\left(\mathsf{Alice}^{J_F}\right)\right) = (\lambda x.x-1)(2) = 1$. In general, it is the case that $[\![\mathsf{mother}(\mathsf{father}(x))]\!]^{J_F}_\sigma = [\![x]\!]^{J_F}_\sigma$ for any assignment σ and

any variable x, which is not exactly what one might expect based on the syntax.

- $[\![\mathsf{mother}(\mathsf{mother}(x))]\!]_\rho^{J_F} = \mathsf{mother}^{J_F}\left(\mathsf{mother}^{J_F}(\rho(x))\right) = (\lambda x.x-1)((\lambda x.x-1)(\rho(x))) = 12-2=10$.

Now we move on to the formulas.

- $J_F \models_\rho \mathsf{female}(\mathsf{Alice})$ iff $[\![\mathsf{Alice}]\!]_\rho^{J_F} \in \mathsf{female}^{J_F}$ iff $1 \in \{1,2,3\}$ – which is true.

- Similarly, $J_F \models_\rho \mathsf{male}(\mathsf{father}(\mathsf{Alice}))$ iff $[\![\mathsf{father}(\mathsf{Alice})]\!]_\rho^{J_F} \in \mathsf{male}^{J_F}$. This is equivalent to $2 \in \{0,1\}$, which is false.

- Likewise, $J_F \models_\rho \mathsf{male}(\mathsf{father}(x))$ iff $[\![\mathsf{father}(x)]\!]_\rho^{J_F} \in \mathsf{male}^{J_F}$. Since under this interpretation $[\![\mathsf{father}(x)]\!]_\rho^{J_F} = \mathsf{father}^{J_F}(\rho(x)) = (\lambda x.x+1)(\rho(x)) = \rho(x)+1 = 13$, and $13 \notin \{0,1\}$, this formula is false under J_F and ρ.

- From the two previous examples we can also conclude that $J_F \not\models_\rho \forall x.\mathsf{male}(\mathsf{father}(x))$, since $J_F \not\models_\sigma \mathsf{male}(\mathsf{father}(x))$ for ρ (which is x-equivalent to itself) or for $\rho[x/1]$.

- Finally, $J_F \models_\rho \forall x.(\mathsf{married}(\mathsf{Bob},x) \to \mathsf{married}(x,\mathsf{Bob}))$ iff, for any $\sigma \equiv_x \rho$, it is the case that $J_F \models_\sigma \mathsf{married}(\mathsf{Bob},x) \to \mathsf{married}(x,\mathsf{Bob})$, and the latter holds iff $\left\langle \mathsf{Bob}^{J_F}, \sigma(x)\right\rangle \notin \mathsf{married}^{J_F}$ or $\left\langle \sigma(x), \mathsf{Bob}^{J_F}\right\rangle \in \mathsf{married}^{J_F}$. Since $\mathsf{married}^{J_F} = \emptyset$, the first condition always holds. ◁

Example N.7. Similarly, we reexamine the examples in the context of signature Σ_N with the alternative interpretation J_N, under an assignment ρ such that $\rho(x) = 2$.

- $[\![\mathsf{succ}(0)]\!]_\rho^{J_N} = \mathsf{succ}^{J_N}\left(0^{J_N}\right) = (\lambda x.0)(3) = 0$

- $[\![+(3,5)]\!]_\rho^{J_N} = +^{J_N}\left(3^{J_N}, 5^{J_N}\right) = (\lambda xy.y+5)(9,13) = 18$

- $[\![8]\!]_\rho^{J_N} = 8^{J_N} = 19$, so under this interpretation it is *not* the case that $3+5$ and 8 denote the same value.[2]

- $[\![\times(+(x,4),3)]\!]_\rho^{J_N} = \times^{J_N}\left(+^{J_N}\left(\rho(x), 4^{J_N}\right), 3^{J_N}\right) = (\lambda xy.x+y)((\lambda xy.y+5)(2,11),9) = (\lambda xy.x+y)(16,9) = 25$

- $[\![\times(3,+(x,4))]\!]_\rho^{J_N} = \times^{J_N}\left([\![3]\!]_\rho^{J_N}, [\![+(x,4)]\!]_\rho^{J_N}\right) = (\lambda xy.x+y)(9,16) = 25$

We now look at the formulas from the same example.

- $J_N \models_\rho \mathsf{even}(2)$ iff $2^{J_N} \in \mathsf{even}^{J_N}$ iff $7 \in \{0\}$, which does not hold.

- $J_N \models_\rho =(+(3,5),8)$ iff $\left\langle [\![+(3,5)]\!]_\rho^{J_N}, [\![8]\!]_\rho^{J_N}\right\rangle \in =^{J_N}$ iff $18 = 19$, which is not the case.

- $J_N \models_\rho \forall n.n \leq \mathsf{succ}(n)$ iff $J_N \models_\sigma n \leq \mathsf{succ}(n)$ for every $\sigma \equiv_n \rho$. For each σ, this is the case iff $\left\langle [\![n]\!]_\sigma^{J_N}, [\![\mathsf{succ}(n)]\!]_\sigma^{J_N}\right\rangle \in \leq^{J_N}$, which according to the definition of $\leq^{J_N}$ is equivalent to $[\![n]\!]_\sigma^{J_N} \neq [\![\mathsf{succ}(n)]\!]_\sigma^{J_N}$. Since $[\![n]\!]_\sigma^{J_N} = \sigma(n)$ and $[\![\mathsf{succ}(n)]\!]_\sigma^{J_N} = \mathsf{succ}^{J_N}\left([\![n]\!]_\sigma^{J_N}\right) = (\lambda x.0)(\sigma(n)) = 0$, this is not the case if $\sigma(n) = 0$. Therefore $J_N \not\models_\rho \forall n.n \leq \mathsf{succ}(n)$.

[2] Also $8^{J_N} \neq 8$. This is one of the reasons why it is convenient to distinguish between the syntactic term 8 and the natural number 8, which we use in the semantics.

- $J_N \models_\rho \forall y.\mathsf{3}+y=y+\mathsf{3}$ iff $J_N \models_\sigma \mathsf{3}+y=y+\mathsf{3}$ for any $\sigma \equiv_y \rho$. Again, for such σ, $J_N \models_\sigma \mathsf{3}+y=y+\mathsf{3}$ iff $[\![\mathsf{3}+y]\!]_\sigma^{J_N} = [\![y+\mathsf{3}]\!]_\sigma^{J_N}$. We have that

$$\begin{aligned}[\![\mathsf{3}+y]\!]_\sigma^{J_N} &= +^{J_N}\left(\mathsf{3}^{J_N}, \sigma(y)\right) = (\lambda xy.y+5)(9,\sigma(y)) = \sigma(y)+5\\ [\![y+\mathsf{3}]\!]_\sigma^{J_N} &= +^{J_N}\left(\sigma(y), \mathsf{3}^{J_N}\right) = (\lambda xy.y+5)(\sigma(y),9) = 14\end{aligned}$$

  so the stated equality does not hold if we pick σ such that $\sigma(y) \neq 9$. Therefore $J_N \not\models_\rho \forall y.\mathsf{3}+y=y+\mathsf{3}$.

- It is also not the case that $J_N \models_\rho \forall xy.x+y=y+x$: if we choose an assignment $\sigma \equiv_{\{x,y\}} \rho$ and such that $\sigma(x) \neq \sigma(y)$, then $J_N \models_\sigma x+y=y+x$ iff $+^{J_N}(\sigma(x),\sigma(y)) = +^{J_N}(\sigma(y),\sigma(x))$, which simplifies to the false statement $\sigma(y)+5=\sigma(x)+5$.

- $J_N \models_\rho \forall xyz.x+(y\times z)=(x+y)\times z$ iff $J_N \models_\sigma x+(y\times z)=(x+y)\times z$ for every $\sigma \equiv_{\{x,y,z\}} \rho$. As before, this is the case iff $[\![x+(y\times z)]\!]_\sigma^{J_N} = [\![(x+y)\times z]\!]_\sigma^{J_N}$.

  We have that

$$\begin{aligned}[\![x+(y\times z)]\!]_\sigma^{J_N} &= +^{J_N}\left([\![x]\!]_\sigma^{J_N}, \times^{J_N}\left([\![y]\!]_\sigma^{J_N}, [\![z]\!]_\sigma^{J_N}\right)\right)\\ &= (\lambda xy.y+5)\left(\sigma(x), (\lambda xy.x+y)\left(\sigma(y),\sigma(z)\right)\right)\\ &= (\lambda xy.y+5)\left(\sigma(x), \sigma(y)+\sigma(z)\right) = \sigma(y)+\sigma(z)+5\\ [\![(x+y)\times z]\!]_\sigma^{J_N} &= \times^{J_N}\left(+^{J_N}\left([\![x]\!]_\sigma^{J_N}, [\![y]\!]_\sigma^{J_N}\right), [\![z]\!]_\sigma^{J_N}\right)\\ &= (\lambda xy.x+y)\left((\lambda xy.y+5)\left(\sigma(x),\sigma(y)\right), \sigma(z)\right)\\ &= (\lambda xy.x+y)\left(\sigma(y)+5, \sigma(z)\right) = \sigma(y)+5+\sigma(z)\end{aligned}$$

  and the two last expressions are always guaranteed to be equal. Therefore $J_N \models_\rho \forall xyz.x+(y\times z)=(x+y)\times z$.

- We already know that $J_N \models_\rho \forall x\exists y.x+y=\mathsf{0}$ iff, for any $\sigma \equiv_x \rho$, we can find some $\theta \equiv_y \sigma$ such that $J_N \models_\theta x+y=\mathsf{0}$. Furthermore, $J_N \models_\theta x+y=\mathsf{0}$ iff $[\![x+y]\!]_\theta^{J_N} = [\![\mathsf{0}]\!]_\theta^{J_N}$ iff $\theta(y)+5=3$, which is not possible in the domain of the natural numbers.

- Finally, $J_N \models_\rho \exists y\forall x.x+y=\mathsf{0}$ if there is an assignment $\sigma \equiv_y \rho$ such that $J_N \models_\theta x+y=\mathsf{0}$ for any $\theta \equiv_x \sigma$. Again, $J_N \models_\theta x+y=\mathsf{0}$ iff $\theta(y)+5=3$, which is not the case for any $\theta(y) \in \mathbb{N}$. However, in a similar interpretation J'_N with the integers as domain this formula would be true: we can choose σ such that $\sigma(y)=-2$, and since $\theta(y)=\sigma(y)$ we would get $J'_N \models_\theta x+y=\mathsf{0}$. ◁

Exercise 19. Consider a first-order signature with $P_2=\{p\}$ and $P_3=\{q\}$ and the interpretation I where the domain is $\mathbb{N}$ and the predicate symbols are interpreted by $p^I = \{\langle m,n\rangle \mid m \leq n\}$ and $q^I = \{\langle m,n,o\rangle \mid m+n=o\}$. Which of the following formulas are true in I?

(a) $\forall xyz(q(x,y,z) \to q(y,x,z))$ (b) $\forall xy(p(x,y) \to q(x,x,y))$

Exercise 20. Consider a first-order signature where $F_0 = \{\mathsf{a}\}$, $F_2 = \{\mathsf{plus}, \mathsf{mult}\}$, $P_1 = \{\mathsf{even}\}$ and $P_2 = \{\mathsf{equals}\}$. Let J be the interpretation with domain $\mathbb{N}$ where

$$\mathsf{a}^J = 0 \qquad \mathsf{plus}^J = \lambda xy.x + 2y \qquad \mathsf{even}^J = \{x \mid \ x \text{ is even}\}$$
$$\mathsf{mult}^J = \lambda xy.x + 3 \qquad \mathsf{equals}^J = \{\langle x, x\rangle \mid x \in \mathbb{N}\}$$

Decide whether:

(a) $J \models \forall xy(\mathsf{equals}(\mathsf{plus}(x, y), \mathsf{plus}(y, x)))$ (b) $J \models \forall x(\mathsf{even}(\mathsf{mult}(x, \mathsf{a})))$

4.2.2 Validity, entailment and counter-models

In the previous section we saw how to give semantics to formulas in the context of a fixed interpretation. However, as in other logics, we also want to be able to reason about validity and entailment, as well as finding interpretations that make particular formulas false. In this section we illustrate how the techniques for working with the semantics are applied in the more abstract setting where no interpretation is given.

Example. We start by showing that the formula $(\forall x\varphi) \to (\exists x\varphi)$ is valid, for every φ.

To this end, let I be an interpretation and ρ be an assigment. By definition, $I \models_\rho \forall x\varphi$ iff $I \models_\sigma \varphi$ for every $\sigma \equiv_x \rho$. Since in particular $\rho \equiv_x \rho$, this implies that $I \models_\rho \varphi$; hence $I \models_\sigma \varphi$ for some $\sigma \equiv_x \rho$ (namely, ρ).

Therefore, if $I \models_\rho \forall x\varphi$, then also $I \models_\rho \exists x\varphi$. So either $I \not\models_\rho \forall x\varphi$ or $I \models_\rho \exists x\varphi$, or equivalently $I \models_\rho (\forall x\varphi) \to (\exists x\varphi)$. Since ρ is arbitrary, we can conclude that $I \models (\forall x\varphi) \to (\exists x\varphi)$, and since I is arbitrary it follows that $\models (\forall x\varphi) \to (\exists x\varphi)$. ◁

Example. We now show that $\exists x.(p(x) \to \forall y.p(y))$ is a valid formula.

Let I be an interpretation and ρ be an assignment. Then $I \models_\rho \exists x.(p(x) \to \forall y.p(y))$ iff $I \models_\sigma p(x) \to \forall y.p(y)$ for some $\sigma \equiv_x \rho$.

Consider the following two possible cases for set p^I: either $p^I = D^I$, or $p^I \neq D^I$.

- Suppose that $p^I = D^I$, and let $\theta \equiv_y \rho$; then $\theta(y) \in p^I$, so $I \models_\theta p(y)$. Since θ is arbitrary, it follows that $I \models_\rho \forall y.p(y)$, from which $I \models_\rho p(x) \to \forall y.p(y)$. Since $\rho \equiv_x \rho$, we conclude that $I \models_\rho \exists x.(p(x) \to \forall y.p(y))$.

- Suppose that $p^I \neq D^I$. Since $p^I \subset D^I$, there exists $d \in D^I \setminus p^I$. Let $\sigma = \rho[x/d]$. Then $\sigma \equiv_x \rho$, and furthermore $\sigma(x) \notin p^I$, so $I \not\models_\sigma p(x)$, from which it follows that $I \models_\sigma p(x) \to \forall y.p(y)$. Since $\sigma \equiv_x \rho$, we again conclude that $I \models_\rho \exists x.(p(x) \to \forall y.p(y))$.

In both cases $I \models_\rho \exists x.(p(x) \to \forall y.p(y))$. Since ρ is arbitrary, it follows that $I \models \exists x.(p(x) \to \forall y.p(y))$, and since I is arbitrary this means that $\exists x.(p(x) \to \forall y.p(y))$ is valid. ◁

Exercise 21.

(a) Show that the formula $(\forall x\varphi \to \psi) \to (\varphi \to \forall x\psi)$ is valid whenever $x \notin FV(\varphi)$.

(b) Show that the formula $(\forall x\varphi) \to \varphi[x/t]$ is valid whenever $t \triangleright x : \varphi$.

We now look at some valid entailments. As in other logics, we can show that they hold in a forward-deductive way, where we use the hypotheses to derive constraints on the interpretation until we can establish the conclusion, or in a backwards-by-contradiction way, where we assume there is an interpretation that makes the hypotheses true and the conclusion false, and try to derive a contradiction. In first-order logic, there is no obvious advantage from using either method: the tricky part of the proofs is always choosing the right way to instantiate variables in the substitutions that need to be considered. Therefore, we will preferentially prove entailments directly, as it is a bit more intuitive.

Example. As a simple first example, we show that $\{\exists x.p(x), \forall x.(p(x) \to p(a))\} \models p(a)$.

Let I be an interpretation such that $I \models \exists x.p(x)$ and $I \models \forall x.(p(x) \to p(a))$, i.e., for any assignment ρ, $I \models_\rho \exists x.p(x)$ and $I \models_\rho \forall x.(p(x) \to p(a))$.

From $I \models_\rho \exists x.p(x)$, we conclude that $I \models_\sigma p(x)$ for some $\sigma \equiv_x \rho$.

From $I \models_\rho \forall x.(p(x) \to p(a))$, we conclude that $I \models_\theta p(x) \to p(a)$ for every $\theta \equiv_x \rho$; in particular, we can take θ to be the substitution σ. Since $I \models_\sigma p(x)$ and $I \models_\sigma p(x) \to p(a)$, we can conclude that $I \models_\sigma p(a)$, which reduces to $[\![a]\!]^I_\sigma \in p^I$ or $a^I \in p^I$.

But this condition is independent of the particular assignment (since $p(a)$ does not contain variables), so also $I \models_\rho p(a)$. ◁

Example. We now show that $\{\forall x.p(x) \to q(f(x)), \forall x.q(x) \to p(f(x)), p(a)\} \models p(f(f(a))$. Again, let I be an interpretation that makes all hypotheses true, i.e., for any substitution ρ we have that $I \models_\rho \forall x.p(x) \to q(f(x))$, $I \models_\rho \forall x.q(x) \to p(f(x))$ and $I \models_\rho p(a)$.

From $I \models_\rho p(a)$, we know that $a^I \in p^I$. From $I \models_\rho \forall x.p(x) \to q(f(x))$, we know that $I \models_\sigma p(x) \to q(f(x))$ for any $\sigma \equiv_x \rho$; in particular, this is the case for the substitution $\sigma = \rho\left[x/a^I\right]$. Since $I \models_\sigma p(x) \to q(f(x))$ iff $\underbrace{I \not\models_\sigma p(x)}_{\sigma(x)=a^I \notin p^I}$ or $\underbrace{I \models_\sigma q(f(x))}_{f^I(\sigma(x))=f^I(a^I)\in q^I}$ and the first condition does not hold, we conclude that $f^I\left(a^I\right) \in q^I$.

Now we consider the hypothesis $I \models_\rho \forall x.q(x) \to p(f(x))$, which equivalently states that $I \models_\theta q(x) \to p(f(x))$ for any $\theta \equiv_x \rho$, and consider the substitution $\theta = \rho\left[x/f^I\left(a^I\right)\right]$. As above, it is the case that $I \models_\theta q(x) \to p(f(x)$ iff $\underbrace{I \not\models_\theta q(x)}_{\theta(x)=f^I(a^I)\notin q^I}$ or $\underbrace{I \models_\theta p(f(x))}_{f^I(\theta(x))=f^I(f^I(a^I))\in p^I}$. Again we know that the former option is not the case, so we conclude that $f^I\left(f^I\left(a^I\right)\right) \in p^I$.

But $f^I\left(f^I\left(a^I\right)\right) \in p^I$ is equivalent to $I \models_\rho p(f(f(a)))$, so we conclude that this must also hold. Since ρ is arbitrary, $I \models p(f(f(a)))$, and thus the desired entailment holds. ◁

A particular interpretation that satisfies the hypotheses of the entailment in the previous example is the interpretation I over the natural numbers where $a^I = 0$, $f^I = \lambda x.x + 1$, $p^I = \{2n \mid n \in \mathbb{N}\}$ and $q^I = \{2n + 1 \mid n \in \mathbb{N}\}$, i.e., f^I is the successor function, p^I is the set of even numbers and q^I is the set of odd numbers. The two hypotheses correspond to the mutually recursive definition of even and odd numbers, and the third hypothesis states that 0 is an even number; from this we can derive that 2 is also even.

Example. We show that $\{\forall x.(p(x) \vee q(x)), \forall x.(q(x) \to p(f(x)))\} \models \exists y.p(y)$.

In this example we need to reason by cases. As before, we start by considering an interpretation I that makes the set of hypotheses true, i.e., such that, for any assignment ρ, $I \models_\rho \forall x.(p(x) \vee q(x))$ and $I \models_\rho \forall x.(q(x) \to p(f(x)))$.

From $I \models_\rho \forall x.(p(x) \vee q(x))$, we know that $I \models_\sigma p(x) \vee q(x)$ for any $\sigma \equiv_x \rho$. There are two cases to consider.

- If $I \models_\sigma p(x)$, then $\sigma(x) \in p^I$. Consider the assignment $\theta = \rho[y/\sigma(x)]$. By construction $\theta(y) \in p^I$, so $I \models_\theta p(y)$, and also by construction $\theta \equiv_y \rho$. Therefore $I \models_\rho \exists y.p(y)$.
- If $I \models_\sigma q(x)$, we need to use the second hypothesis. Since $I \models_\rho \forall x.(q(x) \to p(f(x)))$ and $\rho \equiv_x \sigma$, we can conclude that $I \models_\sigma q(x) \to p(f(x))$. Therefore we necessarily have that $I \models_\sigma p(f(x))$, or equivalently that $f^I(\sigma(x)) \in p^I$. In this case, we consider the assignment $\theta = \rho[y/f^I(\sigma(x))]$. By construction we again have that $\theta(y) \in p^I$, so $I \models_\theta p(y)$, and that $\theta \equiv_y \rho$. Therefore we again conclude that $I \models_\rho \exists y.p(y)$.

In both cases $I \models_\rho \exists y.p(y)$, and since ρ is arbitrary it is always the case that $I \models \exists y.p(y)$. Therefore the initial entailment holds. ◁

Our last example, which we state as a lemma, shows that free variables are implicitly treated by entailment as though they were universally quantified.

Lemma 32. For any formula φ and variable x, $\{\varphi\} \models \forall x\varphi$.

Proof. Let I be an interpretation such that $I \models \varphi$. Then $I \models_\rho \varphi$ for any assignment ρ.

To show that $I \models \forall x\varphi$, consider an assignment σ. By definition, $I \models_\sigma \forall x\varphi$ iff $I \models_\theta \varphi$ for all $\theta \equiv_x \sigma$. But we already established that $I \models_\theta \varphi$ for any θ, so this is clearly the case. □

In light of this result, we can leave out universal quantifiers at the outermost level in formulas without affecting entailments. Nevertheless, it is good practice to include them explicitly for readability.

Exercise 22. Show that each of the following entailments hold.

(a) $\{r(b)\} \models \exists z.r(z)$

(b) $\{\forall x.(p(x) \to p(f(x))), \exists y.p(y)\} \models \exists x.p(f(f(x)))$

(c) $\{p(a), \forall x.(\exists y.p(x) \wedge q(x,y)) \to r(x)\} \models q(a,b) \to r(a)$

Finally, we discuss how to build counterexamples by looking at some falsifiable formulas and some entailments that do not hold.

