Part I

First-order Logic

This part covers the metatheory of first-order logic through completeness. Currently it does not rely on a separate treatment of propositional logic; everything is proved. The source files will exclude the material on quantifiers (and replace "structure" with "valuation", $\mathfrak{M}$ with $\mathfrak{v}$, etc.) if the "FOL" tag is false. In fact, most of the material in the part on propositional logic is simply the first-order material with the "FOL" tag turned off.

If the part on propositional logic is included, this results in a lot of repetition. It is planned, however, to make it possible to let this part take into account the material on propositional logic (and exclude the material already covered, as well as shorten proofs with references to the respective places in the propositional part).

Chapter 1

Introduction to First-Order Logic

1.1 First-Order Logic

fol:int:fol:sec You are probably familiar with first-order logic from your first introduction to formal logic.[1] You may know it as "quantificational logic" or "predicate logic." First-order logic, first of all, is a formal language. That means, it has a certain vocabulary, and its expressions are strings from this vocabulary. But not every string is permitted. There are different kinds of permitted expressions: terms, formulas, and sentences. We are mainly interested in sentences of first-order logic: they provide us with a formal analogue of sentences of English, and about them we can ask the questions a logician typically is interested in. For instance:

- Does ψ follow from φ logically?
- Is φ logically true, logically false, or contingent?
- Are φ and ψ equivalent?

These questions are primarily questions about the "meaning" of sentences of first-order logic. For instance, a philosopher would analyze the question of whether ψ follows logically from φ as asking: is there a case where φ is true but ψ is false (ψ doesn't follow from φ), or does every case that makes φ true also make ψ true (ψ does follow from φ)? But we haven't been told yet what a "case" is—that is the job of *semantics*. The semantics of first-order logic provides a mathematically precise model of the philosopher's intuitive idea of "case," and also—and this is important—of what it is for a sentence φ to be *true in* a case. We call the mathematically precise model that we will develop a structure. The relation which makes "true in" precise, is called the relation of *satisfaction*. So what we will define is "φ is satisfied in $\mathfrak{M}$" (in symbols: $\mathfrak{M} \vDash \varphi$) for sentences φ and structures $\mathfrak{M}$. Once this is done, we can also give precise

[1] In fact, we more or less assume you are! If you're not, you could review a more elementary textbook, such as *forall x* (Magnus et al., 2021).

definitions of the other semantical terms such as "follows from" or "is logically true." These definitions will make it possible to settle, again with mathematical precision, whether, e.g., $\forall x\,(\varphi(x) \rightarrow \psi(x)), \exists x\,\varphi(x) \vDash \exists x\,\psi(x)$. The answer will, of course, be "yes." If you've already been trained to symbolize sentences of English in first-order logic, you will recognize this as, e.g., the symbolizations of, say, "All ants are insects, there are ants, therefore there are insects." That is obviously a valid argument, and so our mathematical model of "follows from" for our formal language should give the same answer.

Another topic you probably remember from your first introduction to formal logic is that there are *derivations*. If you have taken a first formal logic course, your instructor will have made you practice finding such derivations, perhaps even a derivation that shows that the above entailment holds. There are many different ways to give derivations: you may have done something called "natural deduction" or "truth trees," but there are many others. The purpose of derivation systems is to provide tools using which the logicians' questions above can be answered: e.g., a natural deduction derivation in which $\forall x\,(\varphi(x) \rightarrow \psi(x))$ and $\exists x\,\varphi(x)$ are premises and $\exists x\,\psi(x)$ is the conclusion (last line) *verifies* that $\exists x\,\psi(x)$ logically follows from $\forall x\,(\varphi(x) \rightarrow \psi(x))$ and $\exists x\,\varphi(x)$.

But why is that? On the face of it, derivation systems have nothing to do with semantics: giving a formal derivation merely involves arranging symbols in certain rule-governed ways; they don't mention "cases" or "true in" at all. The connection between derivation systems and semantics has to be established by a meta-logical investigation. What's needed is a mathematical proof, e.g., that a formal derivation of $\exists x\,\psi(x)$ from premises $\forall x\,(\varphi(x) \rightarrow \psi(x))$ and $\exists x\,\varphi(x)$ is possible, if, and only if, $\forall x\,(\varphi(x) \rightarrow \psi(x))$ and $\exists x\,\varphi(x)$ together entail $\exists x\,\psi(x)$. Before this can be done, however, a lot of painstaking work has to be carried out to get the definitions of syntax and semantics correct.

1.2 Syntax

We first must make precise what strings of symbols count as sentences of first-order logic. We'll do this later; for now we'll just proceed by example. The basic building blocks—the vocabulary—of first-order logic divides into two parts. The first part is the symbols we use to say specific things or to pick out specific things. We pick out things using constant symbols, and we say stuff about the things we pick out using predicate symbols. E.g, we might use a as a constant symbol to pick out a single thing, and then say something about it using the sentence $P(a)$. If you have meanings for "a" and "P" in mind, you can read $P(a)$ as a sentence of English (and you probably have done so when you first learned formal logic). Once you have such simple sentences of first-order logic, you can build more complex ones using the second part of the vocabulary: the logical symbols (connectives and quantifiers). So, for instance, we can form expressions like $(P(a) \wedge Q(b))$ or $\exists x\,P(x)$. fol:int:syn:sec

In order to provide the precise definitions of semantics and the rules of our derivation systems required for rigorous meta-logical study, we first of all have to give a precise definition of what counts as a sentence of first-order logic. The basic idea is easy enough to understand: there are some simple sentences we can form from just predicate symbols and constant symbols, such as $P(a)$. And then from these we form more complex ones using the connectives and quantifiers. But what exactly are the rules by which we are allowed to form more complex sentences? These must be specified, otherwise we have not defined "sentence of first-order logic" precisely enough. There are a few issues. The first one is to get the right strings to count as sentences. The second one is to do this in such a way that we can give mathematical proofs about *all* sentences. Finally, we'll have to also give precise definitions of some rudimentary operations with sentences, such as "replace every x in φ by b." The trouble is that the quantifiers and variables we have in first-order logic make it not entirely obvious how this should be done. E.g., should $\exists x\, P(a)$ count as a sentence? What about $\exists x\, \exists x\, P(x)$? What should the result of "replace x by b in $(P(x) \land \exists x\, P(x))$" be?

1.3 Formulas

Here is the approach we will use to rigorously specify sentences of first-order logic and to deal with the issues arising from the use of variables. We first define a *different* set of expressions: formulas. Once we've done that, we can consider the role variables play in them—and on the basis of some other ideas, namely those of "free" and "bound" variables, we can define what a sentence is (namely, a formula without free variables). We do this not just because it makes the definition of "sentence" more manageable, but also because it will be crucial to the way we define the semantic notion of satisfaction.

Let's define "formula" for a simple first-order language, one containing only a single predicate symbol P and a single constant symbol a, and only the logical symbols $\neg$, $\land$, and $\exists$. Our full definitions will be much more general: we'll allow infinitely many predicate symbols and constant symbols. In fact, we will also consider function symbols which can be combined with constant symbols and variables to form "terms." For now, a and the variables will be our only terms. We do need infinitely many variables. We'll officially use the symbols v_0, v_1, $\dots$, as variables.

Definition 1.1. The set of *formulas* Frm is defined as follows:

fol:int:fml: fmls-atom

1. $P(a)$ and $P(v_i)$ are formulas ($i \in \mathbb{N}$).

fol:int:fml: fmls-not

2. If φ is a formula, then $\neg\varphi$ is formula.
3. If φ and ψ are formulas, then $(\varphi \land \psi)$ is a formula.

fol:int:fml: fmls-ex

4. If φ is a formula and x is a variable, then $\exists x\, \varphi$ is a formula.

fol:int:fml: fmls-limit

5. Nothing else is a formula.

(1) tells us that $P(a)$ and $P(v_i)$ are formulas, for any $i \in \mathbb{N}$. These are the so-called *atomic* formulas. They give us something to start from. The other clauses give us ways of forming new formulas from ones we have already formed. So for instance, by (2), we get that $\neg P(v_2)$ is a formula, since $P(v_2)$ is already a formula by (1). Then, by (4), we get that $\exists v_2 \, \neg P(v_2)$ is another formula, and so on. (5) tells us that *only* strings we can form in this way count as formulas. In particular, $\exists v_0 \, P(a)$ and $\exists v_0 \, \exists v_0 \, P(a)$ *do* count as formulas, and $(\neg P(a))$ does not, because of the extraneous outer parentheses.