Example. Consider now the formula $(\exists x.p(x)) \to (\forall x.p(x))$. This formula is not valid. In order to construct an interpretation that makes it false, we can start by analysing its semantics to understand how such an interpretation should look.

$$I \not\models_\rho (\exists x.p(x)) \to (\forall x.p(x)) \text{ iff } \underbrace{I \models_\rho \exists x.p(x)}_{I \models_\sigma p(x) \text{ for some } \sigma \equiv_x \rho} \text{ and } \underbrace{I \not\models_\rho (\forall x.p(x))}_{I \not\models_\theta p(x) \text{ for some } \theta \equiv_x \rho}$$

or, in other words, if we can find two assignments σ and θ, both x-equivalent to ρ, such that $\sigma(x) \in p^I$ and $\theta(x) \notin p^I$.

This clearly requires that the domain have at least two distinct elements. So let us try to construct an interpretation I with $D^I = \{\bullet, \circ\}$. We choose $p^I = \{\bullet\}$, and define $\sigma = \rho[x/\bullet]$

and $\theta = \rho[x/\circ]$. These assignments satisfy the conditions we needed, so we conclude that $I \not\models_\rho (\exists x.p(x)) \to (\forall x.p(x))$, as desired. Therefore the formula $(\exists x.p(x)) \to (\forall x.p(x))$ is not valid. ◁

Example. Consider the entailment $\{\forall x \exists y.r(x,y)\} \models \exists x.r(x,x)$.

In order to build an interpretation I such that $I \models \forall x \exists y.r(x,y)$ but $I \not\models \exists x.r(x,x)$, we start by computing the semantic conditions on I that these formulas entail.

Let ρ be an assignment. $I \models_\rho \forall x \exists y.r(x,y)$ iff $I \models_\sigma \exists y.r(x,y)$ for any $\sigma \equiv_x \rho$. In turn, given such a σ, this is equivalent to there existing $\theta \equiv_y \sigma$ for which $I \models_\theta r(x,y)$ or, equivalently, such that $\langle \theta(x), \theta(y) \rangle \in r^I$. From $\theta \equiv_y \sigma$ we know that $\theta(x) = \sigma(x)$, so this condition can be rewritten as $\langle \sigma(x), \theta(y) \rangle \in r^I$.

Since $\sigma(x)$ is arbitrary, it must be the case that for any element $d \in D^I$ it is possible to find another element $d' \in D^I$ such that $\langle d, d' \rangle \in r^I$. Furthermore, these elements must always be distinct: if for some $d \in D^I$ it is the case that $\langle d, d \rangle \in r^I$, then $I \models_{\sigma'} r(x,x)$ for $\sigma' = \rho[x/d]$; since $\rho \equiv_x \sigma'$ it would follow that $I \models_\rho \exists x.r(x,x)$, and since this formula is closed we would have $I \models \exists x.r(x,x)$.

Since $D^I \neq \emptyset$, it must contain at least two distinct elements. Two elements are also enough: by taking $D^I = \{\circ, \diamond\}$ and setting $r^I = \{\langle \circ, \diamond \rangle, \langle \diamond, \circ \rangle\}$ we can fulfill all the required conditions:

- if $\sigma(x) = \circ$, then we define $\theta(y) = \diamond$ to ensure that $\langle \sigma(x), \theta(y) \rangle \in r^I$;
- conversely, if $\sigma(x) = \diamond$, then we define $\theta(y) = \circ$ and again obtain that $\langle \sigma(x), \theta(y) \rangle \in r^I$;
- finally, both $\langle \circ, \circ \rangle \notin r^I$ and $\langle \diamond, \diamond \rangle \notin r^I$.

Thus I shows that $\{\forall x \exists y.r(x,y)\} \not\models \exists x.r(x,x)$. ◁

Exercise 23. Show that the following formulas are not valid.

(a) $p(x) \to \forall y.p(y)$

(b) $(\exists x.p(x)) \vee (\forall x.p(x))$

(c) $\forall x \exists y.q(x,y) \to \forall z.\neg q(z,z)$

(d) $(\forall x.(p(x) \to q(x))) \to (p(x) \to \forall x.q(x))$

Exercise 24. Show that the following entailments do not hold.

(a) $\{\exists x.r(a,x), \exists y.r(y,a)\} \models r(a,a)$

(b) $\{\forall x.r(x, f(x))\} \models \forall x.r(f(x),x)$

We conclude this section with some comments on the variable convention. Earlier we stated that, for any formula φ, we can find another formula ψ that is logically equivalent to φ and satisfies the variable convention. Formula ψ can be constructed as follows: go through the

structure of φ, and for every subformula $\forall x\gamma$ where $x \in FV(\varphi)$, replace $\forall x\gamma$ with $\forall x'.\gamma[x/x']$, where $x' \in \mathcal{X} \setminus (FV(\varphi) \cup V(\gamma))$.

Formulas such as ψ and φ, which can be transformed into each other by recursively replacing some subformulas $\forall x\gamma$ by $\forall x'.\gamma[x/x']$, where x' is fresh, are said to be *α-equivalent*. The semantics of first-order logic is oblivious to α-equivalence, in the sense of the following lemma.

Lemma 33. Let φ be a first-order formula, x be a variable, and y be a variable that does not occur in φ. For every interpretation I and assignment ρ, $I \models_\rho \forall x\varphi$ iff $I \models_\rho \forall y\varphi[x/y]$.

Proof (sketch). Let I be an interpretation and ρ be an assignment. Then $I \models_\rho \forall x\varphi$ iff $I \models_\sigma \varphi$ for every $\sigma \equiv_x \rho$, and $I \models_\rho \forall y\varphi[x/y]$ iff $I \models_\sigma \varphi[x/y]$ for every $\sigma \equiv_y \rho$.

Given an assignment $\sigma \equiv_x \rho$, let $\theta = \rho[y/\sigma(x)]$. Then $\theta \equiv_y \rho$, and furthermore we can prove by structural induction that $I \models_\sigma \varphi$ iff $I \models_\theta \varphi[x/y]$. Conversely, given an assignment $\theta \equiv_y \rho$, we can define $\sigma = \rho[x/\theta(y)]$, obtaining an assignment $\sigma \equiv_x \rho$ for which the same property holds. This establishes the thesis. □

From Lemma 32, we know that $\{p(x)\} \models \forall x.p(x)$, and since $\forall x.p(x)$ is α-equivalent to $\forall y.p(y)$ it also follows that $\{p(x)\} \models \forall y.p(y)$. This illustrates the fundamental difference between variables and constants: since the interpretation of constants is fixed, we do not get a similar entailment when we replace x by a constant.

Example. Consider the entailment $\{p(a)\} \models \forall x.p(x)$. To see that it does not hold, let I be an interpretation with a two-element domain $D^I = \{\circledcirc, ✠\}$ such that $a^I = ✠$ and $p^I = \{✠\}$.

For any ρ, $[\![a]\!]^I_\rho = a^I = ✠$, so $I \models_\rho p(a)$ and thus $I \models p(a)$. However, taking $\sigma = \rho[x/\circledcirc]$ we have that $\sigma(x) \notin p^I$, so $I \not\models_\sigma p(x)$. Since $\sigma \equiv_x \rho$, it follows that $I \not\models_\rho \forall x.p(x)$, and thus $I \not\models \forall x.p(x)$. ◁

Exercise 25. Finish the proof of Lemma 33.

Exercise 26. Formally define the syntactic transformation explained in the text, and prove that:

(a) it always generates a formula that respects the variable convention;

(b) it always generates a formula equivalent to the original one.

4.3 Tableaux calculus

The tableaux calculus for first-order logic extends the calculus for propositional logic with four new rules to deal with the two new quantifiers. As before, a tableau for a set of formulas Γ is a labeled tree whose root node is Γ, and where every node has descendants generated by application of one of the tableau rules.

The new rules for quantifiers have to deal with variables, and for these reason they include a new feature: side conditions, which restrict how variables can be instantiated.

The new rules are the following.

$$\dfrac{\Gamma, \forall x\varphi}{\Gamma, \forall x\varphi, \varphi[x/t]}\,(\forall)[x/t] \qquad \dfrac{\Gamma, \neg(\forall x\varphi)}{\Gamma, \neg\varphi[x/c]}\,(\neg\forall) \qquad \dfrac{\Gamma, \exists x\varphi}{\Gamma, \varphi[x/c]}\,(\exists) \qquad \dfrac{\Gamma, \neg(\exists x\varphi)}{\Gamma, \neg(\exists x\varphi), \neg\varphi[x/t]}\,(\neg\exists)[x/t]$$

The side conditions for these rules are as follows.

- In rules $(\forall)$ and $(\neg\exists)$, t is any term satisfying $t \triangleright x : \varphi$; since these rules are parameterized on t, we also indicate the term we choose next to the rule name.

  These rules do not remove the formula that they target, which has important consequences that we will discuss later.

- In rules $(\neg\forall)$ and $(\exists)$, c is a fresh variable that does not appear in the node where the rule is being applied. The reason for calling it c rather than a more suggestive y is that this fresh variable behaves as a constant, in the sense that it stands for the unknown value that makes the relevant formula true.

As before, we say that a leaf in a tableau is said to be *contradictory* or an *absurd* if it contains both a formula φ and its negation $\neg\varphi$, and that a tableau is *closed* if all its leaves are either contradictory or no more rules can be applied. However, in first-order logic the condition that "no more rules can be applied" to a node is much stronger than in propositional logic, since rules $(\forall)$ and $(\neg\exists)$ can be applied indefinitely to generate new formulas.

Example. The following tableau shows that $\vdash_S (\forall x\varphi) \to (\exists x\varphi)$, for every formula φ. As before, we underline the formula to which the rule is being applied, and box contradictory formulas in a leaf.

$$\dfrac{\dfrac{\dfrac{\dfrac{\underline{\neg((\forall x\varphi) \to (\exists x\varphi))}}{\underline{\forall x\varphi}, \neg(\exists x\varphi)}\,(\neg\to)}{\forall x\varphi, \varphi, \underline{\neg(\exists x\varphi)}}\,(\forall)[x/x]}{\forall x\varphi, \boxed{\varphi}, \neg(\exists x\varphi), \boxed{\neg\varphi}}\,(\neg\exists)[x/x]$$

Observe that in the applications of rules $(\forall)$ and $(\neg\exists)$ we replaced x with itself; this is allowed, since $x \triangleright x : \varphi$ for every formula φ. ◁

Example. In the next example, we show that $\vdash_S \exists x.(p(x) \to \forall y.p(y))$.

$$\underline{\neg\exists x.(p(x) \to \forall y.p(y))}$$
$$\Big|\ (\neg\exists)[x/x]$$
$$\neg\exists x.(p(x) \to \forall y.p(y)), \underline{\neg(p(x) \to \forall y.p(y))}$$
$$\Big|\ (\neg\to)$$
$$\neg\exists x.(p(x) \to \forall y.p(y)), p(x), \underline{\neg\forall y.p(y)}$$
$$\Big|\ (\neg\forall)$$
$$\underline{\neg\exists x.(p(x) \to \forall y.p(y))}, p(x), \neg p(c)$$
$$\Big|\ (\neg\exists)[x/c]$$
$$\neg\exists x.(p(x) \to \forall y.p(y)), \underline{\neg(p(c) \to \forall y.p(y))}, p(x), \neg p(c)$$
$$\Big|\ (\neg\to)$$
$$\neg\exists x.(p(x) \to \forall y.p(y)), \boxed{p(c)}, \neg\forall y.p(y), p(x), \boxed{\neg p(c)}$$

As before, the first application of $(\neg\exists)$ is allowed because $x \triangleright x : \varphi$ for any formula φ. The second application is also allowed because c does not occur in the original formula, so in particular it is not quantified over in it.

This example illustrates the need to keep the original formulas in rules $(\forall)$ and $(\neg\exists)$: we may need to instantiate them several times. In this case, x needs to be replaced with the variable generated from $\neg\forall y.p(y)$, which is only known after the inner formula is expanded. $\triangleleft$

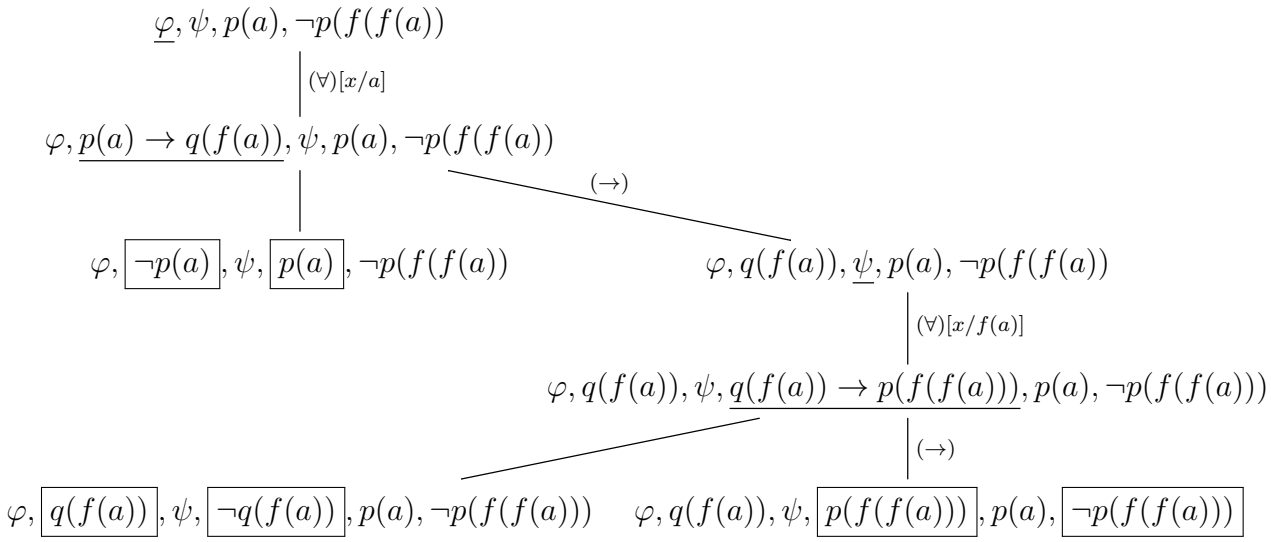
Example. We now show that $\{\exists x.p(x), \forall x.(p(x) \to p(a))\} \vdash_S p(a)$. Recall that the root of the tableau proof contains all the hypotheses and the negation of the formula on the right.

$$\underline{\exists x.p(x)}, \forall x.(p(x) \to p(a)), \neg p(a)$$
$$\Big|\ (\exists)$$
$$p(c), \underline{\forall x.(p(x) \to p(a))}, \neg p(a)$$
$$\Big|\ (\forall)[x/c]$$
$$p(c), \forall x.(p(x) \to p(a)), \underline{p(c) \to p(a)}, \neg p(a)$$

$$\boxed{p(c)}, \forall x.(p(x) \to p(a)), \boxed{\neg p(c)}, \neg p(a) \qquad\qquad p(c), \forall x.(p(x) \to p(a)), \boxed{p(a)}, \boxed{\neg p(a)}$$

Since both leaves are contradictory, this proves that $\{\exists x.p(x), \forall x.(p(x) \to p(a))\} \vdash_S p(a)$. $\triangleleft$

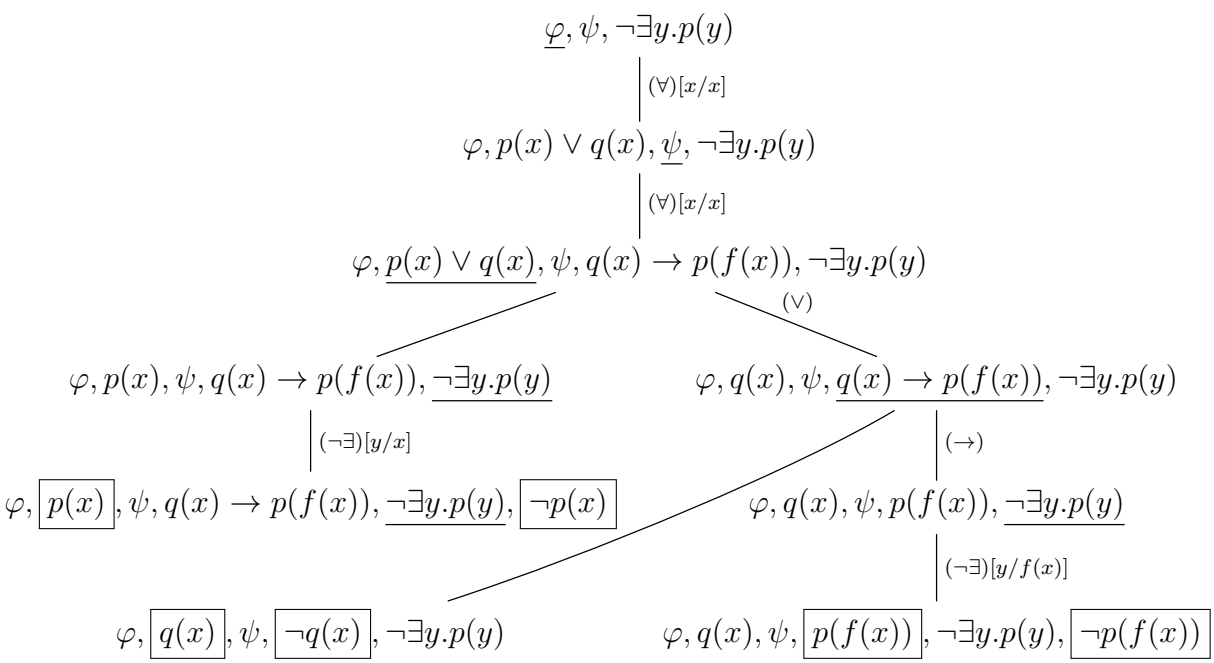
Example. As a next example, we now show that $\{\varphi, \psi, p(a)\} \vdash_S p(f(f(a))$, where φ stands for formula $\forall x.p(x) \to q(f(x))$ and ψ for $\forall x.q(x) \to p(f(x))$.



Again we point out that both applications of $(\forall)$ satisfy the side condition, in this case because x is being replaced by a term with no variables. $\triangleleft$

This example illustrates one of the challenges that arise when one tries to automate first-order reasoning: finding the right instantiations for universally quantified variables. Unlike in propositional logic, where essentially nothing can go wrong, finding a closed tableau requires carefully choosing when to apply rules $(\forall)$ and $(\neg\exists)$, and with which instantiations of the variables.

Example. Finally we show that $\{\underbrace{\forall x.(p(x) \vee q(x))}_{\varphi}, \underbrace{\forall x.(q(x) \to p(f(x)))}_{\psi}\} \vdash_S \exists y.p(y)$.



In this example, the instantiation for rule $(\neg\exists)$ is different in both branches of the proof. $\triangleleft$

Exercise 27. Show that:

(a) $\{r(b)\} \vdash_S \exists z.r(z)$

(b) $\{\forall x.(p(x) \to p(f(x))), \exists y.p(y)\} \vdash_S \exists x.p(f(f(x)))$

(c) $\{p(a), \forall x.(\exists y.p(x) \wedge q(x,y)) \to r(x)\} \vdash_S q(a,b) \to r(a)$

The previous examples suggest that the tableaux calculus for first-order logic is sound and complete. Indeed, the proof of soundness for this calculus is a simple extension of the proof of soundness for the tableaux calculus for propositional logic. We point out that Lemma 4 still holds for all the propositional connectives, as long as we replace the valuation V by an interpretation together with an assignment; and that the inductive construction of Lemma 5 immediately follows through. So the only new ingredients are the four new cases of Lemma 4 generated by the four new rules.

We state the version of this lemma for first-order logic and prove two representative cases. In order to deal with the new variables introduced by rules ($\exists$) and ($\neg\forall$), the assignment may change as the tableau gets larger.

Lemma 34. Let Γ be a set of formulas, I be an interpretation, ρ be an assignment over I and r be a tableaux rule. Then $I \models_\rho \Gamma$ iff $I \models_\sigma \Delta$ for some Δ among the sets of formulas obtained by applying r to Γ and some σ such that $\rho(x) = \sigma(x)$ for all $x \in FV(\Gamma)$.

Proof. We consider the case of rules ($\forall$) and ($\exists$).

- Suppose that $I \models_\rho \Gamma, \forall x\varphi$, and let t be a term such that $t \triangleright x : \varphi$. Since $I \models_\rho \forall x\varphi$, it is also the case that $I \models_\theta \varphi$ for any $\theta \equiv_x \rho$. In particular, $I \models_{\rho[x/[\![t]\!]^I_\rho]} \varphi$, which by Lemma 31 is equivalent to $I \models_\rho \varphi[x/t]$.

  The converse direction is straightforward, since the formula $\forall x\varphi$ is not removed in the descendent node.

- Suppose that $I \models_\rho \Gamma, \exists x\varphi$. Then there exists an assignment $\theta \equiv_x \rho$ such that $I \models_\theta \varphi$.

  Let c be a variable not appearing in Γ, and consider the assignment $\sigma = \rho[c/\theta(x)]$. Note that this assignment coincides with ρ on all variables in Γ. Since c does not appear in φ, it is guaranteed that $c \triangleright x : \varphi$; therefore Lemma 31 applies, yielding $I \models_\sigma \varphi[x/c]$ iff $I \models_{\sigma[x/[\![c]\!]^I_\sigma]} \varphi$. Since $[\![c]\!]^I_\sigma = \theta(x)$, the assignments $\sigma\left[x/[\![c]\!]^I_\sigma\right]$ and θ are actually the same, from which we can conclude that $I \models_\sigma \varphi[x/c]$. □

Lemma 35. Let Γ be a set of formulas, I be an interpretation, ρ be an assignment over I and T be a tableau with root labeled by Γ. Then $I \models_\rho \Gamma$ iff $I \models_\sigma \Delta$ for some Δ labeling a leaf of T and some σ such that $\sigma(x) = \rho(x)$ for all $x \in FV(\Gamma)$.

Proof. By induction on T, as in the proof of Lemma 5. □

Theorem 26 (Soundness)**.** For any set of formulas Γ and formula φ, if $\Gamma \vdash_S \varphi$, then $\Gamma \models \varphi$.

Proof. Let I be an interpretation such that $I \models \Gamma$. If $I \not\models \varphi$, then there exists an assignment ρ such that $I \models_\rho \neg\varphi$. Note that from $I \models \Gamma$ we also know that $I \models_\rho \Gamma$.

Therefore $I \models_\rho \Gamma \cup \{\neg\varphi\}$. By Lemma 35, we conclude that for every leaf of a tableau for $\Gamma \cup \{\neg\varphi\}$ there exists an assignment σ that satisfies one of its leaves. But since $\Gamma \vdash_S \varphi$, there exists a closed tableau for $\Gamma \cup \{\neg\varphi\}$; all leaves of this tableau are contradictory, so none of them can be satisfied by I together with any assignment.

Thus $I \models_\rho \varphi$. □

Also similarly to other logics, from a non-contradictory leaf of a tableau to which no rules can be applied we can generate a counter-example to the original formula or entailment.

Example. We now build a tableau for the formula $\neg((\exists x.p(x)) \to \forall x.p(x))$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\neg((\exists x.p(x)) \to \forall x.p(x))}{\Big|\,(\neg\to)}}{\exists x.p(x), \neg\forall x.p(x)}}{\Big|\,(\exists)}}{p(c), \dfrac{\neg\forall x.p(x)}{\Big|\,(\neg\forall)}}$$

$$p(c), \neg p(d)$$

(Observe that the variable introduced in the application of rule $(\neg\forall)$ is distinct from the one introduced by $(\exists)$.)

This tableau shows that $\not\vdash_S (\exists x.p(x)) \to (\forall x.p(x))$. In order to build a model that makes this formula false, we need to choose a domain with at least two distinct constants – one corresponding to c, another to d. We can choose a domain consisting of precisely these terms: $D^I = \{c, d\}$, and take the assignment ρ such that $\rho(c) = c$ and $\rho(d) = d$. Furthermore, $p^I = \{c\}$. This interpretation together with this assignment make all the formulas in the leaf of the tableau true, therefore by Lemma 35 it also falsifies $(\exists x.p(x)) \to (\forall x.p(x))$. ◁

The model constructed in the previous example is an example of a *term* model: a model whose domain is a set of first-order terms over a given signature, and any closed term is interpreted to itself. The simplest example of a term model over a signature Σ is the *Herbrand model*, whose domain is exactly the set of closed terms over Σ. The model in the example above is not a Herbrand model, since the terms c and d are not constants in the original signature.

Term models play an important role in the model theory of first-order logic, and we will see more examples of them later.

Unfortunately, there are situations where we cannot automatically build a counter-model for a falsifiable formula or for an invalid entailment by building a tableau, because the tableau that we construct does not have any leaves. This is typically the case in the presence of formulas that combine universal and existential quantifiers, since eliminating these formulas keeps introducing new variables, giving new possibilities of continuing the tableau.

Example. We now build a tableau to try to show that $\{\forall x \exists y.r(x, y)\} \vdash_S \exists x.r(x, x)$. We saw previously that $\{\forall x \exists y.r(x, y)\} \not\models \exists x.r(x, x)$, therefore by soundness of the tableaux calculus for first-order logic we know that no closed tableau for this judgement can exist.

Our tableau starts with the node $\forall x \exists y.r(x, y), \neg\exists x.r(x, x)$. There are two rules that we can apply here: $(\forall)$ and $(\neg\exists)$. Since applying rule $(\neg\exists)$ only gives us atomic formulas, we focus on

what happens when we apply rule (∀). We need to choose a value for x, and since there are no variables or constants in play yet, we instantiate x with itself and apply rule (∃) to the result, obtaining

$$\begin{array}{c} \underline{\forall x \exists y.r(x,y)}, \neg\exists x.r(x,x) \\ \Big| \ (\forall)[x/x] \\ \forall x \exists y.r(x,y), \underline{\exists y.r(x,y)}, \neg\exists x.r(x,x) \\ \Big| \ (\exists) \\ \forall x \exists y.r(x,y), r(x,c), \neg\exists x.r(x,x) \end{array}$$

At this stage, we can apply rule (¬∃) twice to derive $\neg r(x,x)$ and $\neg r(c,c)$, but these formulas do not generate any contradictions. The interesting observation is that we can apply rule (∀) again, but now instantiating x with the new variable c that we obtained previously.

$$\begin{array}{c} \underline{\forall x \exists y.r(x,y)}, r(x,c), \neg\exists x.r(x,x) \\ \Big| \ (\forall)[x/c] \\ \forall x \exists y.r(x,y), \underline{\exists y.r(c,y)}, r(x,c), \neg\exists x.r(x,x) \\ \Big| \ (\exists) \\ \forall x \exists y.r(x,y), r(c,d), r(x,c), \neg\exists x.r(x,x) \end{array}$$

Now we can apply rule (∀) once more, instantiating x with d, and afterwards apply rule (∃), introducing yet another fresh variable that we can use to instantiate x in another application of rule (∀), after which we can apply rule (∃) to...

What we can *not* do is close the tableau, because every time we apply rule (∀) we generate a new existentially quantified formula, and every time we eliminate that formula we generate a new variable that gives rise to a new possibility of applying rule (∀). Therefore this process never terminates. ◁

The tableau calculus for first-order logic is actually complete, but its completeness is trickier to prove than in the propositional case. The examples above show that a tableau can have infinite branches, and this is also the case even when the root node is a contradictory formula or set of formulas. Furthermore, since free variables are treated as constants, it is not the case that all entailments are provable: a simple example is $\{p(x)\} \models \forall x.p(x)$, which the tableaux rules cannot distinguish from $\{p(a)\} \models \forall x.p(x)$. Since the second entailment does not hold, it is not provable; and therefore neither is the first one.

The result that we do have is that the tableau calculus can prove all valid entailments not containing any free variables. This restriction is not as strong as it sounds; in particular, it can still prove all valid formulas (given that, if φ is valid and x occurs free in φ, then $\varphi[x/c]$ must also be valid for any constant c). The real limitation is that completeness only guarantees that, if $\Gamma \models \varphi$, then there is *a* closed tableau for $\Gamma \cup \{\neg\varphi\}$ where all leaves are contradictions, but not that *all* tableaux for $\Gamma \cup \{\neg\varphi\}$ have this property.

We only prove weak completeness for the tableau calculus (every valid formula is provable). The stronger result about entailments can be derived from compactness (which is easily proved using the Hilbert calculus, in the next section) and the semantic version of the deduction theorem.

Theorem 27 (Completeness)**.** Let φ be a first-order formula. If $\models \varphi$, then $\vdash_S \varphi$.

Proof. We describe a systematic way to build a tableau for $\vdash_S \varphi$ where any infinite branch can be used to obtain (i) an interpretation I (based on a term model) and (ii) an assignment ρ such that $I \not\models_\rho \varphi$. If φ is valid, then $I \not\models_\rho \varphi$ is impossible, so no such infinite branch may exist. Therefore all branches of the tableau are finite, and as such they must end in a contradictory leaf – otherwise there would again be an interpretation and assignment falsifying φ. Thus this method yields a proof that $\vdash_S \varphi$.

The construction is as follows. Let $\mathcal{X}'$ be the set of variables that do not appear bound in φ, and assume an enumeration of all terms that can be built from the signature Σ and $\mathcal{X}'$ (since $\mathcal{F}$ and $\mathcal{X}$ are countable, such an enumeration is guaranteed to exist – we are not requiring e.g. that it be computable). We view the formulas in a node as a list; when expanding a node, we select the first formula in it, and add the generated formulas at the end. When the main connective in the selected formula dictates that we apply rule ($\forall$) or ($\neg\exists$), we instantiate the relevant variable by the first term that has not been considered previously in the application of that rule to that same formula.

Suppose that there is an infinite branch in the tableau thus constructed. We build an interpretation I as follows:

- D^I is the set of all terms over Σ and $\mathcal{X}'$ (not only closed ones);
- every term is mapped to itself;
- p^I is the set of tuples $\langle t_1, \ldots, t_n \rangle$ such that $p(t_1, \ldots, t_n)$ appears in some node in this branch.

Furthermore, we choose the assignment ρ such that $\rho(x) = x$ for each variable x.

We now show that, for every formula ψ occurring in this branch, $I \models_\rho \psi$. The proof is by structural induction on ψ.

- For atomic formulas this is straightforward by construction of I.
- For the propositional connectives other than negation, this follows directly from their semantics and the induction hypothesis.
- For negated formulas, we also need to consider the case of the main connective inside the negation, and in case of a propositional connective the thesis also follows directly from the induction hypothesis.
- Suppose that ψ is $(\forall x\gamma)$ and that $\sigma \equiv_x \rho$. It is necessarily the case that $\sigma(x) \triangleright x : \gamma$, since the bound variables in γ do not appear in D^I. Therefore $\gamma[x/\sigma(x)]$ also appears in some node in the branch, and by induction hypothesis $I \models_\rho \gamma[x/\sigma(x)]$. By Lemma 31 it follows that $I \models_{\rho[x/[\![\sigma(x)]\!]^I_\rho]} \gamma$, and by construction $[\![\sigma(x)]\!]^I_\rho = \sigma(x)$. Therefore $I \models_\sigma \gamma$, and since σ is arbitrary it follows that $I \models_\rho \forall x\gamma$. The case where ψ is $\neg\exists x\gamma$ is similar.
- Finally suppose that ψ is $(\exists x\gamma)$. By construction there is also a node where $\gamma[x/c]$ is present for some c, and again by induction hypothesis $I \models_\rho \gamma[x/c]$. Again Lemma 31 is applicable, yielding $I \models_{\rho[x/[\![c]\!]^I_\rho]} \gamma$, and since $\rho \equiv_x \rho\left[x/[\![c]\!]^I_\rho\right]$ the thesis follows. The case where ψ is $\neg\forall x\gamma$ is similar.

As a consequence, $I \models_\rho \neg\varphi$, which contradicts the assumption that φ is a valid formula. Therefore the tableau constructed by this algorithm cannot have infinite branches. By soundness it cannot have non-contradictory leaves either; so it is a closed tableau. This establishes that $\vdash_S \varphi$. □

Exercise 28. The following formulas are not valid. For each of them, construct a tableau, and use the ones that have non-contradictory leaves to build a counter-example.

(a) $p(x) \to \forall y.p(y)$

(b) $(\exists x.p(x)) \vee (\forall x.p(x))$

(c) $\forall x \exists y.q(x,y) \to \forall z.\neg q(z,z)$

(d) $(\forall x.(p(x) \to q(x))) \to (p(x) \to \forall x.q(x))$

Exercise 29. The following entailments do not hold. For each of them, construct a tableau, and use the ones that have non-contradictory leaves to build a counter-example.

(a) $\{\exists x.r(a,x), \exists y.r(y,a)\} \models r(a,a)$

(b) $\{\forall x.r(x,f(x))\} \models \forall x.r(f(x),x)$

4.4 Hilbert calculus

The classical system to reason about first-order logic is again the Hilbert calculus. Several of the classical results about the proof theory of this logic were originally formulated in terms of this calculus – in particular Gödel's completeness theorem, which was the first proof of completeness of first-order logic.

As we have seen for other logics, in Hilbert calculi it is easy to work with additional axioms. This makes it for many the best calculus for working with first-order *theories* (sets of formulas entailed by a particular set of axioms). In this section we will work with an example theory, and later we will show a particularly interesting extension of first-order logic: the theory of Peano arithmetic, to which Gödel's famous incompleteness theorem applies.

4.4.1 Axioms, rules, soundness and meta-theorems

The Hilbert calculus for first-order logic is an extension of that for propositional logic, and bears many similarities to the Hilbert calculus for modal logic. There are five schematic axioms: the three axioms for propositional logic, and the two new axioms

$$\forall x(\varphi \to \psi) \to (\varphi \to \forall x\psi) \qquad (x \notin FV(\varphi)) \qquad \text{(Ax.4)}$$

$$\forall x\varphi \to \varphi[x/t] \qquad (t \triangleright x : \varphi) \qquad \text{(Ax.5)}$$

which have side-conditions similar to the ones we already encountered in the tableau calculus for first-order logic. As inference rules, we have Modus Ponens (MP) and the new inference rule

$$\text{from } \varphi \text{ infer } \forall x\varphi \qquad \text{Gen}$$

called *generalization*.

As in the case of propositional logic, we assume $\neg$ and $\rightarrow$ to be the only propositional connectives, and assume that $\vee$, $\wedge$ and $\leftrightarrow$ are defined as abbreviations. Likewise, we treat the existential quantifier as an abbreviation – $\exists x\varphi$ stands for $\neg(\forall x.\neg\varphi)$ – and fold it and unfold it as necessary.

It should come as no surprise that the proviso that $t \triangleright x : \varphi$ is necessary to guarantee soundness of (Ax.5).

Example. Let φ be the formula $\exists y.p(x,y)$. Then $(\forall x\varphi) \rightarrow \varphi[x/y]$ is not valid: just consider an interpretation I with domain $D = \{0,1\}$ and $p^I = \{(0,1),(1,0)\}$. For any assignment ρ, it is the case that $I \models_\rho \forall x\varphi$: if $\sigma \equiv_x \rho$, then we can define $\theta = \sigma[y/1-\sigma(x)]$ to ensure that $\langle [\![x]\!]^I_\theta, [\![y]\!]^I_\theta \rangle = \langle \sigma(x), 1-\sigma(x)\rangle \in p^I$. However, $\varphi[x/y]$ is the formula $\exists y.p(y,y)$, and there is no assignment σ such that $\langle [\![y]\!]^I_\sigma, [\![y]\!]^I_\sigma \rangle = \langle \sigma(y), \sigma(y)\rangle \in p^I$. $\triangleleft$

In the particular case that t is x, we already pointed out that the condition $x \triangleright x : \varphi$ always holds, and this axiom becomes simply $(\forall x\varphi) \rightarrow \varphi$.

Axiom (Ax.4) and the new inference rule (Gen) are similar to the axiom K and the inference rule (Nec) of modal logic, respectively. Indeed, it is possible to present the Hilbert calculus for first-order logic using the axiom scheme $(\forall x(\varphi \rightarrow \psi)) \rightarrow ((\forall x\varphi) \rightarrow (\forall x\psi))$ instead of (Ax.4). As we have mentioned earlier, the quantifiers $\forall x$ and $\exists x$ behave in many ways like modalities $\Box$ and $\Diamond$, with (Ax.5) playing the role of the visibility relation in modal logic.

As usual, we write $\Gamma \vdash_L \varphi$ if there is a valid derivation of φ using hypotheses from Γ, and $\vdash_L \varphi$ in the case $\Gamma = \emptyset$. It should come as no surprise that this calculus is sound.

Theorem 28 (Soundness)**.** If $\Gamma \vdash_L \varphi$, then $\Gamma \models \varphi$.

Proof. By induction on the derivation of $\Gamma \vdash_L \varphi$, as for the case of propositional logic.

The new base cases relate to axioms (Ax.4) and (Ax.5), and we already saw that all instances of these axioms are valid formulas as long as their respective provisos hold (Exercise 21).

In the inductive step, there is a new case regarding the application of (Gen). Since Lemma 32 already established that $\{\varphi\} \models \forall x\varphi$, the thesis follows by induction hypothesis and transitivity of entailment (if $\Gamma \models \varphi$ and $\{\varphi\} \models \forall x\varphi$, then $\Gamma \models \forall x\varphi$). $\Box$

We now present a simple example of a derivation in this calculus.

Example. We show that $\{\forall x.(p(x) \rightarrow q(f(x))), \forall x.(q(x) \rightarrow p(f(x))), p(a)\} \vdash_L p(f(f(a)))$.

1. $p(a)$	(Hyp)
2. $(\forall x.(p(x) \rightarrow q(f(x)))) \rightarrow (p(a) \rightarrow q(f(a)))$	(Ax.5)
3. $\forall x.(p(x) \rightarrow q(f(x)))$	(Hyp)
4. $p(a) \rightarrow q(f(a))$	MP(2,3)
5. $q(f(a))$	MP(4,1)
6. $(\forall x.(q(x) \rightarrow p(f(x)))) \rightarrow (q(f(a)) \rightarrow p(f(f(a))))$	(Ax.5)
7. $\forall x.(q(x) \rightarrow p(f(x)))$	(Hyp)
8. $q(f(a)) \rightarrow p(f(f(a)))$	MP(6,7)
9. $p(f(f(a)))$	MP(8,5)

From soundness of the Hilbert calculus, we can also conclude that

$$\{\forall x.(p(x) \rightarrow q(f(x))), \forall x.(q(x) \rightarrow p(f(x))), p(a)\} \models p(f(f(a)))\,. \quad \triangleleft$$

Before showing some more interesting examples, we prove some results about this calculus that will allow us to simplify writing derivations.

Theorem 29 (Substitution Theorem)**.** Let φ be a valid propositional formula with propositional symbols $p_1, \ldots, p_n$. Then $\vdash_L \varphi[p_1/\psi_1, \ldots, p_n/\psi_n]$, where $\varphi[p_1/\psi_1, \ldots, p_n/\psi_n]$ is obtained from φ by uniformly replacing each p_i by the first-order formula ψ_i.

Proof (sketch). Since φ is valid, we know that $\vdash_L \varphi$ in the Hilbert calculus for propositional logic, as this calculus is complete. Since all axioms of this calculus are also axioms of the Hilbert calculus for first-order logic, and likewise for the inference rule (MP), we can construct a correct derivation of $\varphi[p_1/\psi_1, \ldots, p_n/\psi_n]$ by replacing each p_i by ψ_i uniformly everywhere in the original derivation. □

We denote applications of this theorem in proofs by (Prop).

Exercise 30. Write down a formal proof of Theorem 29, using induction on the derivation of $\vdash_L \psi$.

Using Theorem 29, we can introduce a result to help in reasoning about the existential quantifier.

Lemma 36. For any formula φ, variable x and term t such that $t \triangleright x : \varphi$, the formula $\varphi[x/t] \to \exists x\varphi$ is derivable.

Proof. The derivation is as follows.

1. $(\forall x.(\neg\varphi)) \to \neg\varphi[x/t]$	(Ax.5)
2. $((\forall x.(\neg\varphi)) \to \neg\varphi[x/t]) \to (\varphi[x/t] \to \underbrace{\neg\forall x.(\neg\varphi)}_{\exists x\varphi})$	(Prop)
3. $\varphi[x/t] \to \exists x\varphi$	MP(2,1)

In step 2, we used the propositional tautology $(p \to \neg q) \to (q \to \neg p)$ and Theorem 29, with the instantiation $p \mapsto \forall x.(\neg\varphi)$ and $q \mapsto \varphi[x/t]$. □

We denote applications of this result in future derivations as (Lem.∃). We can now show some more interesting examples of derivations in the Hilbert calculus.

Example. The following derivation shows that $\vdash_L (\forall x\varphi) \to (\exists x\varphi)$.

1. $(\forall x\varphi) \to \varphi$	(Ax.5)
2. $\varphi \to (\exists x\varphi)$	(Lem.∃)
3. $((\forall x\varphi) \to \varphi) \to ((\varphi \to (\exists x\varphi)) \to ((\forall x\varphi) \to (\exists x\varphi)))$	(Prop)
4. $(\varphi \to (\exists x\varphi)) \to ((\forall x\varphi) \to (\exists x\varphi))$	MP(3,1)
5. $(\forall x\varphi) \to (\exists x\varphi)$	MP(4,2)

In step 3, we used the propositional tautology $(p \to q) \to ((q \to r) \to (p \to r))$ with the mapping $p \mapsto \forall x\varphi$, $q \mapsto \varphi$ and $r \mapsto \exists x\varphi$. Note that φ coincides with $\varphi[x/x]$. ◁

Example. We show that $\{\forall x.(p(x) \vee q(x)), \forall x.(q(x) \to p(f(x)))\} \vdash_L \exists y.p(y)$.

1. $\forall x.(p(x) \vee q(x))$	(Hyp)
2. $(\forall x.(p(x) \vee q(x))) \to (p(x) \vee q(x))$	(Ax.5)
3. $p(x) \vee q(x)$	MP(2,1)
4. $p(x) \to \exists y.p(y)$	Lem.$\exists$
5. $\forall x.(q(x) \to p(f(x)))$	(Hyp)
6. $(\forall x.(q(x) \to p(f(x)))) \to (q(x) \to p(f(x)))$	(Ax.5)
7. $q(x) \to p(f(x))$	MP(6,5)
8. $p(f(x)) \to \exists y.p(y)$	Lem.$\exists$
9. $(q(x) \to p(f(x))) \to ((p(f(x)) \to \exists y.p(y)) \to (q(x) \to \exists y.p(y)))$	(Prop)
10. $(p(f(x)) \to \exists y.p(y)) \to (q(x) \to \exists y.p(y))$	MP(9,7)
11. $q(x) \to \exists y.p(y)$	MP(10,8)
12. $(p(x) \to \exists y.p(y)) \to ((q(x) \to \exists y.p(y)) \to ((p(x) \vee q(x)) \to \exists y.p(y)))$	(Prop)
13. $(q(x) \to \exists y.p(y)) \to ((p(x) \vee q(x)) \to \exists y.p(y))$	MP(12,4)
14. $(p(x) \vee q(x)) \to \exists y.p(y)$	MP(13,11)
15. $\exists y.p(y)$	MP(14,3)

Step 9 again uses the propositional tautology $(p \to q) \to ((q \to r) \to (p \to r))$, this time with the mapping $p \mapsto q(x)$, $q \mapsto p(f(x))$ and $r \mapsto \exists y.p(y)$. Step 12 uses the propositional tautology $(p \to r) \to ((q \to r) \to ((p \vee q) \to r))$, also known as *reasoning by cases*, with the mapping $p \mapsto p(x)$, $q \mapsto q(x)$ and $r \mapsto \exists y.p(y)$.

Observe how this proof has the same structure as the semantic proof given in p. 122: we use the first hypothesis to deduce that either $p(x)$ or $q(x)$ must hold, then prove that in either case we can derive $\exists y.p(y)$, thus establishing that $\exists y.p(y)$ always holds. ◁

Another interesting result is the following, which we state as a lemma.

Lemma 37 (Change of variable)**.** Assume that $\vdash_L \forall x\varphi$, and let y be a variable not occurring in φ. Then $\vdash_L \forall y(\varphi[x/y])$.

Proof. The following derivation establishes the thesis.

1. $(\forall x\varphi) \to \varphi[x/y]$	(Ax.5)
2. $\forall y.((\forall x\varphi) \to \varphi[x/y])$	Gen(1)
3. $(\forall y.((\forall x\varphi) \to \varphi[x/y])) \to ((\forall x\varphi) \to (\forall y\varphi[x/y]))$	(Ax.4)
4. $(\forall x\varphi) \to (\forall y\varphi[x/y])$	MP(3,2)

□

The Hilbert calculus for first-order logic also enjoys a deduction theorem, and as in the case of modal logic this result has a restriction on the use of generalization. The notion of "formula depending on a hypothesis" was already introduced in the context of modal logic.

Theorem 30 (Deduction Theorem)**.** Suppose that there exists a derivation showing that $\Gamma \cup \{\varphi\} \vdash_L \psi$ where (Gen) is never applied to a formula depending on a hypothesis in Γ over a variable occurring free in φ. Then $\Gamma \vdash_L \varphi \to \psi$.