This way of defining formulas is called an *inductive definition*, and it allows us to prove things about formulas using a version of proof by induction called *structural induction*. These are discussed in a general way in **??** and **??**, which you should review before delving into the proofs later on. Basically, the idea is that if you want to give a proof that something is true for all formulas, you show first that it is true for the atomic formulas, and then that *if* it's true for any formula φ (and ψ), it's *also* true for $\neg\varphi$, $(\varphi \wedge \psi)$, and $\exists x \, \varphi$. For instance, this proves that it's true for $\exists v_2 \, \neg P(v_2)$: from the first part you know that it's true for the atomic formula $P(v_2)$. Then you get that it's true for $\neg P(v_2)$ by the second part, and then again that it's true for $\exists v_2 \, \neg P(v_2)$ itself. Since all formulas are inductively generated from atomic formulas, this works for any of them.

1.4 Satisfaction

We can already skip ahead to the semantics of first-order logic once we know what formulas are: here, the basic definition is that of a structure. For our simple language, a structure $\mathfrak{M}$ has just three components: a non-empty set $|\mathfrak{M}|$ called the *domain*, what a picks out in $\mathfrak{M}$, and what P is true of in $\mathfrak{M}$. The object picked out by a is denoted $a^{\mathfrak{M}}$ and the set of things P is true of by $P^{\mathfrak{M}}$. A structure $\mathfrak{M}$ consists of just these three things: $|\mathfrak{M}|$, $a^{\mathfrak{M}} \in |\mathfrak{M}|$ and $P^{\mathfrak{M}} \subseteq |\mathfrak{M}|$. The general case will be more complicated, since there will be many predicate symbols and constant symbols, the constant symbols can have more than one place, and there will also be function symbols. fol:int:sat:sec

This is enough to give a definition of satisfaction for formulas that don't contain variables. The idea is to give an inductive definition that mirrors the way we have defined formulas. We specify when an atomic formula is satisfied in $\mathfrak{M}$, and then when, e.g., $\neg\varphi$ is satisfied in $\mathfrak{M}$ on the basis of whether or not φ is satisfied in $\mathfrak{M}$. E.g., we could define:

1. $P(a)$ is satisfied in $\mathfrak{M}$ iff $a^{\mathfrak{M}} \in P^{\mathfrak{M}}$.

2. $\neg\varphi$ is satisfied in $\mathfrak{M}$ iff φ is not satisfied in $\mathfrak{M}$.

3. $(\varphi \wedge \psi)$ is satisfied in $\mathfrak{M}$ iff φ is satisfied in $\mathfrak{M}$, and ψ is satisfied in $\mathfrak{M}$ as well.

Let's say that $|\mathfrak{M}| = \{0, 1, 2\}$, $a^{\mathfrak{M}} = 1$, and $P^{\mathfrak{M}} = \{1, 2\}$. This definition would tell us that $P(a)$ is satisfied in $\mathfrak{M}$ (since $a^{\mathfrak{M}} = 1 \in \{1, 2\} = P^{\mathfrak{M}}$). It tells us further that $\neg P(a)$ is not satisfied in $\mathfrak{M}$, and that in turn $\neg\neg P(a)$ is and $(\neg P(a) \land P(a))$ is not satisfied, and so on.