Proof. By induction on the length of the proof of $\Gamma \cup \{\varphi\} \vdash_L \psi$. The cases where ψ is an instance of an axiom, a hypothesis, or derived by (MP) from previous formulas are treated as in the propositional case.

Suppose that ψ is derived by an application of (Gen) with variable x from θ and that $x \notin FV(\varphi)$. Then we can construct the following derivation.

$n.\ \varphi \to \theta$	(IH)
$n+1.\ \forall x(\varphi \to \theta)$	Gen(n)
$n+2.\ (\forall x(\varphi \to \theta)) \to (\varphi \to \forall x\theta)$	(Ax.4)
$n+3.\ \varphi \to \forall x\theta$	MP($n+1$,$n+2$)

Note that the instance of (Ax.4) in step $n+2$ satisfies the side condition for this axiom, since by hypothesis variable x does not occur free in φ. □

As an example, we show that the normality axiom for $\forall x$ is provable in this system.

Example. The following derivation shows that $\{\forall x(\varphi \to \psi), \forall x\varphi\} \vdash_L \forall x\psi$.

1. $\forall x(\varphi \to \psi)$	(Hyp)
2. $(\forall x(\varphi \to \psi)) \to (\varphi \to \psi)$	(Ax.5)
3. $\varphi \to \psi$	MP(2,1)
4. $\forall x\varphi$	(Hyp)
5. $(\forall x\varphi) \to \varphi$	(Ax.5)
6. φ	MP(5,4)
7. ψ	MP(3,6)
8. $\forall x\psi$	Gen(7)

Since $x \notin FV(\forall x\varphi)$, we can apply the Deduction Theorem to conclude that

$$\{\forall x(\varphi \to \psi)\} \vdash_L (\forall x\varphi) \to (\forall x\psi)$$

and since again $x \notin FV(\forall x(\varphi \to \psi))$ we can apply the Deduction Theorem a second time to obtain

$$\vdash_L (\forall x(\varphi \to \psi)) \to ((\forall x\varphi) \to (\forall x\psi)),$$

which is the normality axiom for $\forall x$. ◁

Example. We now prove that $\vdash_L \exists x.(p(x) \to \forall y.p(y))$. We start by showing that $\{\forall x.\neg(p(x) \to \forall y.p(y))\} \vdash_L \bot$.

1. $\forall x.\neg(p(x) \to \forall y.p(y))$	(Hyp)
2. $(\forall x.\neg(p(x) \to \forall y.p(y))) \to \neg(p(y) \to \forall y.p(y))$	(Ax.5)
3. $\neg(p(y) \to \forall y.p(y))$	MP(2,1)
4. $(\neg(p(y) \to \forall y.p(y))) \to p(y)$	(Prop)
5. $p(y)$	MP(4,3)
6. $\forall y.p(y)$	Gen(5)
7. $(\neg(p(y) \to \forall y.p(y))) \to \neg\forall y.p(y)$	(Prop)
8. $\neg\forall y.p(y)$	MP(7,3)

9. $((\forall y.p(y)) \to ((\neg\forall y.p(y)) \to \bot))$	(Prop)
10. $(\neg\forall y.p(y)) \to \bot$	MP(10,6)
11. $\bot$	MP(11,8)

Observe that y occurs both free and bound in the formulas in steps 2–4 and 7. Although this violates the variable convention, it is practical to do so: otherwise the formulas in steps 6 and 8 would have different quantifiers, and we would have to invoke Lemma 37.

Steps 4 and 7 use the propositional tautologies $\neg(p \to q) \to p$ and $\neg(p \to q) \to \neg q$, which characterize negated implications, with mapping $p \mapsto p(y)$ and $q \mapsto \forall y.p(y)$. Step 10 uses the propositional tautology $p \to (\neg p \to \bot)$, with the mapping $p \mapsto \forall y.p(y)$.

Applying the Deduction Theorem we conclude that $\vdash_L (\forall x.\neg(p(x) \to \forall y.p(y))) \to \bot$. Since this formula is propositionally equivalent to $\neg(\forall x.\neg(p(x) \to \forall y.p(y)))$, or simply $\exists x.(p(x) \to \forall y.p(y))$, we can conclude that this last formula is also derivable. ◁

The proof strategy in this example is a bit different than the semantic analysis of the same formula. This is because the Hilbert calculus does not allow us to talk about existential witnesses (the element that makes φ true when $\exists x\varphi$ holds), and as such forces us to make these proofs by contradiction.

Example. As another example, we show that $\{\exists x.p(x), \forall x.(p(x) \to p(a))\} \vdash_L p(a)$. As a first step, we derive $p(a)$ from an extended set of hypotheses that also includes $\neg p(a)$.

1. $\neg p(a)$	(Hyp)
2. $\forall x.(p(x) \to p(a))$	(Hyp)
3. $(\forall x.(p(x) \to p(a))) \to (p(x) \to p(a))$	(Ax.5)
4. $p(x) \to p(a)$	MP(3,2)
5. $(p(x) \to p(a)) \to (\neg p(a) \to \neg p(x))$	(Prop)
6. $\neg p(a) \to \neg p(x)$	MP(5,4)
7. $\neg p(x)$	MP(6,1)
8. $\forall x.\neg p(x)$	Gen(7)
9. $\exists x.p(x)$	(Hyp)
10. $(\forall x.\neg p(x)) \to ((\exists x.p(x)) \to p(a))$	(Prop)
11. $(\exists x.p(x)) \to p(a)$	MP(10,8)
12. $p(a)$	MP(11,9)

In step 5 we used the propositional tautology $(p \to q) \to (\neg q \to \neg p)$ with the mapping $p \mapsto p(x)$, $q \mapsto p(a)$, and in step 9 we used the propositional tautology $p \to (\neg p \to q)$ with the mapping $p \mapsto \forall x.\neg p(x)$, $q \mapsto p(a)$. (Note that $\exists x.p(x)$ stands for $\neg\forall x.\neg p(x)$.)

Applying the Deduction Theorem, we conclude that $\{\exists x.p(x), \forall x.(p(x) \to p(a))\} \vdash_L \neg p(a) \to p(a)$. From the propositional tautology $(\neg p \to p) \to p$, we can use this result to conclude that $\{\exists x.p(x), \forall x.(p(x) \to p(a))\} \vdash_L p(a)$. ◁

We conclude this section with another useful theorem. The main argument in its proof is similar to that in the proof of Theorem 29; however, there are some technical subtleties arising from the side conditions in axioms (Ax.4) and (Ax.5).

Theorem 31 (Replacement Theorem)**.** Suppose that $\Gamma \vdash_L \varphi$, let $t_1, \ldots, t_n$ be any terms such that $t_i \triangleright x_i : \Gamma \cup \{\varphi\}$, and assume that in the derivation of $\Gamma \vdash_L \varphi$ there are no applications of rule (Gen) that generalize some x_i. Then $\Gamma[x_1/t_1, \ldots, x_n/t_n] \vdash_L \varphi[x_1/t_1, \ldots, x_n/t_n]$.

Proof (sketch). The proof strategy is similar to the previous case – we can uniformly replace each x_i with t_i in the derivation of $\Gamma \vdash_L \varphi$ to obtain a derivation of $\Gamma[x_1/t_1, \ldots, x_n/t_n] \vdash_L \varphi[x_1/t_1, \ldots, x_n/t_n]$.

However, some care must be taken because of the side conditions in (Ax.4) and (Ax.5). So we first prove that we can find a derivation of $\Gamma \vdash_L \varphi$ such that $t_i \triangleright x_i : \psi$ for all $1 \leq i \leq n$ and each ψ occurring in the derivation.

For (Ax.4), we note that if $(\forall x(\gamma \to \psi)) \to (\gamma \to \forall x\psi)$ is a valid instance of this axiom, then for any $y \notin FV(\gamma \to \psi)$ the formula $(\forall y(\gamma \to \psi[x/y])) \to (\gamma \to \forall y(\psi[x/y]))$ is also a valid instance of this axiom. Likewise, for (Ax.5), if $(\forall x\psi) \to \psi[x/t]$ is a valid instance of this axiom and $y \notin FV(\psi)$, then $(\forall y(\psi[x/y])) \to \psi[x/t]$ is also a valid instance of this axiom.

Therefore, we can transform the derivation as follows: we replace the first instance ψ of (Ax.4) or (Ax.5) that does not satisfy $t_i \triangleright x_i : \psi$ for all $1 \leq i \leq n$ with one that does, choosing a new variable y that does not appear elsewhere in the derivation. We then replace x by y wherever needed in order to make the derivation valid (i.e., we recursively change every formula that is obtained by applying inference rules from ψ). This may require changing other formulas higher up in the derivation, for example if ψ is an instance $(\forall x.\gamma) \to \gamma[x/t]$ of (Ax.5) that is combined by (MP) with $(\forall x.\gamma)$. The only problematic case is when such a formula is a hypothesis, and then we use Lemma 37 to derive the correct variant of the formula; the side condition on the use of (Gen) guarantees that we do not need to change variables that occur free in Γ. When this recursive process finishes, we continue to the next instance of (Ax.4) or (Ax.5), until the whole derivation is as required. □

Again, the formal proof of this result would proceed by induction on the derivation of $\Gamma \vdash_L \varphi$.

Exercise 31. Write down a precise description (using induction) of the construction in the proof of Theorem 31. The key ingredient is defining the set of formulas that need to be edited.

Exercise 32. Show that:

(a) $\{r(b)\} \vdash_L \exists z.r(z)$

(b) $\{\forall x.(p(x) \to p(f(x))), \exists y.p(y)\} \vdash_L \exists x.p(f(f(x)))$

(c) $\{p(a), \forall x.(\exists y.p(x) \wedge q(x,y)) \to r(x)\} \vdash_L q(a,b) \to r(a)$

4.4.2 Theories

We have said earlier that the most important problem in logic is that of entailment. In first-order logic, this is also the case; and in many practical domains we are interested in working with a fixed set of hypotheses Γ that characterize the set of models we want to consider.

Definition. Let Γ be a set of first-order formulas. We write $\Gamma^{\vdash_L}$ to denote the set of formulas that are derivable from Γ using the Hilbert calculus.

By soundness of the Hilbert calculus, $\Gamma^{\vdash_L} \subseteq \Gamma^{\models}$.

Definition. A *first-order theory* is a set of first-order formulas Γ such that $\Gamma^{\vdash_L} = \Gamma$.

A set of formulas $\Delta \subseteq \Gamma$ such that $\Delta^{\vdash_L} = \Gamma$ is called a *presentation* or an *axiomatization* of the theory Γ.

When working in the context of a theory presented by a set Δ, it is customary to write $\vdash_\Delta \varphi$ for $\Delta \vdash_L \varphi$ and $\Gamma \vdash_\Delta \varphi$ for $\Gamma \cup \Delta \vdash_L \varphi$. This notation is useful to distinguish the axioms of the theory from additional hypotheses that may be present in particular entailments.

Example M.9. The theory of monoids is presented by the following set Δ_M.

$$
\begin{array}{lll}
\Delta_M = \{\forall x.(x \cdot \star \mathsf{\ eq\ } x) & \text{(M1)} & \text{(identity on the right)} \\
\quad \forall x.(\star \cdot x \mathsf{\ eq\ } x) & \text{(M2)} & \text{(identity on the left)} \\
\quad \forall xyz.(x \cdot (y \cdot z) \mathsf{\ eq\ } (x \cdot y) \cdot z) & \text{(M3)} & \text{(associativity of } \cdot) \\
\quad \forall x.(x \mathsf{\ eq\ } x) & \text{(M4)} & \text{(reflexivity of } \mathsf{eq}) \\
\quad \forall xy.(x \mathsf{\ eq\ } y \to y \mathsf{\ eq\ } x) & \text{(M5)} & \text{(symmetry of } \mathsf{eq}) \\
\quad \forall xyz.(x \mathsf{\ eq\ } y \to (y \mathsf{\ eq\ } z \to x \mathsf{\ eq\ } z)) & \text{(M6)} & \text{(transitivity of } \mathsf{eq}) \\
\quad \forall xx'yy'.(x \mathsf{\ eq\ } x' \to (y \mathsf{\ eq\ } y' \to (x \cdot y \mathsf{\ eq\ } x' \cdot y'))) & \text{(M7)} & \text{(compatibility)} \\
\quad \} & &
\end{array}
$$

The first three formulas axiomatize the properties of the monoid operation, while the next three require eq to be interpreted as an equivalence relation.[3] The last axiom ensures that equality behaves well with respect to the monoid operation; the relevance of this axiom will be shown below. ◁

Exercise 33. Let $\langle M, \times_M, 1_M \rangle$ be a monoid, i.e., M is a set, $\times_M$ is an associative operation over M, and $1_M \in M$ satisfies $m \times_M 1_M = 1_M \times m = m$ for any $m \in M$. Show that the interpretation $I(M)$ defined by $D_{I(M)} = M$, $\star^{I(M)} = 1_M$, $\cdot^{I(M)} = \times_M$ and $\mathsf{eq}^{I(M)} = \{\langle m, m \rangle \mid m \in M\}$ is a model of Δ_M.

However, it is not the case that for any interpretation I with domain D such that $I \models \Delta_M$ the triple $\langle D, \cdot^I, \star^I \rangle$ is a monoid – this is because of the freedom of how equality is interpreted.

Exercise 34. Let I be an interpretation with domain D such that $\mathsf{eq}^I = \{\langle d, d \rangle \mid d \in D\}$. Show that: if $I \models \Delta_M$, then $\langle D, \cdot^I, \star^I \rangle$ is a monoid.

---

[3] In first-order logic, it is not possible to axiomatize a predicate symbol so that it is necessarily interpreted as equality (this is a consequence of the Downwards Skolem–Löwenheim Theorem, which we will see later). There is an extension of first-order logic, called *first-order logic with equality*, that adds a distinguished predicate symbol to the language which must always be interpreted as the equality relation over the domain of any interpretation.

In general, in order to obtain a monoid from an arbitrary model of Δ_M, we need to choose a set M that makes it possible to interpret eq as equality on M. This is done via a general construction, known as *quotient by an equivalence relation*: we identify all elements that are related by eq^I.

Lemma 38. Let I be an interpretation with domain D such that $I \models \Delta_M$. Then $\langle M, \times_M, 1_M\rangle$ is a monoid, where:

- $M = \{[d] \mid d \in D\}$, where $[d] = \{d' \in D \mid \langle d, d'\rangle \in \mathsf{eq}^I\}$;
- $1_M = [\star^I]$;
- $\times_M = \lambda[d][d'].[d \cdot^I d']$.

Proof. First, we show that, for any $d, d' \in D$, $[d] = [d']$ iff $\langle d, d'\rangle \in \mathsf{eq}^I$ – we use this fact repeatedly in the proof below.

Assume that $[d] = [d']$, i.e., that $\langle d, d''\rangle \in \mathsf{eq}^I$ iff $\langle d', d''\rangle \in \mathsf{eq}^I$. To show that $\langle d, d'\rangle \in \mathsf{eq}^I$, note that $I \models \forall x.(x\ \mathsf{eq}\ x)$ (M4), i.e., $I \models_\sigma x\ \mathsf{eq}\ x$ for any assignment σ.[4] This is equivalent to $\langle \sigma(x), \sigma(x)\rangle \in \mathsf{eq}^I$, and since $\sigma(x)$ is arbitrary we conclude that $\langle d, d\rangle, \langle d', d'\rangle \in \mathsf{eq}^I$. By taking $d'' = d'$, we conclude that $\langle d, d'\rangle \in \mathsf{eq}^I$.

Conversely, suppose that $\langle d, d'\rangle \in \mathsf{eq}^I$. Assume first that $\langle d', d''\rangle \in \mathsf{eq}^I$. Since $I \models \forall xyz.(x\ \mathsf{eq}\ y \to (y\ \mathsf{eq}\ z \to x\ \mathsf{eq}\ z))$ (M6), we know that $I \models_\sigma x\ \mathsf{eq}\ y \to (y\ \mathsf{eq}\ z \to x\ \mathsf{eq}\ z)$ for every assignment σ, that is, it is always the case that one of $\langle \sigma(x), \sigma(y)\rangle \notin \mathsf{eq}^I$, $\langle \sigma(y), \sigma(z)\rangle \notin \mathsf{eq}^I$ or $\langle \sigma(x), \sigma(z)\rangle \in \mathsf{eq}^I$. Taking $\sigma(x) = d$, $\sigma(y) = d'$ and $\sigma(z) = d''$, the first two conditions are not met, so we conclude that $\langle d, d''\rangle \in \mathsf{eq}^I$.

Finally, if $\langle d'', d'\rangle \in \mathsf{eq}^I$, we use the fact that $I \models \forall xy.(x\ \mathsf{eq}\ y \to y\ \mathsf{eq}\ x)$ (M5) and observe that this is equivalent to $I \models_\sigma x\ \mathsf{eq}\ y \to y\ \mathsf{eq}\ x$ for every assignment σ, i.e., if $\langle \sigma(x), \sigma(y)\rangle \in \mathsf{eq}^I$, then also $\langle \sigma(y), \sigma(x)\rangle \in \mathsf{eq}^I$. In particular, taking $\sigma(x) = d''$ and $\sigma(y) = d'$ we conclude that $\langle d', d''\rangle \in \mathsf{eq}^I$, and we can repeat the reasoning above to show that $\langle d, d''\rangle \in \mathsf{eq}^I$.

The next step is showing that $\times_M$ is well-defined: since $\langle d_1, d_2\rangle \in \mathsf{eq}^I$ implies that $[d_1] = [d_2]$, we need to show that the result of $\times_M$ is the same regardless of which particular element from $[d]$ and $[d']$ is chosen. This is a consequence of the fact that I must satisfy the compatibility axiom (M7).

We now move on to the monoid properties. To show that 1_M is an identity for $\times_M$, we observe that $I \models \forall x.(x \cdot \star\ \mathsf{eq}\ x)$ (M1) iff $I \models_\sigma x \cdot \star\ \mathsf{eq}\ x$ for any σ, which holds iff $\left\langle [\![x \cdot \star]\!]^I_\sigma, [\![x]\!]^I_\sigma \right\rangle \in \mathsf{eq}^I$. Since $[\![x \cdot \star]\!]^I_\sigma = \sigma(x) \times_M 1_M$ and $[\![x]\!]^I_\sigma = \sigma(x)$, we conclude that $\left\langle d \cdot^I \star^I, d\right\rangle \in \mathsf{eq}^I$ for every $d \in D$. Let $m \in M$; then $m = [d]$ for some $d \in D$, and therefore $m \times_M 1_M = [d \cdot^I \star^I] = [d] = m$. The other equality is proved similarly using (M2).

For associativity, we again observe that $I \models \forall xyz.(x \cdot (y \cdot z)\ \mathsf{eq}\ (x \cdot y) \cdot z)$ (M3) iff, for every σ, $I \models_\sigma x \cdot (y \cdot z)\ \mathsf{eq}\ (x \cdot y) \cdot z$, which is equivalent to $\left\langle [\![x \cdot (y \cdot z)]\!]^I_\sigma, [\![(x \cdot y) \cdot z]\!]^{I_M}_\sigma \right\rangle \in \mathsf{eq}^I$. Since $[\![x \cdot (y \cdot z)]\!]^I_\sigma = \sigma(x) \cdot^I \big(\sigma(y) \cdot^I \sigma(z)\big)$ and $[\![(x \cdot y) \cdot z]\!]^I_\sigma = \big(\sigma(x) \cdot^I \sigma(y)\big) \cdot^I \sigma(z)$, we conclude that, for any $d_1, d_2, d_3 \in D$, it is the case that $\left\langle d_1 \cdot^I \big(d_2 \cdot^I d_3\big), \big(d_1 \cdot^I d_2\big) \cdot^I d_3\right\rangle \in \mathsf{eq}^I$. Let $m_1, m_2, m_3 \in M$; then there exist $d_1, d_2, d_3 \in D$ such that $m_i = [d_i]$ for $i = 1, 2, 3$; then

$$\begin{aligned} m_1 \times_M (m_2 \times_M m_3) &= [d_1] \times_M ([d_2] \times_M [d_3]) = \big[d_1 \cdot^I \big(d_2 \cdot^I d_3\big)\big] \\ &= \big[\big(d_1 \cdot^I d_2\big) \cdot^I d_3\big] = ([d_1] \times_M [d_2]) \times_M [d_3] = (m_1 \times_M m_2) \times_M m_3\,. \end{aligned}$$

Therefore $\langle M, \times_M, \star_M\rangle$ is a monoid. □

[4] The semantics gives us directly that $I \models_\sigma x\ \mathsf{eq}\ x$ for any $\sigma \equiv_x \rho$, where ρ is arbitrary. Since any σ is x-equivalent to some assignment ρ, this is equivalent to the formulation in the text.

We now show some examples of derivations in the theory of monoids. We refer to steps that use hypotheses in Δ_M by the labels (M1)–(M7).

Example M.10. A well-known property of monoids is that the identity element is unique, which can be expressed by the formula $\forall u.((\forall x.(x \cdot u \ \mathsf{eq}\ x)) \to u \ \mathsf{eq}\ \star)$. We show that this formula is derivable in the theory of monoids.

As a first step, we show that $\{\forall x.(x \cdot u \ \mathsf{eq}\ x)\} \vdash_{\Delta_M} u \ \mathsf{eq}\ \star$. We recall the informal mathematical proof: if $\forall x.(x \cdot u \ \mathsf{eq}\ x)$ holds, then in particular $\star \cdot u \ \mathsf{eq}\ \star$; but from (M2) we also know that $\star \cdot u \ \mathsf{eq}\ u$, hence it follows that $u = \star$.

1. $\forall x.(x \cdot u \ \mathsf{eq}\ x)$	(Hyp)
2. $(\forall x.(x \cdot u \ \mathsf{eq}\ x)) \to (\star \cdot u \ \mathsf{eq}\ \star)$	(Ax.5)
3. $\star \cdot u \ \mathsf{eq}\ \star$	MP(2,1)
4. $\forall x.(\star \cdot x \ \mathsf{eq}\ x)$	(M2)
5. $(\forall x.(\star \cdot x \ \mathsf{eq}\ x)) \to (\star \cdot u \ \mathsf{eq}\ u)$	(Ax.5)
6. $\star \cdot u \ \mathsf{eq}\ u$	MP(5,4)
7. $\forall xy.(x \ \mathsf{eq}\ y \to y \ \mathsf{eq}\ x)$	(M5)
8. $(\forall xy.(x \ \mathsf{eq}\ y \to y \ \mathsf{eq}\ x)) \to (\forall y.(\star \cdot u \ \mathsf{eq}\ y \to y \ \mathsf{eq}\ \star \cdot u))$	(Ax.5)
9. $\forall y.(\star \cdot u \ \mathsf{eq}\ y \to y \ \mathsf{eq}\ \star \cdot u)$	MP(8,7)
10. $(\forall y.(\star \cdot u \ \mathsf{eq}\ y \to y \ \mathsf{eq}\ \star \cdot u)) \to (\star \cdot u \ \mathsf{eq}\ u \to u \ \mathsf{eq}\ \star \cdot u)$	(Ax.5)
11. $\star \cdot u \ \mathsf{eq}\ u \to u \ \mathsf{eq}\ \star \cdot u$	MP(10,9)
12. $u \ \mathsf{eq}\ \star \cdot u$	MP(11,6)
13. $\forall xyz.(x \ \mathsf{eq}\ y \to (y \ \mathsf{eq}\ z \to x \ \mathsf{eq}\ z))$	(M6)
14. $(\forall xyz.(x \ \mathsf{eq}\ y \to (y \ \mathsf{eq}\ z \to x \ \mathsf{eq}\ z))) \to (\forall yz.(u \ \mathsf{eq}\ y \to (y \ \mathsf{eq}\ z \to u \ \mathsf{eq}\ z)))$	(Ax.5)
15. $\forall yz.(u \ \mathsf{eq}\ y \to (y \ \mathsf{eq}\ z \to u \ \mathsf{eq}\ z))$	MP(14,13)
16. $(\forall yz.(u \ \mathsf{eq}\ y \to (y \ \mathsf{eq}\ z \to u \ \mathsf{eq}\ z))) \to (\forall z.(u \ \mathsf{eq}\ \star \cdot u \to (\star \cdot u \ \mathsf{eq}\ z \to u \ \mathsf{eq}\ z)))$	(Ax.5)
17. $\forall z.(u \ \mathsf{eq}\ \star \cdot u \to (\star \cdot u \ \mathsf{eq}\ z \to u \ \mathsf{eq}\ z))$	MP(16,15)
18. $(\forall z.(u \ \mathsf{eq}\ \star \cdot u \to (\star \cdot u \ \mathsf{eq}\ z \to u \ \mathsf{eq}\ z))) \to (u \ \mathsf{eq}\ \star \cdot u \to (\star \cdot u \ \mathsf{eq}\ \star \to u \ \mathsf{eq}\ \star)))$	(Ax.5)
19. $u \ \mathsf{eq}\ \star \cdot u \to (\star \cdot u \ \mathsf{eq}\ \star \to u \ \mathsf{eq}\ \star)$	MP(18,17)
20. $\star \cdot u \ \mathsf{eq}\ \star \to u \ \mathsf{eq}\ \star$	MP(19,12)
21. $u \ \mathsf{eq}\ \star$	MP(20,3)

Compare the structure of this proof with the informal proof given above. Steps 1–3 establish that $\star \cdot u \ \mathsf{eq}\ \star$; steps 4–6 establish that $\star \cdot u \ \mathsf{eq}\ u$; and the remainder of the proof concerns manipulating equalities, first using symmetry to derive $u \ \mathsf{eq}\ \star \cdot u$ (steps 7–12), and then transitivity to yield the thesis. The length of the proof is a consequence of working with low-level axioms – instantiating the transitivity axiom (M6), for example, consumes 6 steps of the proof.

Applying the Deduction Theorem, we conclude that $\vdash_{\Delta_M} (\forall x.(x \cdot u \ \mathsf{eq}\ x)) \to u \ \mathsf{eq}\ \star$. Finally, by applying (Gen), we obtain that $\vdash_{\Delta_M} \forall u.((\forall x.(x \cdot u \ \mathsf{eq}\ x)) \to u \ \mathsf{eq}\ \star)$.

As a consequence of soundness of the Hilbert calculus for first-order logic, we conclude that the corresponding semantic property holds in all monoids. ◁

Example M.11. As a lengthy second example, we prove another known property of monoids. An element y is called a *left inverse* of x if $y \cdot x \ \mathsf{eq}\ \star$, and a *right inverse* if $x \cdot y \ \mathsf{eq}\ \star$. If x has

both a left inverse and a right inverse, then they must coincide: $\forall xuv.(u \cdot x \;\mathsf{eq}\; \star \to (x \cdot v \;\mathsf{eq}\; \star \to u \;\mathsf{eq}\; v))$. We show that this formula is also derivable in the theory of monoids.

Recall the informal proof of this result. Evaluating the product $u \cdot (x \cdot v)$ gives $u \cdot (x \cdot v) = u \cdot \star = u$; but by associativity we also have $u \cdot (x \cdot v) = (u \cdot x) \cdot v = \star \cdot v = v$. Therefore $u = v$.

The first step is showing that $\{u \cdot x \;\mathsf{eq}\; \star, x \cdot v \;\mathsf{eq}\; \star\} \vdash_{\Delta_M} u \;\mathsf{eq}\; v$. We construct the derivation in chunks, following this proof closely; we replace the steps instantiating axioms (M3), (M5), (M6) and (M7) – the sequences in steps 7–11 and 13–19 in the previous proof – with vertical dots.

We start by establishing that $u \cdot (x \cdot v) = u$.

1. $\forall x.(x \;\mathsf{eq}\; x)$	(M4)
2. $(\forall x.(x \;\mathsf{eq}\; x)) \to (u \;\mathsf{eq}\; u)$	(Ax.5)
3. $u \;\mathsf{eq}\; u$	MP(2,1)
4. $x \cdot v \;\mathsf{eq}\; \star$	(Hyp)
5. $\forall xx'yy'(x \;\mathsf{eq}\; x' \to (y \;\mathsf{eq}\; y' \to (x \cdot y \;\mathsf{eq}\; x' \cdot y')))$	(M7)
$\vdots$	(Ax.5) and MP
13. $u \;\mathsf{eq}\; u \to (x \cdot v \;\mathsf{eq}\; \star \to (u \cdot (x \cdot v) \;\mathsf{eq}\; u \cdot \star))$	MP(12,11)
14. $x \cdot v \;\mathsf{eq}\; \star \to (u \cdot (x \cdot v) \;\mathsf{eq}\; (u \cdot \star))$	MP(13,3)
15. $u \cdot (x \cdot v) \;\mathsf{eq}\; (u \cdot \star)$	MP(14,4)
16. $\forall x.(x \cdot \star \;\mathsf{eq}\; x)$	(M1)
17. $(\forall x.(x \cdot \star \;\mathsf{eq}\; x)) \to (u \cdot \star \;\mathsf{eq}\; u)$	(Ax.5)
18. $u \cdot \star \;\mathsf{eq}\; u$	MP(17,16)
19. $\forall xyz.(x \;\mathsf{eq}\; y \to (y \;\mathsf{eq}\; z \to x \;\mathsf{eq}\; z))$	(M6)
$\vdots$	(Ax.5) and MP
25. $(u \cdot (x \cdot v) \;\mathsf{eq}\; (u \cdot \star)) \to ((u \cdot \star \;\mathsf{eq}\; u) \to ((u \cdot (x \cdot v)) \;\mathsf{eq}\; u))$	MP(24,23)
26. $(u \cdot \star \;\mathsf{eq}\; u) \to ((u \cdot (x \cdot v)) \;\mathsf{eq}\; u)$	MP(25,15)
27. $(u \cdot (x \cdot v)) \;\mathsf{eq}\; u$	MP(26,18)

Similarly, we prove that $(u \cdot x) \cdot v = v$.

28. $\forall x.(x \;\mathsf{eq}\; x)$	(M4)
29. $(\forall x.(x \;\mathsf{eq}\; x)) \to (v \;\mathsf{eq}\; v)$	(Ax.5)
30. $v \;\mathsf{eq}\; v$	MP(29,28)
31. $u \cdot x \;\mathsf{eq}\; \star$	(Hyp)
32. $\forall xx'yy'(x \;\mathsf{eq}\; x' \to (y \;\mathsf{eq}\; y' \to (x \cdot y \;\mathsf{eq}\; x' \cdot y')))$	(M7)
$\vdots$	(Ax.5) and MP
40. $((u \cdot x) \;\mathsf{eq}\; \star) \to ((v \;\mathsf{eq}\; v) \to ((u \cdot x) \cdot v) \;\mathsf{eq}\; (\star \cdot v))$	MP(39,38)
41. $v \;\mathsf{eq}\; v \to ((u \cdot x) \cdot v) \;\mathsf{eq}\; (\star \cdot v))$	MP(40,31)
42. $((u \cdot x) \cdot v) \;\mathsf{eq}\; (\star \cdot v)$	MP(41,30)

43. $\forall x.(\star \cdot x \mathsf{\ eq\ } x)$	(M2)
44. $(\forall x.(\star \cdot x \mathsf{\ eq\ } x)) \to (\star \cdot v \mathsf{\ eq\ } v)$	(Ax.5)
45. $\star \cdot v \mathsf{\ eq\ } v$	MP(44,43)
46. $\forall xyz.(x \mathsf{\ eq\ } y \to (y \mathsf{\ eq\ } z \to x \mathsf{\ eq\ } z))$	(M6)
$\vdots$	(Ax.5) and MP
52. $((u \cdot x) \cdot v \mathsf{\ eq\ } (\star \cdot v)) \to ((\star \cdot v \mathsf{\ eq\ } v) \to ((u \cdot x) \cdot v \mathsf{\ eq\ } v))$	MP(51,50)
53. $(\star \cdot v \mathsf{\ eq\ } v) \to (u \cdot x) \cdot v \mathsf{\ eq\ } v$	MP(52,42)
54. $(u \cdot x) \cdot v \mathsf{\ eq\ } v$	MP(53,45)

Finally, we use associativity of $\cdot$ and properties of equality to derive the conclusion.

55. $\forall xyz.(x \cdot (y \cdot z) \mathsf{\ eq\ } (x \cdot y) \cdot z)$	(M3)
$\vdots$	(Ax.5) and MP
61. $u \cdot (x \cdot v) \mathsf{\ eq\ } (u \cdot x) \cdot v$	MP(60,59)
62. $\forall xy.(x \mathsf{\ eq\ } y \to y \mathsf{\ eq\ } x)$	(M5)
$\vdots$	(Ax.5) and MP
66. $(u \cdot (x \cdot v) \mathsf{\ eq\ } u) \to (u \mathsf{\ eq\ } u \cdot (x \cdot v))$	MP(65,64)
67. $u \mathsf{\ eq\ } u \cdot (x \cdot v)$	MP(66,27)
68. $\forall xyz.(x \mathsf{\ eq\ } y \to (y \mathsf{\ eq\ } z \to x \mathsf{\ eq\ } z))$	(M6)
$\vdots$	(Ax.5) and MP
74. $(u \mathsf{\ eq\ } u \cdot (x \cdot v)) \to ((u \cdot (x \cdot v) \mathsf{\ eq\ } (u \cdot x) \cdot v) \to (u \mathsf{\ eq\ } (u \cdot x) \cdot v))$	MP(73,72)
75. $(u \cdot (x \cdot v) \mathsf{\ eq\ } (u \cdot x) \cdot v) \to (u \mathsf{\ eq\ } (u \cdot x) \cdot v)$	MP(74,67)
76. $u \mathsf{\ eq\ } (u \cdot x) \cdot v$	MP(75,61)
77. $\forall xyz.(x \mathsf{\ eq\ } y \to (y \mathsf{\ eq\ } z \to x \mathsf{\ eq\ } z))$	(M6)
$\vdots$	(Ax.5) and MP
83. $(u \mathsf{\ eq\ } (u \cdot x) \cdot v) \to (((u \cdot x) \cdot v \mathsf{\ eq\ } v) \to u \mathsf{\ eq\ } v)$	MP(82,81)
84. $((u \cdot x) \cdot v \mathsf{\ eq\ } v) \to u \mathsf{\ eq\ } v$	MP(83,76)
85. $u \mathsf{\ eq\ } v$	MP(84,54)

We now apply the Deduction Theorem twice to obtain $\vdash_{\Delta_M} (u \cdot x \mathsf{\ eq\ } \star) \to ((x \cdot v \mathsf{\ eq\ } \star) \to u \mathsf{\ eq\ } v)$. Finally, by applying (Gen) three times, we conclude that $\vdash_{\Delta_M} \forall xuv.(u \cdot x \mathsf{\ eq\ } \star \to (x \cdot v \mathsf{\ eq\ } \star \to u \mathsf{\ eq\ } v))$. ◁

Exercise 35. Prove that the following are derivable in the theory of monoids.

(a) $\vdash_{\Delta_M} \forall x.(x \cdot \star \mathsf{\ eq\ } \star \cdot x)$

(b) $\vdash_{\Delta_M} \forall xy.((x \cdot \star) \cdot (\star \cdot y) \mathsf{\ eq\ } x \cdot y)$

(c) $\{x \cdot x \mathsf{\ eq\ } y\} \vdash_{\Delta_M} x \cdot y \mathsf{\ eq\ } y \cdot x$

(d) $\{x \cdot y \mathsf{\ eq\ } \star\} \vdash_{\Delta_M} \forall w \exists z.(x \cdot z \mathsf{\ eq\ } w)$

For each property, start by sketching a mathematical proof, and then expand it into a formal proof in the Hilbert calculus.

When working with theories, it is useful to have as few axioms as possible. This means that it should not be possible to prove any axiom from the remaining ones.

Definition. A formula φ is said to be *independent* from Γ if φ cannot be proved from Γ. A set of formulas Γ is said to be independent if γ is independent from $\Gamma \setminus \{\gamma\}$, for each $\gamma \in \Gamma$.

Example M.12. We show that axiom (M1) is independent from the remaining axioms in the theory of monoids. In order to do this, we exhibit a model that satisfies axioms (M2–7), but not (M1).

This model is a variation of the interpretation I_M from Example M.3. We keep the same domain $D_M = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}^*$ and the interpretation of $\star$ and eq, but redefine $\cdot^{I_M}$ such that $\omega \cdot^{I_M} \omega' = \omega'$.

Since we did not change the interpretation of eq, axioms (M4–7) are guaranteed to hold. For axioms (M1–3), we have that:

- $I_M \models \forall x.(x \cdot \star \mathsf{eq}\ x)$ iff $\rho(x) \cdot^{I_M} \star^{I_M} = \rho(x)$ for any assignment ρ, which fails if $\rho(x) \neq \epsilon$ since $\rho(x) \cdot^{I_M} \star^{I_M} = \star^{I_M} = \epsilon$.
- $I_M \models \forall x.(\star \cdot x\ \mathsf{eq}\ x)$ iff $\star^{I_M} \cdot^{I_M} \rho(x) = \rho(x)$ for any assignment ρ, which is true by definition of $\cdot^{I_M}$.
- $I_M \models \forall xyz.(x \cdot (y \cdot z)\ \mathsf{eq}\ (x \cdot y) \cdot z)$ iff $\rho(x) \cdot^{I_M} \big(\rho(y) \cdot^{I_M} \rho(z)\big) = \big(\rho(x) \cdot^{I_M} \rho(y)\big) \cdot^{I_M} \rho(z)$ for any assignment ρ. By definition of $\cdot^{I_M}$, we have that

$$\rho(x) \cdot^{I_M} \big(\rho(y) \cdot^{I_M} \rho(z)\big) = \rho(x)$$
$$\big(\rho(x) \cdot^{I_M} \rho(y)\big) \cdot^{I_M} \rho(z) = \rho(x) \cdot^{I_M} \rho(z) = \rho(x)$$

  and thus this axiom is also valid.

Therefore this interpretation makes all axioms true except for (M1). By soundness of the Hilbert calculus, this implies that (M1) is not derivable from (M2–7). ◁

Exercise 36. Prove that axiom (M2) is independent from the remaining axioms in the theory of monoids.

Exercise 37. A *commutative* monoid is a monoid that also satisfies the axiom $\forall xy.(x \cdot y\ \mathsf{eq}\ y \cdot x)$ (MC). The theory of commutative monoids includes axioms (M1) and (M3–7) from the theory of monoids, together with axiom (MC).

(a) Prove that axiom (M2) is derivable in the theory of commutative monoids.

(b) Prove that axiom (MC) is not derivable in the theory of monoids.

Exercise 38. An *idempotent* monoid is a monoid that also satisfies the axiom $\forall x.(x \cdot x\ \mathsf{eq}\ x)$ (MI). Show that the theory of idempotent monoids is strictly stronger than the theory of monoids by proving that axiom (MI) can not be derived in the theory of monoids.

4.4.3 Completeness

The proof of completeness of the Hilbert calculus for first-order logic follows the same idea as that for propositional logic. As in the case of the tableaux calculus, it is however made more complicated by the presence of existential quantifiers. The key ingredient of the proof is to show that every consistent set of formulas can be extended to a maximal consistent set, and use this fact to define a construction of a term model that satisfies exactly the formulas in that maximal consistent set. However, we may need to extend the signature of the logic: taking as domain the set of all closed terms (terms that do not contain variables) is in general not enough.

Example. Consider the signature Σ with $F_0 = \{a, b\}$, $P_1 = \{p\}$, and all other sets F_n and P_n empty. The set $\Gamma = \{p(a), p(b), \exists x.\neg p(x)\}$ is consistent, but it does not have a model with domain $D = \{a, b\}$ where $a^I = a$ and $b^I = b$. ◁

From this point onwards, we assume Σ to be a fixed first-order signature, and we define an extended signature Σ^+ by adding to F_0 an infinite set $\mathcal{D} = \{d_n \mid n \in \mathbb{N}\}$ of new constants.

For proving completeness, having additional constants in the signature is not a problem: all formulas that can be written over Σ are also well-formed formulas over Σ^+, and we show later that every such formula provable with a derivation using formulas over Σ^+ can also be proven using only formulas over Σ.

Let $\{\psi_n \mid n \in \mathbb{N}\}$ be an enumeration of the formulas over Σ^+ with one free variable. Letting x_n be the (only) free variable in ψ_n, we define

$$\theta_n = (\exists x_n \neg \psi_n) \to \neg \psi_n[x_n/c_n]$$

where $c_n \in \mathcal{D}$ does not occur in either $\{c_0, \ldots, c_{n-1}\}$ or $\{\psi_0, \ldots, \psi_n\}$. Note that θ_n is a closed formula for all n.

The constants c_n are known as *Skolem variables*, and the construction described above is commonly referred to as *Skolemization*. Intuitively, the formulas θ_n ensure that, if an existential formula holds, then there is a particular instance of it that can be explicitly derived. Furthermore, the constants making each formula true are all distinct.

Example. Suppose that we picked an enumeration such that $\psi_0 = p(x, x)$, $\psi_1 = \forall x.(p(d_1, x) \vee p(y, x))$ and $\psi_2 = p(a, y) \to \exists x.p(x, d_0)$.

Then $x_0 = x$. Since we can choose any element of $\mathcal{D}$ for c_0, assume that we take $c_0 = d_0$. Formula θ_0 then becomes

$$\exists x.\neg p(x, x) \to \neg p(d_0, d_0)\,.$$

Next, we have $x_1 = y$, and for c_1 we can choose any element of $\mathcal{D}$ distinct from d_0 (since this is c_0) and d_1 (which occurs in ψ_1). Taking $c_1 = d_2$, we get

$$\exists y.\neg \forall x.(p(d_1, x) \vee p(y, x)) \to \neg \forall x.(p(d_1, x) \vee p(d_2, x))$$

as θ_1.

Finally, $x_2 = y$, and for c_2 we can take any element of $\mathcal{D}$ distinct from d_0, d_1 and d_2. Taking $c_2 = d_3$, we obtain

$$\exists y.\neg(p(a, y) \to \exists x.p(x, d_0)) \to \neg(p(a, d_3) \to \exists x.p(x, d_0))$$

as θ_2. ◁

The reason for negating ψ_n is practical convenience. This is immaterial, since the formula $\neg\psi_n$ also appears in the enumeration, and thus we also get a witness for $\exists x_n.\neg\neg\psi_n$, which is equivalent to $\exists x_n\psi_n$.

Lemma 39. Let Γ be a consistent set of formulas over Σ^+ and define

$$\begin{cases}\Gamma_0 = \Gamma \\ \Gamma_{n+1} = \Gamma_n \cup \{\theta_n\}\end{cases}$$

Then $\Gamma^+ = \bigcup \Gamma_n$ is consistent.

Proof. Assume that Γ^+ is not consistent, and choose the least n such that Γ_{n+1} is inconsistent. Then $\Gamma_n \cup \{\theta_n\} \vdash_L \neg\theta_n$. Since θ_n does not have free variables, the Deduction Theorem applies, and we conclude that $\Gamma_n \vdash_L \theta_n \to \neg\theta_n$; using the propositional tautology $(p \to \neg p) \to \neg p$ with the mapping $p \mapsto \theta_n$, we can infer that $\Gamma_n \vdash_L \neg\theta_n$. Again by propositional reasoning we conclude that $\Gamma_n \vdash_L \exists x_n\neg\psi_n$ and $\Gamma_n \vdash_L \psi_n[x_n/c_n]$.

Since c_n does not occur in any formula in either Γ_n or ψ_n, we can replace it by a fresh variable y everywhere in the derivation of $\Gamma_n \vdash_L \neg\psi_n[x_n/c_n]$ to conclude that $\Gamma_n \vdash_L \psi_n[x_n/y]$. We can then build the derivation

1. $\psi_n[x/y]$	Lemma
2. $\forall y\psi_n[x/y]$	Gen(1)
3. $(\forall y\psi_n[x/y]) \to \psi_n$	Ax.5
4. ψ_n	MP(3,2)
5. $\psi_n \to \neg\neg\psi_n$	Prop
6. $\neg\neg\psi_n$	MP(5,4)
7. $\forall x_n\neg\neg\psi_n$	Gen(6)

where step 5 uses the propositional tautology $p \to \neg\neg p$ with mapping $p \mapsto \psi_n$.

But then $\Gamma_n \vdash_L \forall x_n\neg\neg\psi_n$ and $\Gamma_n \vdash_L \exists x_n\neg\psi_n$, which expands to $\Gamma_n \vdash_L \neg\forall x_n\neg\neg\psi_n$. Thus Γ_n is inconsistent, contradicting our choice of n. □

Lemma 40. Let Γ be a consistent set of closed formulas. Then Γ has a model.

Proof. Assume that Γ is a consistent set of closed formulas over a signature Σ. Construct the extended signature Σ^+ and the set Γ^+ as above. It is simple to check that Γ is also consistent when we allow derivations to use all formulas over Σ^+, and therefore Γ^+ is also consistent.