The trouble comes when we want to give a definition for the quantifiers: we'd like to say something like, "$\exists v_0\, P(v_0)$ is satisfied iff $P(v_0)$ is satisfied." But the structure $\mathfrak{M}$ doesn't tell us what to do about variables. What we actually want to say is that $P(v_0)$ is satisfied *for some value of* v_0. To make this precise we need a way to assign elements of $|\mathfrak{M}|$ not just to a but also to v_0. To this end, we introduce variable *assignments*. A variable assignment is simply a function s that maps variables to elements of $|\mathfrak{M}|$ (in our example, to one of 1, 2, or 3). Since we don't know beforehand which variables might appear in a formula we can't limit which variables s assigns values to. The simple solution is to require that s assigns values to *all* variables v_0, v_1, ... We'll just use only the ones we need.

Instead of defining satisfaction of formulas just relative to a structure, we'll define it relative to a structure $\mathfrak{M}$ *and* a variable assignment s, and write $\mathfrak{M}, s \vDash \varphi$ for short. Our definition will now include an additional clause to deal with atomic formulas containing variables:

1. $\mathfrak{M}, s \vDash P(a)$ iff $a^{\mathfrak{M}} \in P^{\mathfrak{M}}$.
2. $\mathfrak{M}, s \vDash P(v_i)$ iff $s(v_i) \in P^{\mathfrak{M}}$.
3. $\mathfrak{M}, s \vDash \neg\varphi$ iff not $\mathfrak{M}, s \vDash \varphi$.
4. $\mathfrak{M}, s \vDash (\varphi \land \psi)$ iff $\mathfrak{M}, s \vDash \varphi$ and $\mathfrak{M}, s \vDash \psi$.

Ok, this solves one problem: we can now say when $\mathfrak{M}$ satisfies $P(v_0)$ for the value $s(v_0)$. To get the definition right for $\exists v_0\, P(v_0)$ we have to do one more thing: We want to have that $\mathfrak{M}, s \vDash \exists v_0\, P(v_0)$ iff $\mathfrak{M}, s' \vDash P(v_0)$ for *some* way s' of assigning a value to v_0. But the value assigned to v_0 does not necessarily have to be the value that $s(v_0)$ picks out. We'll introduce a notation for that: if $m \in |\mathfrak{M}|$, then we let $s[m/v_0]$ be the assignment that is just like s (for all variables other than v_0), except to v_0 it assigns m. Now our definition can be:

5. $\mathfrak{M}, s \vDash \exists v_i\, \varphi$ iff $\mathfrak{M}, s[m/v_i] \vDash \varphi$ for some $m \in |\mathfrak{M}|$.

Does it work out? Let's say we let $s(v_i) = 0$ for all $i \in \mathbb{N}$. $\mathfrak{M}, s \vDash \exists v_0\, P(v_0)$ iff there is an $m \in |\mathfrak{M}|$ so that $\mathfrak{M}, s[m/v_0] \vDash P(v_0)$. And there is: we can choose $m = 1$ or $m = 2$. Note that this is true even if the value $s(v_0)$ assigned to v_0 by s itself—in this case, 0—doesn't do the job. We have $\mathfrak{M}, s[1/v_0] \vDash P(v_0)$ but not $\mathfrak{M}, s \vDash P(v_0)$.

If this looks confusing and cumbersome: it is. But the added complexity is required to give a precise, inductive definition of satisfaction for all formulas, and we need something like it to precisely define the semantic notions. There are other ways of doing it, but they are all equally (in)elegant.

1.5 Sentences

Ok, now we have a (sketch of a) definition of satisfaction ("true in") for structures and formulas. But it needs this additional bit—a variable assignment—and what we wanted is a definition of sentences. How do we get rid of assignments, and what are sentences? fol:int:snt:sec

You probably remember a discussion in your first introduction to formal logic about the relation between variables and quantifiers. A quantifier is always followed by a variable, and then in the part of the sentence to which that quantifier applies (its "scope"), we understand that the variable is "bound" by that quantifier. In formulas it was not required that every variable has a matching quantifier, and variables without matching quantifiers are "free" or "unbound." We will take sentences to be all those formulas that have no free variables.

Again, the intuitive idea of when an occurrence of a variable in a formula φ is bound, which quantifier binds it, and when it is free, is not difficult to get. You may have learned a method for testing this, perhaps involving counting parentheses. We have to insist on a precise definition—and because we have defined formulas by induction, we can give a definition of the free and bound occurrences of a variable x in a formula φ also by induction. E.g., it might look like this for our simplified language:

1. If φ is atomic, all occurrences of x in it are free (that is, the occurrence of x in $P(x)$ is free).

2. If φ is of the form $\lnot\psi$, then an occurrence of x in $\lnot\psi$ is free iff the corresponding occurrence of x is free in ψ (that is, the free occurrences of variables in ψ are exactly the corresponding occurrences in $\lnot\psi$).

3. If φ is of the form $(\psi \land \chi)$, then an occurrence of x in $(\psi \land \chi)$ is free iff the corresponding occurrence of x is free in ψ or in χ.

4. If φ is of the form $\exists x\, \psi$, then no occurrence of x in φ is free; if it is of the form $\exists y\, \psi$ where y is a different variable than x, then an occurrence of x in $\exists y\, \psi$ is free iff the corresponding occurrence of x is free in ψ.

Once we have a precise definition of free and bound occurrences of variables, we can simply say: a sentence is any formula without free occurrences of variables.

1.6 Semantic Notions

We mentioned above that when we consider whether $\mathfrak{M}, s \vDash \varphi$ holds, we (for convenience) let s assign values to all variables, but only the values it assigns to variables in φ are used. In fact, it's only the values of *free* variables in φ that matter. Of course, because we're careful, we are going to prove this fact. Since sentences have no free variables, s doesn't matter at all when it comes to fol:int:sem:sec whether or not they are satisfied in a structure. So, when φ is a sentence we can define $\mathfrak{M} \vDash \varphi$ to mean "$\mathfrak{M}, s \vDash \varphi$ for all s," which as it happens is true iff $\mathfrak{M}, s \vDash \varphi$ for at least one s. We need to introduce variable assignments to get a working definition of satisfaction for formulas, but for sentences, satisfaction is independent of the variable assignments.

Once we have a definition of "$\mathfrak{M} \vDash \varphi$," we know what "case" and "true in" mean as far as sentences of first-order logic are concerned. On the basis of the definition of $\mathfrak{M} \vDash \varphi$ for sentences we can then define the basic semantic notions of validity, entailment, and satisfiability. A sentence is valid, $\vDash \varphi$, if every structure satisfies it. It is entailed by a set of sentences, $\Gamma \vDash \varphi$, if every structure that satisfies all the sentences in Γ also satisfies φ. And a set of sentences is satisfiable if some structure satisfies all sentences in it at the same time.

Because formulas are inductively defined, and satisfaction is in turn defined by induction on the structure of formulas, we can use induction to prove properties of our semantics and to relate the semantic notions defined. We'll collect and prove some of these properties, partly because they are individually interesting, but mainly because many of them will come in handy when we go on to investigate the relation between semantics and derivation systems. In order to do so, we'll also have to define (precisely, i.e., by induction) some syntactic notions and operations we haven't mentioned yet.

1.7 Substitution

fol:int:sub:sec

We'll discuss an example to illustrate how things hang together, and how the development of syntax and semantics lays the foundation for our more advanced investigations later. Our derivation systems should let us derive $P(a)$ from $\forall v_0\, P(v_0)$. Maybe we even want to state this as a rule of inference. However, to do so, we must be able to state it in the most general terms: not just for P, a, and v_0, but for any formula φ, and term t, and variable x. (Recall that constant symbols are terms, but we'll consider also more complicated terms built from constant symbols and function symbols.) So we want to be able to say something like, "whenever you have derived $\forall x\, \varphi(x)$ you are justified in inferring $\varphi(t)$—the result of removing $\forall x$ and replacing x by t." But what exactly does "replacing x by t" mean? What is the relation between $\varphi(x)$ and $\varphi(t)$? Does this always work?