We can then extend Γ^+ to a maximal consistent set Γ^* using exactly the same construction as in the propositional case. From Γ^* we define an interpretation I as follows:

- D is the set of all closed terms over Σ^+;
- $c^I = c$ for all constants c;
- $f^I(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$ for all $f \in F_n$;
- $p^I = \{\langle t_1, \ldots, t_n\rangle \mid p(t_1, \ldots, t_n) \in \Gamma^*\}$ for all $p \in P_n$.

We first prove that $[\![t]\!]^I_\rho = t$ for every closed term t and assignment ρ. This is straightforward to show by induction: for every constant c we have $[\![c]\!]^I_\rho = c^I = c$ by definition, and for every

function symbol $f \in F_n$ we have $[\![f(t_1, \ldots, t_n)]\!]^I_\rho = f^I\left([\![t_1]\!]^I_\rho, \ldots, [\![t_n]\!]^I_\rho\right)$ which, by definition of f^I and induction hypothesis, is equal to $f(t_1, \ldots, t_n)$.

We can now prove that $I \models \varphi$ iff $\varphi \in \Gamma^*$ by structural induction on φ.

- If φ is an atomic formula $p(t_1, \ldots, t_n)$, then $I \models_\rho p(t_1, \ldots, t_n)$ iff $\left\langle [\![t_1]\!]^I_\rho, \ldots, [\![t_n]\!]^I_\rho \right\rangle \in p^I$; from the previous result, this is equivalent to $\langle t_1, \ldots, t_n \rangle \in p^I$, which holds iff $p(t_1, \ldots, t_n) \in \Gamma^*$ by construction.

- If φ is $\neg\psi$ or $\psi \to \gamma$, then the proof is similar to the propositional case, using the induction hypothesis.

- If φ is $\forall x.\psi$, then by definition $I \models_\rho \forall x\psi$ iff $I \models_\sigma \psi$ for every $\sigma \equiv_x \rho$.

  By Lemma 31, $I \models_\sigma \psi$ iff $I \models_\rho \psi[x/\sigma(x)]$: since $\sigma(x)$ is a closed term, we have both that $\sigma = \rho\left[x/[\![\sigma(x)]\!]^I_\rho\right]$ and that $\sigma(x) \triangleright x : \psi$. Thus $I \models_\rho \forall x\psi$ iff $I \models_\rho \varphi[x/t]$ for every closed term t, i.e., for every $t \in D$.

  Furthermore, for every $t \in D$, the formula $\psi[x/t]$ is closed, and therefore the induction hypothesis applies. Thus $I \models_\rho \forall x\psi$ iff $\varphi[x/t] \in \Gamma^*$ for all closed terms t.

  Since Γ^* is maximal consistent, either $\forall x\psi \in \Gamma^*$ or $\neg\forall x\psi \in \Gamma^*$. In the latter case, the equivalent formula $\exists x\neg\psi$ must also be in Γ^*, and since ψ has exactly one free variable it must be one of the ψ_n in the enumeration used for constructing $\Gamma^+ \subseteq \Gamma^*$. Therefore $\theta_n = (\exists x\neg\psi) \to \neg\psi[x/c_n] \in \Gamma^*$, and again since Γ^* is maximal consistent this implies that $\neg\psi[x/c_n] \in \Gamma^*$, which is incompatible with $\varphi[x/t] \in \Gamma^*$ for all closed terms t.

  Therefore $I \models_\rho \forall x\psi$ iff $\forall x\psi \in \Gamma^*$.

In particular, $I \models \Gamma$.

From I we can obtain an interpretation over Σ by keeping the domain D unchanged, but defining c^I only for constants c in Σ. Since all formulas in Γ are interpreted in the same way as before, I is a model of Γ (over Σ). □

Exercise 39. Let Γ be a consistent set of formulas over a signature Σ, and Σ' be a signature that extends Σ. Show that Γ is also consistent when we allow derivations to use all formulas over Σ'.

Theorem 32 (Gödel's Completeness Theorem)**.** If $\Gamma \models \varphi$, then $\Gamma \vdash_L \varphi$.

Proof. The proof is by contradiction. Suppose that $\Gamma \nvdash_L \varphi$. Then $\Gamma \cup \{\neg\varphi\}$ is consistent, again arguing as in the propositional case.

Assume first that Γ and φ do not have free variables. Then the previous lemma guarantees that $\Gamma \cup \{\neg\varphi\}$ has a model, whence $\Gamma \not\models \varphi$.

Suppose that Γ has free variables, but φ does not. Then $\Gamma \models \varphi$ iff $\forall\Gamma \models \varphi$, where $\forall\Gamma$ is the set obtained by universally quantifying all free variables in each formula in Γ. Applying the previous argument to $\forall\Gamma \cup \{\varphi\}$ yields the thesis.

Finally, if φ has free variables, then start by replacing them uniformly in φ and Γ with fresh constants. We are now in the previous case, and we can repeat the same argument again. □

There are two refinements of Gödel's completeness proof that have important practical consequences.

Theorem 33 (Downwards Skolem–Löwenheim Theorem)**.** Let Γ be a set of formulas. If Γ has a model, then it has a model with countable domain.

Proof. If Γ has a model, then it is consistent (otherwise, by soundness, its model would satisfy both φ and $\neg\varphi$ for some φ, which is a contradiction). Applying the construction in the proof of Lemma 40 to Γ yields a model with a countable domain. □

Therefore, it is impossible to find a set of first-order axioms that ensures e.g. that all its models are finite, or that all its models have the cardinality of the continuum.

Theorem 34 (Upwards Skolem–Löwenheim Theorem)**.** Let Γ be a set of formulas. If Γ has a model of cardinality α, then it has a model of cardinality β for all $\beta > \alpha$.

Proof. In the construction in the proof of Lemma 40, extend the domain D to include β elements, and extend the definition of p^I so that $p^I(t_1, \ldots, t_n)$ iff $p^I(t_1^\circ, \ldots, t_n^\circ)$, where t_i° is obtained from t_i by replacing every constant not in Σ^+ by c_0. It is easy to check that this extended model satisfies the same formulas as the original one. □

As a consequence, it is also impossible to find e.g. a set of formulas whose models are all isomorphic to the natural numbers.

Exercise 40. Check that the claim at the end of the proof of Theorem 34 holds.

The Upwards Skolem–Löwenheim Theorem (and its proof) apply to many other logics whose models are similar to first-order logic; in particular, it can be adapted to modal logic. The Downwards Skolem–Löwenheim is more specific to first-order logic, and has important applications in determining that some theories are not axiomatizable.

Exercise 41. Consider a first-order signature with a single constant a and a single binary predicate symbol $=$. The formula $\forall x.x = \mathsf{a}$ intuitively states that every element is equal to a. So it seems to imply that, if $I \models \forall x.x = \mathsf{a}$, then D^I should be a singleton. Why doesn't this contradict the Upwards Skolem–Löwenheim Theorem?

4.5 Peano's theory of arithmetic

In previous sections, we discussed the concept of a first-order theory: a deductively closed set of formulas that is presented by a set of axioms. In the late 19th and early 20th century, when many researchers were focused on the goal of making all of mathematics precise based on set theory and first-order logic, finitely axiomatizable theories were studied intensely. The success of these, especially in the field of algebra (where the theory of monoids is a typical example) encouraged people to try to axiomatize more and more complex domains – until several theoretical results about first-order logic revealed unsolvable limitations in this approach.

The Skolem–Löwenheim theorems that were presented at the end of the previous section actually predate Gödel's completeness theorem. The original proofs, of course, were much more complex than the one we gave, which is an adaptation of the proof of Gödel's later result. For mathematics, these results have a profound impact: they imply that neither number theory

(which deals with properties of the set of natural numbers) nor calculus (which deals with real numbers) can be properly axiomatized in first-order logic. For number theory, this follows from the Upwards Skolem–Löwenheim theorem, since any theory that has the natural numbers as a model also admits uncountable models (and the natural numbers are countable). For calculus, this follows from the Downwards Skolem–Löwenheim theorem, since any theory that has the real numbers as a model also admits countable models (and the reals are uncountable).

Today we do not necessarily see these limitations as serious problems. The realization of the existence of non-standard models of the theory of natural numbers gave rise to the field of non-standard analysis, which has the potential to change mathematics in the future. On the other hand, philosophers have argued that uncountable sets are an abstraction anyway, and that in practice we only need to reason about a countable set of real numbers – and the development of constructive mathematics and computable real analysis are two good examples of how such an argument has value. Furthermore, a theory of arithmetic had been available from the late 1880s, which by the 1920s had become firmly established (and is still used today).

The situation changed dramatically in 1931, when Gödel published his two incompleteness theorems. They showed not only that the theory of arithmetic was unable to prove all valid statements about natural numbers, but also that the problem does not lie in the particular set of axioms, but is rather a fundamental property of the theory: it does not admit any finite axiomatization. Furthermore, by expanding the argument, Gödel proved that any logic system strong enough to include arithmetic as a theory is either inconsistent or incomplete, and that, in particular, it cannot prove its own consistency (unless it is inconsistent).

At first sight, this result seems to contradict the completeness theorem for first-order logic, and indeed this is a very common misunderstanding. As we will see below, there is a subtlety in the theory of arithmetic: it is not a first-order theory, in the sense that one of its axioms cannot be written inside first-order logic. However, these results are traditionally presented together with first-order logic, and we chose to follow this tradition in these notes.

4.5.1 Peano's axioms

The theory of arithmetic has the following signature Σ_P.

$$F_0 = \{\mathsf{0}\} \qquad F_1 = \{'\} \qquad F_2 = \{+, \times\} \qquad F_n = \emptyset, \quad n \geq 3$$
$$P_2 = \{\approx\} \qquad P_n = \emptyset, \quad n \geq 3$$

In the intended semantics for this signature, the symbol $\mathsf{0}$ represents the natural number 0, and $'$ denotes the successor operation $\lambda x.x + 1$. The operations $+$ and $\times$ have the expected meaning, and $\approx$ denotes equality. As in previous sections, we write the binary function and predicate symbols inline.

We already saw that having an intended interpretation in mind does not automatically assign the meaning we want to the syntax. Therefore, the theory of arithmetic also has a number of axioms that ensure that sum, multiplication and equality behave as expected on the syntactic objects $\mathsf{0}$, $\mathsf{0}'$, $\mathsf{0}''$, etc. – which, regardless of the interpretation, are isomorphic to the natural numbers. We refer to these terms as natural numbers in the remainder of the presentation, and often write $\mathsf{1}$, $\mathsf{2}$, $\mathsf{3}$, etc. instead of $\mathsf{0}'$, $\mathsf{0}''$, $\mathsf{0}'''$, etc.

The axiomatization we present is one of the standard ones in use, and it is close to Peano's original formulation. It consists of nine schematic axioms: four dealing with equality, two with addition, two with multiplication, and the induction axiom. They are schematic axioms in the

sense that they contain free variables, which we are free to instantiate by any terms.[5] We write $\vdash_P \varphi$ to denote that φ can be proven from the Peano axioms using the Hilbert calculus for first-order logic.

The axioms dealing with equality are the following.

$$x \approx y \to (x \approx z \to y \approx z) \tag{P1}$$
$$x \approx y \to x' \approx y' \tag{P2}$$
$$\neg(\mathsf{0} \approx x') \tag{P3}$$
$$x' \approx y' \to x \approx y \tag{P4}$$

These axioms state that equality is Euclidean (P1), that successor is injective (P4), that equal numbers have equal successors (P2), and that $\mathsf{0}$ is not the successor of any number (P3). As we saw when discussing modal logic, Euclidean relations are equivalence relations over the set of elements that are related to some other element.

The next axioms recursively define addition and multiplication.

$$x + \mathsf{0} \approx x \tag{P5}$$
$$x + y' \approx (x + y)' \tag{P6}$$
$$x \times \mathsf{0} \approx \mathsf{0} \tag{P7}$$
$$x \times y' \approx x \times y + x \tag{P8}$$

These axioms allow us to compute the value of sums and products. Since axiom (P5) also puts every natural number in relation with some other number, we can use it to prove that equality is an equivalence relation.

Example. We show that $\approx$ is an equivalence relation. For reflexivity, we instantiate (P1) such that both premises are instances of (P5).

1. $x + \mathsf{0} \approx x \to (x + \mathsf{0} \approx x \to x \approx x)$	(P1)
2. $x + \mathsf{0} \approx x$	(P5)
3. $x + \mathsf{0} \approx x \to x \approx x$	MP(1,2)
4. $x \approx x$	MP(3,2)
5. $\forall x.(x \approx x)$	Gen(4)

We denote the use of reflexivity in proofs by $\mathrm{Refl}(t)$, where t is the desired instantiation of x. This can be seen as a shortcut for inserting steps 1–4 of the previous proof with x replaced by t. Using reflexivity and (P1), symmetry is now easy to derive.

1. $x \approx y \to (x \approx x \to y \approx x)$	(P1)
2. $(x \approx y \to (x \approx x \to y \approx x)) \to (x \approx x \to (x \approx y \to y \approx x))$	(Prop)
3. $x \approx x \to (x \approx y \to y \approx x)$	MP(2,1)
4. $x \approx x$	$\mathrm{Refl}(x)$
5. $x \approx y \to y \approx x$	MP(3,4)
6. $\forall y.(x \approx y \to y \approx x)$	Gen(5)
7. $\forall xy.(x \approx y \to y \approx x)$	Gen(6)

[5] Presenting these axioms as universally quantified formulas would be equivalent, but as we saw in the previous section it would make all proofs much longer due to the need of instantiating them using (Ax.5) and, possibly, to rename bound variables.

In step 2, we used the propositional tautology $(p \to (q \to r)) \to (q \to (p \to r))$, with the mapping $p \mapsto x \approx y$, $q \mapsto x \approx x$ and $r \mapsto y \approx x$.

We will use symmetry in later proofs as an inference rule, allowing us to conclude $y \approx x$ from $x \approx y$. We denote this by Sym(n), where n is the step where $x \approx y$ occurs in the proof; this notation can be seen as an abbreviation for including steps 1–5 of the previous proof with the appropriate instantiations of x and y, together with an application of MP with the formula in step n.

Finally we prove transitivity. The easiest way to to this is using the Deduction Theorem: we show first that $\{x \approx y, y \approx z\} \vdash_P x \approx z$.

1. $y \approx x \to (y \approx z \to x \approx z)$	(P1)
2. $x \approx y$	(Hyp)
3. $y \approx x$	Sym(2)
4. $y \approx z \to x \approx z$	MP(1,3)
5. $y \approx z$	(Hyp)
6. $x \approx z$	MP(4,5)

Since we did not use the generalization rule in this derivation, we can apply the Deduction Theorem twice to conclude that $\vdash_P x \approx y \to (y \approx z \to x \approx z)$. By appplying (Gen) three times, we obtain $\vdash_P \forall xyz.(x \approx y \to (y \approx z \to x \approx z))$.

We will also use transitivity in later proofs as an inference rule, denoting by Trans(n,m) the inclusion of a proof of the convenient instance of $x \approx y \to (y \approx z \to x \approx z)$ together with two steps of MP (with the formulas in steps n and m of the proof). ◁

Example. As another example, we prove that $\vdash_P \mathsf{1} + \mathsf{1} \approx \mathsf{2}$. The idea is simple: we use axioms (P5) and (P6) to unfold the recursive definition of sum $(1+1 = (1+0)' = 1' = 2)$, and properties of equality to simplify the result.

For clarity, we first present the derivation where we write out $\mathsf{1}$ and $\mathsf{2}$ explicitly.

1. $\mathsf{0}' + \mathsf{0}' \approx (\mathsf{0}' + \mathsf{0})'$	(P6)
2. $\mathsf{0}' + \mathsf{0} \approx \mathsf{0}'$	(P5)
3. $\mathsf{0}' + \mathsf{0} \approx \mathsf{0}' \to (\mathsf{0}' + \mathsf{0})' \approx \mathsf{0}''$	(P2)
4. $(\mathsf{0}' + \mathsf{0})' \approx \mathsf{0}''$	MP(3,2)
5. $\mathsf{0}' + \mathsf{0}' \approx \mathsf{0}''$	Trans(1,4)

Using the notation for natural numbers, this proof becomes

1. $\mathsf{1} + \mathsf{1} \approx (\mathsf{1} + \mathsf{0})'$	(P6)
2. $\mathsf{1} + \mathsf{0} \approx \mathsf{1}$	(P5)
3. $\mathsf{1} + \mathsf{0} \approx \mathsf{1} \to (\mathsf{1} + \mathsf{0})' \approx \mathsf{2}$	(P2)
4. $(\mathsf{1} + \mathsf{0})' \approx \mathsf{2}$	MP(3,2)
5. $\mathsf{1} + \mathsf{1} \approx \mathsf{2}$	Trans(1,4)

It is important to get used to reading e.g. the first formula in this proof as an instance of (P6), even though $'$ does not appear explicitly. ◁

Exercise 42. Write out the previous derivation in full, i.e., expand the last step into an invocation of the lemma $\forall xyz.(x \approx y \to (y \approx z \to x \approx z))$ followed by the explicit instantiation of variables x, y and z, and the appropriate steps of Modus Ponens. Can you see why using symmetry and transitivity of equality as derived inference rules is advantageous?

Exercise 43. Show that:

(a) $\vdash_P 2 + 2 \approx 4$

(b) $\vdash_P (2 + 1) + 1 \approx 4$

Example. We now show an example with multiplication, and prove that $\vdash_P 1 \times 1 \approx 1$. Similar to the previous example, we now use axioms (P7) and (P8) to unfold the recursive definition of multiplication ($1 \times 1 = 1 \times 0 + 1 = 0 + 1$) followed by the computation of $0 + 1$ in a manner similar to the previous example.

1. $1 \times 1 \approx 1 \times 0 + 1$	(P8)
2. $1 \times 0 + 1 \approx (1 \times 0 + 0)'$	(P6)
3. $1 \times 0 + 0 \approx 1 \times 0$	(P5)
4. $1 \times 0 \approx 0$	(P7)
5. $1 \times 0 + 0 \approx 0$	Trans(3,4)
6. $1 \times 0 + 0 \approx 0 \to (1 \times 0 + 0)' \approx 1$	(P2)
7. $(1 \times 0 + 0)' \approx 1$	MP(6,5)
8. $1 \times 0 + 1 \approx 1$	Trans(2,7)
9. $1 \times 1 \approx 1$	Trans(1,8)

◁

Exercise 44. Show that:

(a) $\vdash_P 2 \times 1 \approx 2$

(b) $\vdash_P 1 \times 2 \approx 2$

(c) $\vdash_P (2 \times 1) + 1 \approx 3$

(d) $\vdash_P 2 \times (1 + 1) \approx 4$

So far we have not used axioms (P3) and (P4). These axioms are used in proofs of *in*equality: given two unequal numbers, (P4) is applied until one of them reaches 0, and (P3) then yields the conclusion (possibly combined with symmetry of equality).

Example. We show that $\vdash_P \neg(2 \approx 4)$.

1. $2 \approx 4 \to 1 \approx 3$	(P4)
2. $1 \approx 3 \to 0 \approx 2$	(P4)
3. $(2 \approx 4 \to 1 \approx 3) \to ((1 \approx 3 \to 0 \approx 2) \to (2 \approx 4 \to 0 \approx 2))$	Prop

4. $(1 \approx 3 \to 0 \approx 2) \to (2 \approx 4 \to 0 \approx 2)$	MP(3,1)
5. $2 \approx 4 \to 0 \approx 2$	MP(4,2)
6. $\neg(0 \approx 2)$	(P3)
7. $(2 \approx 4 \to 0 \approx 2) \to (\neg(0 \approx 2) \to \neg(2 \approx 4))$	Prop
8. $\neg(0 \approx 2) \to \neg(2 \approx 4)$	MP(7,5)
9. $\neg(2 \approx 4)$	MP(8,6)

In step 3, we used the tautology $(p \to q) \to ((q \to r) \to (p \to r))$ with the mapping $p \mapsto 2 \approx 4$, $q \mapsto 1 \approx 3$ and $r \mapsto 0 \approx 2$. In step 7, we used the tautology $(p \to q) \to (\neg q \to \neg p)$ with the mapping $p \mapsto 2 \approx 4$ and $q \mapsto 0 \approx 2$. ◁

Exercise 45. Show that:

(a) $\vdash_P \neg(3 \approx 0)$

(b) $\vdash_P \neg(5 \approx 2)$

(c) $\vdash_P \neg(2 + 1 \approx 4)$

(d) $\vdash_P \neg((1 \times 1) + 1) \approx 3$

The last axiom in the theory of arithmetic is the *induction axiom*.

$$\varphi\,[x/0] \to [\forall x\,(\varphi \to \varphi[x/x']) \to \forall x\varphi] \qquad \text{(P9)}$$

In this axiom, x can be instantiated with any variable z, and φ can be instantiated by any first-order formula such that $FV(\varphi) = \{z\}$ (so φ has exactly one free variable). This axiom makes the theory of arithmetic a proper extension of first-order logic: unlike the previous axioms, which can be written in first-order logic by universally quantifying all of their free variables, this axiom includes a quantification over all first-order formulas. This cannot be written in first-order logic.

The induction axiom embodies the usual principle of induction over the natural numbers: in order to prove that some property φ holds for all natural numbers ($\forall x\varphi$), we first prove that it holds for 0 ($\varphi[x/0]$), and then show that if the property holds for some number x, then it also holds for $x + 1$ ($\forall x.(\varphi \to \varphi[x/x'])$).

Working with the induction axiom is the novel part of writing proofs in Peano arithmetic, and this axiom is essential to prove even the basic properties of equality and the arithmetic operations.

Example. As a first example of using the induction axiom, we prove that $\vdash_P \forall x.(0 + x \approx x)$. Observe that this is *not* the result of applying (Gen) to axiom (P5).

In order to match $\forall x.(0 + x \approx x)$ with the conclusion of the induction axiom, we take φ to be $0 + x \approx x$. Then $\varphi[x/0]$ becomes $0 + 0 \approx 0$, which is an instance of (P5), while $\varphi[x/x']$ becomes $0 + x' \approx x'$. We first prove that $\{\varphi\} \vdash_P \varphi[x/x']$, i.e., that $\{0 + x \approx x\} \vdash_P 0 + x' \approx x'$.

1. $0 + x \approx x$	(Hyp)
2. $0 + x \approx x \to (0 + x)' \approx x'$	(P2)
3. $(0 + x)' \approx x'$	MP(2,1)
4. $0 + x' \approx (0 + x)'$	(P6)
5. $0 + x' \approx x'$	Trans(4,3)

Using the Deduction Theorem, we conclude that $\vdash_P \mathsf{0} + x \approx x \to \mathsf{0} + x' \approx x'$, which corresponds to $\vdash_P \varphi \to \varphi[x/x']$. We can now complete our proof.