To make this precise, we define the operation of *substitution*. Substitution is actually tricky, because we can't just replace all x's in φ by t, and not every t can be substituted for any x. We'll deal with this, again, using inductive definitions. But once this is done, specifying an inference rule as "infer $\varphi(t)$ from $\forall x\, \varphi(x)$" becomes a precise definition. Moreover, we'll be able to show that this is a good inference rule in the sense that $\forall x\, \varphi(x)$ entails $\varphi(t)$. But to prove this, we have to again prove something that may at first glance prompt you to ask "why are we doing this?" That $\forall x\, \varphi(x)$ entails $\varphi(t)$ relies on the fact that whether or not $\mathfrak{M} \vDash \varphi(t)$ holds depends only on the value of the term t, i.e., if we let m be whatever element of $|\mathfrak{M}|$ is picked out by t, then $\mathfrak{M}, s \vDash \varphi(t)$ iff $\mathfrak{M}, s[m/x] \vDash \varphi(x)$. This holds even when t contains variables, but we'll have to be careful with how exactly we state the result.

1.8 Models and Theories

Once we've defined the syntax and semantics of first-order logic, we can get to work investigating the properties of structures and the semantic notions. We can also define derivation systems, and investigate those. For a set of sentences, we can ask: what structures make all the sentences in that set true? Given a set of sentences Γ, a structure $\mathfrak{M}$ that satisfies them is called a *model of* Γ. We might start from Γ and try to find its models—what do they look like? How big or small do they have to be? But we might also start with a single structure or collection of structures and ask: what sentences are true in them? Are there sentences that *characterize* these structures in the sense that they, and only they, are true in them? These kinds of questions are the domain of *model theory*. They also underlie the *axiomatic method*: describing a collection of structures by a set of sentences, the axioms of a theory. This is made possible by the observation that exactly those sentences entailed in first-order logic by the axioms are true in all models of the axioms. fol:int:mod:sec

As a very simple example, consider preorders. A preorder is a relation R on some set A which is both reflexive and transitive. A set A with a two-place relation $R \subseteq A \times A$ on it is exactly what we would need to give a structure for a first-order language with a single two-place relation symbol P: we would set $|\mathfrak{M}| = A$ and $\mathsf{P}^{\mathfrak{M}} = R$. Since R is a preorder, it is reflexive and transitive, and we can find a set Γ of sentences of first-order logic that say this:

$$\forall \mathsf{v}_0\, \mathsf{P}(\mathsf{v}_0, \mathsf{v}_0)$$
$$\forall \mathsf{v}_0\, \forall \mathsf{v}_1\, \forall \mathsf{v}_2\, ((\mathsf{P}(\mathsf{v}_0, \mathsf{v}_1) \land \mathsf{P}(\mathsf{v}_1, \mathsf{v}_2)) \rightarrow \mathsf{P}(\mathsf{v}_0, \mathsf{v}_2))$$

These sentences are just the symbolizations of "for any x, Rxx" (R is reflexive) and "whenever Rxy and Ryz then also Rxz" (R is transitive). We see that a structure $\mathfrak{M}$ is a model of these two sentences Γ iff R (i.e., $\mathsf{P}^{\mathfrak{M}}$), is a preorder on A (i.e., $|\mathfrak{M}|$). In other words, the models of Γ are exactly the preorders. Any property of all preorders that can be expressed in the first-order language with just P as predicate symbol (like reflexivity and transitivity above), is entailed by the two sentences in Γ and vice versa. So anything we can prove about models of Γ we have proved about all preorders.

For any particular theory and class of models (such as Γ and all preorders), there will be interesting questions about what can be expressed in the corresponding first-order language, and what cannot be expressed. There are some properties of structures that are interesting for all languages and classes of models, namely those concerning the size of the domain. One can always express, for instance, that the domain contains exactly n elements, for any $n \in \mathbb{Z}^+$. One can also express, using a set of infinitely many sentences, that the domain is infinite. But one cannot express that the domain is finite, or that the domain is non-enumerable. These results about the limitations of first-order languages are consequences of the compactness and Löwenheim-Skolem theorems.