1. $\mathsf{0} + \mathsf{0} \approx \mathsf{0}$	P5
2. $\mathsf{0} + x \approx x \to \mathsf{0} + x' \approx x'$	Lemma
3. $\forall x.(\mathsf{0} + x \approx x \to \mathsf{0} + x' \approx x')$	Gen(2)
4. $\mathsf{0} + \mathsf{0} \approx \mathsf{0} \to ((\forall x.(\mathsf{0} + x \approx x \to \mathsf{0} + x' \approx x')) \to \forall x.(\mathsf{0} + x \approx x))$	(P9)
5. $(\forall x.(\mathsf{0} + x \approx x \to \mathsf{0} + x' \approx x')) \to \forall x.(\mathsf{0} + x \approx x)$	MP(4,1)
6. $\forall x.(\mathsf{0} + x \approx x)$	MP(5,3)

Although it may seem complex, this proof corresponds simply to a very detailed explanation of the informal mathematical proof by induction: the base case is $0 + 0 = 0$ (an axiom), while for the inductive step we have that $0 + x' = (0 + x)' = x'$ using the induction hypothesis. ◁

Example. The formula in the previous example is the counterpart to (P5) that could be used to define addition recursively on the first argument (instead of on the second). We now show that the similar counterpart to (P6), the formula $\forall xy.(x' + y \approx (x + y)')$, also holds.

We prove this by first showing that $\vdash_P \forall y.(x' + y \approx (x + y)')$. This is proven by induction on y. Taking φ to be $x' + y \approx (x + y)'$, we first prove $\varphi[y/\mathsf{0}]$.

1. $x' + \mathsf{0} \approx x'$	(P5)
2. $x + \mathsf{0} \approx x$	(P5)
3. $x + \mathsf{0} \approx x \to (x + \mathsf{0})' \approx x'$	(P2)
4. $(x + \mathsf{0})' \approx x'$	MP(3,2)
5. $x' \approx (x + \mathsf{0})'$	Sym(4)
6. $x' + \mathsf{0} \approx (x + \mathsf{0})'$	Trans(1,5)

For the inductive step, we again resort to the Deduction Theorem, and start by proving that $\{x' + y \approx (x + y)'\} \vdash_P x' + y' \approx (x + y')'$. We use axiom (P6) to rewrite the expression $x' + y'$, in order to obtain something that helps us use the hypothesis.

1. $x' + y' \approx (x' + y)'$	(P6)
2. $x' + y \approx (x + y)'$	(Hyp)
3. $x + y' \approx (x + y)'$	(P6)
4. $(x + y)' \approx x + y'$	Sym(3)
5. $x' + y \approx x + y'$	Trans(2,4)
6. $x' + y \approx x + y' \to (x' + y)' \approx (x + y')'$	(P2)
7. $(x' + y)' \approx (x + y')'$	MP(6,5)
8. $x' + y' \approx (x + y')'$	Trans(1,7)

From the Deduction Theorem we conclude that $\vdash_P x' + y \approx (x + y)' \to x' + y' \approx (x + y')'$, and by applying (Gen) we obtain $\vdash_P \forall y.(x' + y \approx (x + y)' \to x' + y' \approx (x + y')')$.

We can now complete our proof using the induction axiom.

1. $\underbrace{x' + \mathsf{0} \approx (x + \mathsf{0})'}_{\varphi[y/\mathsf{0}]}$	Lemma
2. $\forall y.(\underbrace{x' + y \approx (x + y)'}_{\varphi} \rightarrow \underbrace{x' + y' \approx (x + y')'}_{\varphi[y/y']})$	Lemma
3. $\varphi[y/\mathsf{0}] \rightarrow ((\forall y(\varphi \rightarrow \varphi[y/y'])) \rightarrow \forall y.\underbrace{(x' + y \approx (x + y)')}_{\varphi})$	(P9)
4. $(\forall y(\varphi \rightarrow \varphi[y/y'])) \rightarrow \forall y.(x' + y \approx (x + y)')$	MP(3,1)
5. $\forall y.(x' + y \approx (x + y)')$	MP(4,2)
6. $\forall xy.(x' + y \approx (x + y)')$	Gen(5)

This concludes the proof that $\forall xy.(x' + y \approx (x + y)')$. ◁

As a last example, we show that addition is commutative.

Example. Using the previous results, we prove that $\vdash_P \forall xy.(x + y \approx y + x)$. Again, we start by using induction on y over formula $x + y \approx y + x$, which we denote by φ.

The base case is simple.

1. $x + \mathsf{0} \approx x$	(P5)
2. $\mathsf{0} + x \approx x$	Lemma
3. $x \approx \mathsf{0} + x$	Sym(2)
4. $x + \mathsf{0} \approx \mathsf{0} + x$	Trans(1,3)

The lemma in step 2 is simply an instantiation of a formula proved earlier.

For the inductive step, we start as usual by assuming $x + y \approx y + x$ as a hypothesis.

1. $x + y' \approx (x + y)'$	(P5)
2. $x + y \approx y + x$	(Hyp)
3. $x + y \approx y + x \rightarrow (x + y)' \approx (y + x)'$	(P2)
4. $(x + y)' \approx (y + x)'$	MP(3,2)
5. $x + y' \approx (y + x)'$	Trans(1,4)
6. $y' + x \approx (y + x)'$	Lemma
7. $(y + x)' \approx y' + x$	Sym(6)
8. $x + y' \approx y' + x$	Trans(5,7)

The lemma in step 6 is obtained from the formula in the previous example by first performing a change of bound variables, and afterwards instantiating them as shown. By applying the Deduction Theorem and then generalizing over y, we conclude that $\vdash_P \forall y.(x + y \approx y + x \rightarrow x + y' \approx y' + x)$.

Finally, we apply the induction axiom.

1. $\underbrace{x + \mathsf{0} \approx \mathsf{0} + x}_{\varphi[y/\mathsf{0}]}$	Lemma
2. $\forall y.(\underbrace{x + y \approx y + x}_{\varphi} \rightarrow \underbrace{x + y' \approx y' + x}_{\varphi[y/y']})$	Lemma

3. $\varphi[y/\mathsf{0}] \to ((\forall y(\varphi \to \varphi[y/y'])) \to \forall y.\underbrace{(x + y \approx y + x)}_{\varphi})$	(P9)
4. $(\forall y(\varphi \to \varphi[y/y'])) \to \forall y.(x + y \approx y + x)$	MP(3,1)
5. $\forall y.(x + y \approx y + x)$	MP(4,2)
6. $\forall xy.(x + y \approx y + x)$	Gen(5)

We have thus shown that $\vdash_P \forall xy.(x + y \approx y + x)$. ◁

Exercise 46. Formally derive the lemmas in the previous example from the universally quantified formulas proved earlier.

Exercise 47. Show that the following properties of multiplication are provable in the theory of arithmetic.

(a) $\vdash_P \forall x.(\mathsf{0} \times x \approx \mathsf{0})$ (b) $\vdash_P \forall xy.(x' \times y \approx x \times y + y)$ (c) $\vdash_P \forall xy.(x \times y \approx y \times x)$

As we mentioned earlier, the induction principle is not expressible in first-order. This has some possibly unexpected consequences: in the theory of arithmetic, we can prove that every term is equal to some natural number.

Example. From Peano's axioms, we can prove that $\vdash_P \forall x.(x \approx \mathsf{0} \vee \exists y.(x \approx y'))$.

We prove this property by induction on x, taking φ to be $x \approx \mathsf{0} \vee \exists y.(x \approx y')$.

The informal proof is as follows. For the base case, reflexivity immediately gives us $0 = 0$, which establishes the first disjunct (steps 1–3 in the derivation below). For the inductive case, we also use reflexivity to infer $x' = x'$, from which the second disjunct follows taking $y = x$ (steps 4–7 in the derivation below).

1. $\mathsf{0} \approx \mathsf{0}$	Refl($\mathsf{0}$)
2. $\mathsf{0} \approx \mathsf{0} \to (\mathsf{0} \approx \mathsf{0} \vee \exists y.(\mathsf{0} \approx y'))$	Prop
3. $\underbrace{\mathsf{0} \approx \mathsf{0} \vee \exists y.(\mathsf{0} \approx y')}_{\varphi[x/\mathsf{0}]}$	MP(2,1)
4. $x' \approx x'$	Refl(x')
5. $x' \approx x' \to \exists y.(x' \approx y')$	(Lem.$\exists$)
6. $\exists y.(x' \approx y')$	MP(5,4)
7. $(\exists y.(x' \approx y')) \to ((\mathsf{0} \approx x \vee \exists y.(x \approx y')) \to (\mathsf{0} \approx x' \vee \exists y.(x' \approx y')))$	Prop
8. $(\mathsf{0} \approx x \vee \exists y.(x \approx y')) \to (\mathsf{0} \approx x' \vee \exists y.(x' \approx y'))$	MP(7,6)
9. $\forall x.(\underbrace{(\mathsf{0} \approx x \vee \exists y.(x \approx y'))}_{\varphi} \to \underbrace{(\mathsf{0} \approx x' \vee \exists y.(x' \approx y'))}_{\varphi[x/x']})$	Gen(9)
10. $\varphi[x/\mathsf{0}] \to (\forall x\,(\varphi \to \varphi[x/x']) \to \forall x\varphi)$	(P9)
11. $\forall x\,(\varphi \to \varphi[x/x']) \to \forall x\varphi$	MP(9,3)
12. $\forall x\varphi$	MP(11,8)

In step 2 we used the propositional tautology $p \to (p \vee q)$ with the mapping $p \mapsto \mathsf{0} \approx \mathsf{0}$ and $q \mapsto \exists y.(\mathsf{0} \approx y')$. In step 7 we used the propositional tautology $p \to (q \to (r \vee p))$ with the mapping $p \mapsto \exists y.(x' \approx y')$, $q \mapsto (\mathsf{0} \approx x \vee \exists y.(x \approx y'))$ and $r \mapsto \mathsf{0} \approx x'$. ◁

This result has some interesting consequences. For example, if we require that $\approx$ be interpreted as equality, then the upwards Skolem–Löwenheim Theorem fails, since this formula states that every element in the domain must be equal to the interpretation of a natural number – and therefore the domain must be countable. More generally, this formula implies that any model of the theory of arithmetic is isomorphic to the "intuitive" one whose domain is $\mathbb{N}$ and the function and predicate symbols are interpreted in the obvious way.

We now show some more examples of how simple properties of natural numbers can be proved in the theory of arithmetic, using the induction axiom.

Example. An important property of arithmetic is that equality is compatible with the arithmetic operations, in the sense that if we add or multiply equal quantities we should get equal results (recall the axiom (M7) in the theory of monoids, Example 9).

We prove that equality is compatible with addition on the left, i.e., $\vdash_P \forall xyz.(y \approx z \to y + x \approx z + x)$. A simple way to obtain this proof is to start by proving by induction on x that $\{y \approx z\} \vdash_P \forall x.(y + x \approx z + x)$.

For the base case, we first prove that $\{y \approx z\} \vdash_P y + \mathsf{0} \approx z + \mathsf{0}$.

1. $y + \mathsf{0} \approx y$	(P5)
2. $y \approx z$	(Hyp)
3. $y + \mathsf{0} \approx z$	Trans(1,2)
4. $z + \mathsf{0} \approx z$	(P5)
5. $z \approx z + \mathsf{0}$	Sym(4)
6. $y + \mathsf{0} \approx z + \mathsf{0}$	Trans(3,5)

For the inductive step, we start by proving that $\{y \approx z, y + x \approx z + x\} \vdash_P y + x' \approx z + x'$.

1. $y + x' \approx (y + x)'$	(P6)
2. $y + x \approx z + x$	(Hyp)
3. $y + x \approx z + x \to (y + x)' \approx (z + x)'$	(P2)
4. $(y + x)' \approx (z + x)'$	MP(3,2)
5. $y + x' \approx (z + x)'$	Trans(1,4)
6. $z + x' \approx (z + x)'$	(P6)
7. $(z + x)' \approx z + x'$	Sym(6)
8. $y + x' \approx z + x'$	Trans(5,7)

Applying the Deduction Theorem, we can conclude that $\{y \approx z\} \vdash_P y + x \approx z + x \to y + x' \approx z + x'$. By applying (Gen), we obtain $\{y \approx z\} \vdash_P \forall x.(y + x \approx z + x \to y + x' \approx z + x')$.

Finally we combine these proofs with the induction axiom.

1. $\underbrace{y + 0 \approx z + 0}_{\varphi[x/0]}$	Lemma
2. $\forall x.(\underbrace{y + x \approx z + x}_{\varphi} \to \underbrace{y + x' \approx z + x'}_{\varphi[x/x']})$	Lemma
3. $\varphi \to ((\forall x.(\varphi \to \varphi[x/x'])) \to \forall x.(\underbrace{y + x \approx z + x}_{\varphi}))$	(P9)
4. $(\forall x\,(\varphi \to \varphi[x/x']) \to \forall x.(y + x \approx z + x))$	MP(3,1)
5. $\forall x.(y + x \approx z + x)$	MP(4,2)
6. $\forall x.(y + x \approx z + x) \to y + x \approx z + x$	(Ax.5)
7. $y + x \approx z + x$	MP(6,5)

Again invoking the Deduction Theorem, we can conclude that $\vdash_P y \approx z \to y + x \approx z + x$. Applying (Gen) over z, y and finally x yields $\vdash_P \forall xyz.(y \approx z \to y + x \approx z + x)$. ◁

Exercise 48. Show that:

(a) $\vdash_P \forall xyz.(y \approx z \to x + y \approx x + z)$

(b) $\vdash_P \forall x_1x_2y_1y_2.(x_1 \approx x_2 \to (y_1 \approx y_2 \to (x_1 + y_1 \approx x_2 + y_2)))$

Hint. The first formula can be proven from the property in the last example, using commutativity of addition. The second formula follows by combining both results.

Exercise 49. Show that:

(a) $\vdash_P \forall xyz.(y \approx z \to y \times x \approx z \times x)$

(b) $\vdash_P \forall xyz.(y \approx z \to x \times y \approx x \times z)$

(c) $\vdash_P \forall x_1x_2y_1y_2.(x_1 \approx x_2 \to (y_1 \approx y_2 \to (x_1 \times y_1 \approx x_2 \times y_2)))$

As a final example, we prove that sum is associative.

Example. We now prove that $\vdash_P \forall xyz.((x + y) + z \approx x + (y + z))$. To show this, we take φ to be $(x + y) + z \approx x + (y + z)$ and use induction over z.

For the base case, we need to show that $\vdash_P (x + y) + 0 \approx x + (y + 0)$.

1. $(x + y) + 0 \approx x + y$	(P5)
2. $(y + 0) \approx y$	(P5)
3. $(y + 0 \approx y) \to x + (y + 0) \approx x + y$	Lemma
4. $x + (y + 0) \approx x + y$	MP(3,2)
5. $x + y \approx x + (y + 0)$	Sym(4)
6. $(x + y) + 0 \approx x + (y + 0)$	Trans(1,5)

The lemma used in this proof is again an instance of a formula proved earlier.

For the inductive step, we again use the Deduction Theorem, and start by showing that $\{(x + y) + z \approx x + (y + z)\} \vdash_P (x + y) + z' \approx x + (y + z')$.

1. $(x + y) + z' \approx ((x + y) + z)'$	(P6)
2. $(x + y) + z \approx x + (y + z)$	(Hyp)
3. $(x + y) + z \approx x + (y + z) \rightarrow ((x + y) + z)' \approx (x + (y + z))'$	(P2)
4. $((x + y) + z)' \approx (x + (y + z))'$	MP(3,2)
5. $(x + y) + z' \approx (x + (y + z))'$	Trans(1,4)
6. $y + z' \approx (y + z)'$	(P6)
7. $y + z' \approx (y + z)' \rightarrow x + (y + z') \approx x + (y + z)'$	Lemma
8. $x + (y + z') \approx x + (y + z)'$	MP(7,6)
9. $x + (y + z)' \approx (x + (y + z))'$	(P6)
10. $x + (y + z') \approx (x + (y + z))'$	Trans(8,9)
11. $(x + (y + z))' \approx x + (y + z')$	Sym(10)
12. $(x + y) + z' \approx x + (y + z')$	Trans(5,11)

As before, the Deduction Theorem and an application of generalization yield $\vdash_P \forall z.((x + y) + z \approx x + (y + z) \rightarrow (x + y) + z' \approx x + (y + z'))$.

We now combine the previous derivations in the final proof.

1. $\varphi[z/0]$	Lemma
2. $\forall z(\varphi \rightarrow \varphi[z/z'])$	Lemma
3. $\varphi[z/0] \rightarrow ((\forall z(\varphi \rightarrow \varphi[z/z'])) \rightarrow \forall z.((x + y) + z \approx x + (y + z)))$	(P9)
4. $(\forall z(\varphi \rightarrow \varphi[z/z'])) \rightarrow \forall z.((x + y) + z \approx x + (y + z))$	MP(3,1)
5. $\forall z.((x + y) + z \approx x + (y + z))$	MP(4,2)
6. $\forall yz.((x + y) + z \approx x + (y + z))$	Gen(5)
7. $\forall xyz.((x + y) + z \approx x + (y + z))$	Gen(6)

This concludes our proof. ◁

Exercise 50. Use the theory of arithmetic to prove that multiplication is associative, i.e., show that $\vdash_P \forall xyz.((x \times y) \times z \approx x \times (y \times z))$.

4.5.2 Godelizations and representability

Since the theory of Peano arithmetic is not a first-order theory, we cannot invoke Gödel's completeness theorem to claim that all true statements about natural numbers are provable from the axioms of this theory. Indeed, Gödel himself proved that this is not the case – more strongly, he proved that the theory of arithmetic is expressive enough that we can write a first-order formula that is true iff it is not provable (assuming soundness).

This proof strategy has since been used in different contexts – it is closely connected to Turing's proof of undecidability of the halting problem, for example. The main idea is to encode the system inside itself: we write formulas using the signature of arithmetic that describe the whole process of writing proofs in the theory of arithmetic. In this section, we present such an encoding.

The construction we give is slightly more general than the description above: it can be applied to *any* signature containing Σ_P, and *any* theory including the theory of arithmetic, as long as both signature and theory fulfill some additional computability requirements. These are formally expressed by the notion of godelization.

Definition. A *godelization* of a set U is an injective function $g : U \to \mathbb{N}$ such that:

- g is computable;
- $g(U)$ is a decidable set;
- there is an algorithm that, given $n \in g(U)$, finds u with $g(u) = n$.

A set $S \subseteq U$ is (semi-)decidable if $g(S)$ is (semi-)decidable.

A godelization is an algorithmic way to transform an arbitrary set into a set of natural numbers. Thus, we can lift concepts from computability theory (which are defined for natural numbers) to an arbitrary set. We often say that $g(u)$ is the *Gödel number* of u.

Observe that, if there is a godelization of U, then U is necessarily countable.

Given a first-order signature $\Sigma = \langle \mathcal{F}, \mathcal{P} \rangle$, we assume that there is a godelization $g : \Sigma \to \mathbb{N}$ such that each F_n and each P_n is a decidable set, and $\bigcup_{n\in\mathbb{N}} F_n$ and $\bigcup_{n\in\mathbb{N}} P_n$ are also decidable. In other words, we can decide whether a natural number corresponds to a function symbol or a predicate symbol, with or without specifying the arity.

In particular, the sets F_n and P_n are all countable, and we can write f_k^n (respectively p_k^n) to denote the k-th element of F_n (respectively P_n) according to the godelization g. If F_n or P_n are finite, then f_k^n and p_k^n are undefined when k is larger than the number of elements in F_n or P_n.

Every godelization of Σ can be systematically extended to the set of all formulas. There are several standard ways to do this; the one we present is one that happens to be cherished by the authors of these notes.

Definition. Let g be a godelization of Σ. We define a godelization g^T of the set of all terms over Σ as follows.

$$\begin{aligned} g^T(c) &= 3 \times g(c) \\ g^T(x_i) &= 3 \times i + 1 \\ g^T\left(f_k^n(t_1, \ldots, t_n)\right) &= 3 \times 2^n \times 3^k \times 5^{g^T(t_1)} \times \cdots \times p_{n+2}^{g^T(t_n)} + 2 \end{aligned}$$

where p_n denotes the n-th prime number (so $p_0 = 2$, $p_1 = 3$, etc.)

This function is indeed a godelization: to compute its inverse, given a number m we first divide m by 3 and look at the remainder to determine whether the corresponding term is a constant, a variable or a function application. In the last case, we then decompose $(m-2)/3$ in prime factors to determine k, n and $g^T(t_i)$, and recursively continue. If this process fails at some point (e.g. because the number of arguments to a function is wrong, or because there is no constant c with $g(c) = m/3$), then m does not represent any term.

Example. Consider the terms c_2, $f_1^1(x_0)$, and $f_0^2\left(c_2, f_1^1(x_0)\right)$. Following the above definition, we have that:

$$\begin{aligned} g^T(c_2) &= 3 \times 2 = 6 \\ g^T\left(f_1^1(x_0)\right) &= 3 \times 2^1 \times 3^1 \times \underbrace{5^{g^T(x_0)}}_{5^{3\times 0+1}=5} + 2 = 3 \times 2 \times 3 \times 5 + 2 = 92 \\ g^T\left(f_0^2\left(c_2, f_1^1(x_0)\right)\right) &= 3 \times 2^2 \times 3^0 \times 5^{g^T(c_2)} \times 7^{g^T\left(f_1^1(x_0)\right)} + 2 \\ &= 3 \times 2^2 \times 3^0 \times 5^6 \times 7^{92} + 2 = 12 \times 5^6 \times 7^{92} + 2 \end{aligned}$$

The last value is of the order of magnitude of 10^{83}. These examples show that g^T quickly becomes extremely large, even for simple terms. ◁

Example. Now we consider the inverse problem – that of determining the term corresponding to a given number.

- To solve $g^T(t) = 127$, we first divide 127 by 3, obtaining quotient 42 and remainder 1. From the remainder we know that t is a variable, and 42 is the index of that variable. Thus t is x_{42}.

- To solve $g^T(t) = 36$, we again start by dividing 36 by 3, obtaining as a result 12 and remainder 0. The remainder tells is that t is a constant, and 12 is its index. Therefore t is c_{12}.

- To solve $g^T(t) = 157{,}502$, we also start by dividing 8 by 3, obtaining the result 52,500 and remainder 2. Therefore this term is a function application, and to continue we need to factor 52,500 in prime factors. We obtain $52{,}500 = 2^2 \times 3^1 \times 5^4 \times 7^1$. The exponents of 2 and 3 tell us that the function symbol being applied is f_1^2, and the arguments are the terms t_1 and t_2 such that $g^T(t_1) = 4$ and $g^T(t_2) = 1$.

  We now solve these two equations using the same process. Since $4 = 3 \times 1 + 1$, we conclude that $t_1 = x_1$; and from $1 = 3 \times 0 + 1$ we recover $t_2 = x_0$. Therefore t is the term $f_1^2(x_1, x_0)$.

- To solve $g^T(t) = 15{,}437$, we again start by dividing 15,437 by 3, obtaining 5,145 and remainder 2. The remainder tells us that t is again a function application, so we continue by finding the prime factorization of 5,145. Since $5{,}145 = 2^0 \times 3^1 \times 5^1 \times 7^3$, we conclude that it does not correspond to a valid term: the function symbol t should start with is f_1^0, but since this symbol does not take arguments the exponents of all prime numbers higher than 3 should be 0.

  So 15,437 is not the Gödel number of any term.

- To solve $g^T(t) = 35{,}282$, we start by dividing 35,282 by 3, obtaining 11,760 and remainder 2. Again t is a function application, so we factor 11,760 in prime factors. Since $11{,}760 = 2^4 \times 3^1 \times 5^1 \times 7^2$, we conclude that it is the application of function symbol f_1^4 to the terms with Gödel numbers 1, 2, 0 and 0 (the last two zeroes are the exponents of 11 and 13, the next two prime numbers, in this factorization).

  Now we recur: $g^T(t') = 1$ has the solution $t' = x_0$, since $1 = 3 \times 0 + 1$, and $g^T(t') = 0$ has the solution $t' = c_0$, since $0 = 3 \times 0$. Solving $g^T(t') = 2$ requires again a bit more work: since $2 = 3 \times 0 + 2$, we know that t' is a function application. But 0 cannot be written

as a product of prime numbers, therefore there is no solution to this equation. So 35,282 is also not the Gödel number of any term. ◁

Example. If we choose the godelization for Σ_P such that $f_0^0 = \mathsf{0}$, $f_0^1 = '$, $f_0^2 = +$, $f_1^2 = \times$ and $p_0^2 = \approx$, then:

- $g(\mathsf{0}) = 0$;
- $g(\mathsf{1}) = 3 \times 2^1 \times 3^0 \times 5^0 + 2 = 8$;
- $g(\mathsf{2}) = 3 \times 2^1 \times 3^0 \times 5^8 + 2 = 2{,}343{,}752$;
- $g(\mathsf{1} + \mathsf{0}) = 3 \times 2^2 \times 3^0 \times 5^8 \times 7^0 + 2 = 4{,}687{,}502$;
- $g(x_1 \times \mathsf{0}) = 3 \times 2^2 \times 3^1 \times 5^5 \times 7^0 + 2 = 112{,}502$;
- there is no term with Gödel number 36 (since c_{12} is not defined);
- the term with Gödel number 157,502 is $x_1 \times x_0$. ◁

Exercise 51. Compute the Gödel numbers of the following terms.

(a) $f_4^1(x_3)$ (b) x_{25} (c) $f_0^3(c_1, c_0, x_0)$ (d) $f_2^2\left(f_0^1(c_0), x_2\right)$

Exercise 52. If we were only interested in working with the theory of arithmetic, we could define more efficient godelizations of the set of terms directly. A possibility would be the function g^P defined as follows.

$$g^P(\mathsf{0}) = 0 \qquad g^P(x_n) = 4n + 1 \qquad g^P(t_1 + t_2) = 4\Phi(t_1, t_2) + 3$$
$$g^P(t') = 4g^P(t) + 2 \qquad g^P(t_1 \times t_2) = 4\Phi(t_1, t_2) + 4$$

where Φ is the function $\Phi(n, m) = \frac{(m+n)(m+n+1)}{2} + n$.

(a) Show that Φ is a godelization of $\mathbb{N}^2$. (Hint: start by finding out how to compute n and m from $\Phi(n, m)$.)

(b) Show that g^P is a godelization of the set of all terms in the theory of arithmetic.

(c) Show that g^P is surjective: every natural number is the Gödel number of some term.

(d) Compute $g^P(\mathsf{2})$, $g^P(\mathsf{1} + \mathsf{0})$ and $g^P(x_1 \times \mathsf{0})$.

(e) Find terms t_1, t_2, t_3 and t_4 such that $g^P(t_1) = 36$, $g^P(t_2) = 15{,}437$, $g^P(t_3) = 35{,}282$ and $g^P(t_4) = 157{,}502$.

Using a similar idea, we can now define a godelization of all formulas over Σ.

Definition. Given a godelization g of Σ, we can extend it to a godelization g^F of the set of all first-order formulas over Σ as follows.

$$\begin{aligned} g^F\left(p_k^n\left(t_1,\ldots,t_n\right)\right) &= 4\times 2^n\times 3^k\times 5^{g^T(t_1)}\times\cdots\times p_{n+2}^{g^T(t_n)} \\ g^F(\neg\varphi) &= 4\times g^F(\varphi)+1 \\ g^F(\varphi\to\psi) &= 4\times\Phi\left(g^F(\varphi),g^F(\psi)\right)+2 \\ g^F\left(\forall x_i\varphi\right) &= 4\times\Phi\left(i,g^F(\varphi)\right)+3 \end{aligned}$$

where Φ is the godelization of $\mathbb{N}^2$ given by $\Phi(n,m)=\frac{(m+n)(m+n+1)}{2}+n$.

Example. Consider the formula $\varphi=\forall x_1\left(\neg p_1^2\left(x_1,c_0\right)\right)$. First we compute $g^T(x_1)=3\times 1+1=4$ and $g^T(c_0)=3\times 0=0$. From this we can derive $g^F\left(p_1^2\left(x_1,c_0\right)\right)=4\times 2^2\times 3^1\times 5^4\times 7^0=7{,}500$. Then $g^F\left(\neg p_1^2\left(x_1,c_0\right)\right)=4\times 7{,}500+1=30{,}001$, and finally $g^F(\varphi)=4\times\Phi(1,30{,}001)=450{,}105{,}004$. ◁

The crucial point in our proof is that the theory of arithmetic is expressive enough to encode its own derivations, expressed as predicates over natural numbers corresponding to formulas.

Definition. A relation $R\subseteq\mathbb{N}^n$ is *representable* in Peano arithmetic if there exists a formula φ_R with n free variables $y_1,\ldots,y_n$ such that $R(k_1,\ldots,k_n)$ iff $\vdash_P\varphi_R\left[y_1/\mathsf{k_1},\ldots,y_n/\mathsf{k_n}\right]$.[6]

A function $f:\mathbb{N}^n\to\mathbb{N}$ is *representable* if there exists a formula φ_f with $n+1$ free variables $y_1,\ldots,y_{n+1}$ such that:

- $f(k_1,\ldots,k_n)=k$ iff $\vdash_P\varphi_f\left[y_1/\mathsf{k_1},\ldots,y_n/\mathsf{k_n},y_{n+1}/\mathsf{k}\right]$;
- $\vdash_P\exists^1 y_{n+1}\varphi_f\left[y_1/\mathsf{k_1},\ldots,y_n/\mathsf{k_n}\right]$ for all $k_1,\ldots,k_n$.

where $\exists^1 x\psi$, which intuitively reads "there exists a unique x such that ψ", is defined as $(\exists x\psi)\wedge(\forall xy.((\psi\wedge\psi[x/y])\to x\approx y))$.

Representability of a function is a specialization of the corresponding notion of representability of a relation: a function $f:\mathbb{N}^n\to\mathbb{N}$ is a relation $R\subseteq\mathbb{N}^{n+1}$ with the additional property that for each $k_1,\ldots,k_n$ there is exactly one k such that $R(k_1,\ldots,k_n,k)$ – namely, $k=f(k_1,\ldots,k_n)$.

Theorem 35. Every recursive relation and function is representable in the theory of arithmetic.

We omit the proof of this theorem, as it requires some background in computability theory. There are several possibilities, depending on which model of recursion theory one prefers. One possibility is to show that every Turing machine can be encoded as a finite set of formulas, and use this construction to show that the function or relation computed by every Turing machine is representable. Another possibility is to consider the set of partial recursive functions as defined by Kleene and show by structural induction that every partial recursive function can be represented by a first-order formula. The choice of computation model is immaterial, assuming the Church–Turing thesis.

[6] Recall that k is a natural number, while k is the syntactic term in the theory of arithmetic obtained by k applications of $'$ to $\mathsf{0}$.

Example. As we do not prove Theorem 35, we show instead directly that some relations and functions are representable.

- The relation $\leq$ is representable. Indeed, consider the formula $\varphi_{\leq}$ defined as $\exists z.(x+z \approx y)$. Then $\varphi_{\leq}$ has the free variables x and y, and $\varphi_{\leq}(x/\mathsf{n}, y/\mathsf{m})$ is true precisely when $n \leq m$ (as then we can instantiate z with $\mathsf{m} - \mathsf{n}$).
- The remainder function is representable. Indeed, consider the formula φ_{mod} defined as $(\exists w.(x \times w + z \approx y)) \wedge (\varphi_{\leq}(\mathsf{0}, z) \wedge \varphi_{\leq}(z', y))$. This formula has three variables x, y and z. Furthermore, if $\varphi_{\mathsf{mod}}(x/\mathsf{n}, y/\mathsf{m}, z/\mathsf{r})$ holds, then the last two conditions constrain r to be between 0 and $m-1$, and the only instantiation of w that can make the existentially quantified subformula true is $\frac{\mathsf{m}-\mathsf{r}}{\mathsf{n}}$ – and only in the case that $m - r$ is divisible by n. There is only one value of r with this property, namely the remainder of n divided by m.
- The set of even numbers is representable by the formula $\varphi_{\mathsf{mod}}(x, \mathsf{2}, \mathsf{0})$. ◁

Representing more complex functions, such as exponentials, requires more advanced techniques that are beyond the scope of this course.

Exercise 53. Show directly that the following relations are representable:

(a) $\{m \mid m > 3\}$ (b) $\{\langle m, n\rangle \mid m < n\}$ (c) $\{\langle m, n\rangle \mid m \neq n\}$

Exercise 54. Show directly that the following functions are representable. The notation $\lfloor x \rfloor$ denotes the largest integer not greater than x.

(a) $\lambda mn.|m - n|$

(b) $\lambda mn.\begin{cases} m-n, & n \geq m \\ 0, & \text{otherwise} \end{cases}$

(c) $\lambda n.n^3 + 2n^2 + n$

(d) $\lambda mn.\lfloor m/n \rfloor$ (division with remainder)

(e) $\lambda n.\lfloor \sqrt{n} \rfloor$ (integer square root)

4.5.3 Incompleteness of Peano arithmetic

We now have the necessary ingredients to present Gödel's proof of incompleteness of Peano arithmetic. As is usual, we work assuming the Church–Markov thesis: every function that is effectively computable (i.e., that can be computed by a precisely formulated process in natural language) is recursive. According to Theorem 35, every such function is representable.[7]

The setting for this theorem is a theory T over a signature Σ, such that T includes the theory of arithmetic and the set of axioms of T is recursive. The idea is to use a diagonalization argument in the style of Cantor to obtain a formula χ such that both $\vdash_T \chi$ and $\vdash_T \neg\chi$ lead to a contradiction. This requires encoding derivability as a relation on Gödel numbers of formulas and sequences of formulas.

[7]For completeness' sake we mention that there exist explicit formalizations of the construction we describe below, where all the formulas are written down precisely. We see no advantage in including such a presentation, as such an exercise is mostly mechanical if one understands the principles explained in the text.

Definition. Given a godelization g^F of the set of formulas of T, we define g^* to be its canonical extension to sequences, given by

$$g^*(\varphi_1, \ldots, \varphi_n) = p_1^{g^F(\varphi_1)} \times \cdots \times p_n^{g^F(\varphi_n)}.$$

Lemma 41. The relation D defined by $D(x, y)$ iff x is the Gödel number of a Hilbert-style proof of y is representable.

Proof. By Theorem 35, it suffices to show that D is recursive. Given x and y, we decide whether $D(x, y)$ holds as follows: we factor x in prime factors, which allows us to compute $\varphi_1, \ldots, \varphi_n$ (since g^F is a godelization). We then check whether $g^F(\varphi_n) = y$ and whether each φ_i is an instance of an axiom or follows from applying an inference rule to elements of $\{\varphi_1, \ldots, \varphi_{i-1}\}$; since this set is finite, this procedure always terminates. If all tests succeed, we conclude that $D(x, y)$ holds, otherwise $D(x, y)$ does not hold. □

The next step is defining two relations $\omega_1, \omega_2 \subseteq \mathbb{N}^2$ that specialize D in a way that embodies a form of self-reference.

Lemma 42. Define a relation $\omega_1 \subseteq \mathbb{N}^2$ as follows: $\omega_1(u, v)$ holds iff $u = g^F(\varphi)$ for some formula φ with one free variable w, $v = g^*(\varphi_1, \ldots, \varphi_n)$ for some sequence of formulas $\varphi_1, \ldots, \varphi_n$, and $\varphi_1, \ldots, \varphi_n$ is a valid derivation in T of $\varphi[w/\mathsf{u}]$.

Define $\omega_2 \subseteq \mathbb{N}^2$ in a similar way, but requiring $\varphi_1, \ldots, \varphi_n$ to be a valid derivation of $\neg\varphi[w/\mathsf{u}]$.

Then ω_1 and ω_2 are both recursive relations.

Proof. Straightforward adaptation of the proof of the previous lemma. □

Since ω_1 and ω_2 are recursive, there exist formulas ψ_1 and ψ_2 representing ω_1 and ω_2, respectively. Without loss of generality, we assume that $FV(\psi_1) = FV(\psi_2) = \{x, y\}$.

We now define χ to be the formula $\forall y.\, (\psi_1 \to \exists z.(z \leq y \wedge \psi_2[y/z]))$, whose only free variable is x. From its construction, χ can be read as: if x is $g^F(\varphi)$ and y corresponds to a derivation of $\varphi[w/x]$, then there exists $z \leq y$ that corresponds to a derivation of $\neg\varphi[w/x]$. (Note that $\leq$ is not a predicate symbol in Σ_P, but since it is a recursive relation we can represent the condition $z \leq y$ by a first-order formula with free variables y and z; we gave an example of such a formula earlier.)

For any formula φ, let $c = g^F(\varphi)$. If $\models \chi[x/\mathsf{c}]$, then $\nvdash_T \psi[w/\mathsf{c}]$: $\vdash_T \varphi[w/\mathsf{c}]$ would imply that also $\vdash_T \neg\varphi[w/\mathsf{c}]$, and we assumed T to be consistent.

The final trick is to choose the correct instance of φ.

Theorem 36 (Gödel's Incompleteness Theorem)**.** If T is consistent, then there is a closed formula ξ such that $\models \xi$ but $\nvdash \xi$.

Proof. Let $c = g^F(\chi)$ and take ξ to be the formula $\chi\,[x/\mathsf{c}]$.

- If $\vdash_T \xi$, then $\models \xi$, whence $\nvdash_T \chi\,[w/\mathsf{c}]$. But this formula is exactly ξ, so $\nvdash_T \xi$, which is a contradiction.
- If $\vdash_T \neg\xi$, then $\models \neg\xi$, whence $\vdash_T \chi\,[w/\mathsf{c}]$. But this formula is exactly ξ, so we also conclude that $\vdash_T \xi$, contradicting the assumption that T is consistent.

Therefore $\nvdash_T \xi$ and $\nvdash_T \neg\xi$. However, ξ is a closed formula using only symbols in the theory of arithmetic. Since the interpretation of the closed terms in the theory of arithmetic must be isomorphic to the natural numbers in all models of T, it follows that either $\models \xi$ or $\models \neg\xi$. Since the latter yields a contradiction, it must be the case that $\models \xi$ and $\nvdash_T \xi$. □

4.6 Exercises

Exercise 55. Show that, for any interpretation I and assignment ρ, $I \models_\rho \exists x_1 \dots \exists x_n \varphi$ iff $I \models_\sigma \varphi$ for some σ that is $\{x_1, \dots, x_n\}$-equivalent to ρ.

Exercise 56. Let $\forall\varphi$ stand for the formula $\forall x_1 \dots x_n \varphi$, where $\{x_1, \dots, x_n\} = FV(\varphi)$.

(a) For any interpretation I, show that $I \models \varphi$ iff $I \models \forall\varphi$.

(b) If ρ is an assignment, is it also the case that $I \models_\rho \varphi$ iff $I \models_\rho \forall\varphi$?

Exercise 57. Consider a first-order signature with $F_0 = \{a, b\}$, $F_2 = \{f, g\}$ and $P_1 = \{p\}$, together with an interpretation J where the domain is $\mathbb{N}$ and

$$a^J = 0 \qquad f^J = \lambda mn.m \times n \qquad p^J = \{\langle n, n\rangle \mid n \in \mathbb{N}\}$$
$$b^J = 1 \qquad g^J = \lambda mn.m + n$$

Which of the following formulas are true in J?

(a) $\exists x \forall y.p(x, y)$

(b) $\forall xy \exists z.p(g(x, z), y)$

(c) $\forall xy.(p(f(x, y), a) \to (p(x, a) \vee p(y, a)))$

(d) $\forall x \exists y.(p(x, g(y, y)) \vee p(x, g(g(y, y), b)))$

Exercise 58. Consider a first-order signature with $P_1 = \{p\}$, $P_2 = \{q\}$ and $P_3 = \{r\}$. Let I be the interpretation with domain $\mathbb{Z}$ such that

$$p^I = \mathbb{N} \qquad q^I = \{\langle i, j\rangle \mid |i| = |j|\} \qquad r^I = \{\langle i, j, k\rangle \mid ij = ik\}$$

Check whether $I \models \forall xyz.(r(x, y, z) \to ((p(y) \leftrightarrow p(z)) \wedge q(y, z)))$

Exercise 59. Consider a first-order signature with $P_1 = \{p, q\}$. Find interpretations that make the following formulas false.

(a) $((\forall x.p(x)) \to (\forall x.q(x))) \to \forall x.(p(x) \to q(x))$

(b) $(\forall x.(p(x) \vee q(x))) \to ((\forall x.p(x)) \vee (\forall x.q(x)))$

Exercise 60. Consider a first-order signature where $P_2 = \{t\}$ and the set of formulas

$$\Gamma = \{\forall x.\neg t(x, x), \forall xyz.((t(x, y) \wedge t(y, z)) \to t(x, z)), \forall x \exists y.t(x, y)\}$$

(a) Show that Γ is contradictory in every interpretation with a finite domain.

(b) Find an interpretation I such that $I \models \Gamma$.

Exercise 61. Consider a first-order signature where $P_2 = \{t\}$ and the set of formulas

$$\Gamma = \{\forall xy.(t(x,y) \to (\exists z.(t(x,z) \land t(z,y)))), \forall x \exists y.t(x,y), \forall x.\neg t(x,x)\}$$

Find an interpretation J with domain $\{1,2,3\}$ such that $J \models \Gamma$.

Exercise 62. Consider a first-order signature with $F_1 = \{f,g\}$, $P_1 = \{p,q\}$ and $P_2 = \{t\}$. Using the semantics of first-order logic, decide whether the following formulas are valid.

(a) $((\forall x.p(f(x))) \land (\forall x.p(g(x)))) \to \exists x.(p(f(x)) \land p(g(x)))$

(b) $(\exists yz \forall x.((p(x) \to q(y)) \land (q(z) \to p(x)))) \to \forall x \exists y.(p(x) \leftrightarrow q(y))$

(c) $(\exists y.p(f(y))) \to ((\forall y \exists x.p(f(x)) \to \neg p(y)) \to \exists x.\neg p(f(x)))$

(d) $(\forall x.t(x,f(x))) \to ((\forall xy.p(x) \land t(x,y) \to p(y)) \to (p(x) \to p(f(f(x)))))$

(e) $(\forall x.p(x) \to t(x,f(x))) \to ((\forall y.t(y,f(y)) \to t(f(y),y)) \to (\exists x.((t(f(x),x) \to p(x)) \to p(f(x)))))$

Exercise 63. Show that $\forall x$ is an S5-modality, i.e., that it satisfies the modal axioms K, T, 4 and 5,

(a) directly from the semantics;

(b) using the tableaux calculus for first-order logic;

(c) using the Hilbert calculus for first-order logic.

Exercise 64.

(a) Show that $\{\exists y \forall x \varphi\} \models \forall x \exists y \varphi$ for any formula φ and variables x and y.

(b) Find a formula φ for which $\{\forall x \exists y \varphi\} \not\models \exists y \forall y \varphi$.

Exercise 65. Consider a first-order signature with $P_1 = \{p,q\}$ and $P_2 = \{r\}$. Using the semantics of first-order logic, decide which of the following entailments are valid.

(a) $\{p(x) \to q(y), p(a)\} \models \forall z.q(z)$

(b) $\{\forall xy.(p(x,y) \vee q(x)), \neg q(a), \forall z.\neg p(z,b)\} \models \bot$

(c) $\models (\forall y \exists x.r(x,y)) \to (\exists x \forall y.r(x,y))$

(d) $\{\forall xy.(r(x,y) \to r(y,x)), \forall xyz.((r(x,y) \wedge r(y,z)) \to r(x,z))\} \models \forall xy.(r(x,y) \to r(x,x))$

(e) $\{\forall xyz.(r(x,y) \to (r(x,z) \to r(y,z))), \exists w.r(w,a)\} \models r(a,a)$

(f) $\{\forall xyz.(r(x,y) \to (r(x,z) \vee r(z,y))), \forall x.r(x,x), \forall xy.(r(x,y) \vee \neg r(y,x))\} \models \forall xy.r(x,y)$

(g) $\{\forall x.\neg r(x,x), \forall xyz.((r(x,y) \wedge r(z,x)) \to r(z,y))\} \models \forall xy.(\neg r(x,y) \vee \neg r(y,x))$

Exercise 66. For each of the formulas in Exercise 62, decide their validity using the tableau calculus for first-order logic.

Exercise 67. Consider again the entailments from Exercise 65. Decide which ones are valid using the tableau calculus for first-order logic.

Exercise 68. Prove each of the valid formulas in Exercise 62 using the Hilbert calculus for first-order logic.

Exercise 69. Prove each of the valid entailments in Exercise 65 using the Hilbert calculus for first-order logic.

Exercise 70. Using the Peano axioms, show that the following are theorems of arithmetic.

(a) $\forall xy.(x \approx y \to x'' \approx y'')$

(b) $\forall xy.(x''' \approx y' \to \neg x' \approx y)$

(c) $\forall xy.(x + y \approx 0 \to (x \approx 0 \wedge y \approx 0))$

Exercise 71. Write a formula that represents the quaternary relation R such that $R(m,n,q,r)$ holds iff dividing m by n returns q with remainder r.