1.9 Soundness and Completeness

fol:int:scp: sec

We'll also introduce derivation systems for first-order logic. There are many derivation systems that logicians have developed, but they all define the same derivability relation between sentences. We say that Γ *derives* φ, $\Gamma \vdash \varphi$, if there is a derivation of a certain precisely defined sort. Derivations are always finite arrangements of symbols—perhaps a list of sentences, or some more complicated structure. The purpose of derivation systems is to provide a tool to determine if a sentence is entailed by some set Γ. In order to serve that purpose, it must be true that $\Gamma \vDash \varphi$ if, and only if, $\Gamma \vdash \varphi$.

If $\Gamma \vdash \varphi$ but not $\Gamma \vDash \varphi$, our derivation system would be too strong, prove too much. The property that if $\Gamma \vdash \varphi$ then $\Gamma \vDash \varphi$ is called *soundness*, and it is a minimal requirement on any good derivation system. On the other hand, if $\Gamma \vDash \varphi$ but not $\Gamma \vdash \varphi$, then our derivation system is too weak, it doesn't prove enough. The property that if $\Gamma \vDash \varphi$ then $\Gamma \vdash \varphi$ is called *completeness*. Soundness is usually relatively easy to prove (by induction on the structure of derivations, which are inductively defined). Completeness is harder to prove.

Soundness and completeness have a number of important consequences. If a set of sentences Γ derives a contradiction (such as $\varphi \wedge \neg\varphi$) it is called *inconsistent*. Inconsistent Γs cannot have any models, they are unsatisfiable. From completeness the converse follows: any Γ that is not inconsistent—or, as we will say, *consistent*—has a model. In fact, this is equivalent to completeness, and is the form of completeness we will actually prove. It is a deep and perhaps surprising result: just because you cannot prove $\varphi \wedge \neg\varphi$ from Γ guarantees that there is a structure that is as Γ describes it. So completeness gives an answer to the question: which sets of sentences have models? Answer: all and only consistent sets do.

The soundness and completeness theorems have two important consequences: the compactness and the Löwenheim-Skolem theorem. These are important results in the theory of models, and can be used to establish many interesting results. We've already mentioned two: first-order logic cannot express that the domain of a structure is finite or that it is non-enumerable.

Historically, all of this—how to define syntax and semantics of first-order logic, how to define good derivation systems, how to prove that they are sound and complete, getting clear about what can and cannot be expressed in first-order languages—took a long time to figure out and get right. We now know how to do it, but going through all the details can still be confusing and tedious. But it's also important, because the methods developed here for the formal language of first-order logic are applied all over the place in logic, computer science, and linguistics. So working through the details pays off in the long run.

Chapter 2

Syntax of First-Order Logic

2.1 Introduction

In order to develop the theory and metatheory of first-order logic, we must first define the syntax and semantics of its expressions. The expressions of first-order logic are terms and formulas. Terms are formed from variables, constant symbols, and function symbols. Formulas, in turn, are formed from predicate symbols together with terms (these form the smallest, "atomic" formulas), and then from atomic formulas we can form more complex ones using logical connectives and quantifiers. There are many different ways to set down the formation rules; we give just one possible one. Other systems will chose different symbols, will select different sets of connectives as primitive, will use parentheses differently (or even not at all, as in the case of so-called Polish notation). What all approaches have in common, though, is that the formation rules define the set of terms and formulas *inductively*. If done properly, every expression can result essentially in only one way according to the formation rules. The inductive definition resulting in expressions that are *uniquely readable* means we can give meanings to these expressions using the same method—inductive definition. fol:syn:itx:sec

2.2 First-Order Languages

Expressions of first-order logic are built up from a basic vocabulary containing *variables*, *constant symbols*, *predicate symbols* and sometimes *function symbols*. From them, together with logical connectives, quantifiers, and punctuation symbols such as parentheses and commas, *terms* and *formulas* are formed. fol:syn:fol:sec

explanation Informally, predicate symbols are names for properties and relations, constant symbols are names for individual objects, and function symbols are names for mappings. These, except for the identity predicate $=$, are the *non-logical symbols* and together make up a language. Any first-order language $\mathcal{L}$ is determined by its non-logical symbols. In the most general case, $\mathcal{L}$ contains infinitely many symbols of each kind.

In the general case, we make use of the following symbols in first-order logic:

1. Logical symbols
   a) Logical connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\to$ (conditional), $\leftrightarrow$ (biconditional), $\forall$ (universal quantifier), $\exists$ (existential quantifier).
   b) The propositional constant for falsity $\bot$.
   c) The propositional constant for truth $\top$.
   d) The two-place identity predicate $=$.
   e) A denumerable set of variables: v_0, v_1, v_2, ...
2. Non-logical symbols, making up the *standard language* of first-order logic
   a) A denumerable set of n-place predicate symbols for each $n > 0$: A^n_0, A^n_1, A^n_2, ...
   b) A denumerable set of constant symbols: c_0, c_1, c_2, ...
   c) A denumerable set of n-place function symbols for each $n > 0$: f^n_0, f^n_1, f^n_2, ...
3. Punctuation marks: (,), and the comma.

Most of our definitions and results will be formulated for the full standard language of first-order logic. However, depending on the application, we may also restrict the language to only a few predicate symbols, constant symbols, and function symbols.

Example 2.1. The language $\mathcal{L}_A$ of arithmetic contains a single two-place predicate symbol $<$, a single constant symbol o, one one-place function symbol $\prime$, and two two-place function symbols $+$ and $\times$.

Example 2.2. The language of set theory $\mathcal{L}_Z$ contains only the single two-place predicate symbol $\in$.

Example 2.3. The language of orders $\mathcal{L}_\leq$ contains only the two-place predicate symbol $\leq$.

Again, these are conventions: officially, these are just aliases, e.g., $<$, $\in$, and $\leq$ are aliases for A^2_0, o for c_0, $\prime$ for f^1_0, $+$ for f^2_0, $\times$ for f^2_1.

.

You may be familiar with different terminology and symbols than the ones we use above. Logic texts (and teachers) commonly use $\sim$, $\neg$, or ! for "negation", $\wedge$, $\cdot$, or & for "conjunction". Commonly used symbols for the "conditional" or "implication" are $\to$, $\Rightarrow$, and $\supset$. Symbols for "biconditional," "bi-implication," or "(material) equivalence" are $\leftrightarrow$, $\Leftrightarrow$, and $\equiv$. The $\bot$ symbol is variously called "falsity," "falsum,", "absurdity," or "bottom." The $\top$ symbol is variously called "truth," "verum," or "top." intro

It is conventional to use lower case letters (e.g., a, b, c) from the beginning of the Latin alphabet for constant symbols (sometimes called names), and lower case letters from the end (e.g., x, y, z) for variables. Quantifiers combine with variables, e.g., x; notational variations include $\forall x$, $(\forall x)$, (x), Πx, $\bigwedge_x$ for the universal quantifier and $\exists x$, $(\exists x)$, (Ex), Σx, $\bigvee_x$ for the existential quantifier.

explanation We might treat all the propositional operators and both quantifiers as primitive symbols of the language. We might instead choose a smaller stock of primitive symbols and treat the other logical operators as defined. "Truth functionally complete" sets of Boolean operators include $\{\neg, \lor\}$, $\{\neg, \land\}$, and $\{\neg, \rightarrow\}$—these can be combined with either quantifier for an expressively complete first-order language.

You may be familiar with two other logical operators: the Sheffer stroke $|$ (named after Henry Sheffer), and Peirce's arrow $\downarrow$, also known as Quine's dagger. When given their usual readings of "nand" and "nor" (respectively), these operators are truth functionally complete by themselves.

2.3 Terms and Formulas

Once a first-order language $\mathcal{L}$ is given, we can define expressions built up from the basic vocabulary of $\mathcal{L}$. These include in particular *terms* and *formulas*. fol:syn:frm:sec

Definition 2.4 (Terms). The set of *terms* $\mathrm{Trm}(\mathcal{L})$ of $\mathcal{L}$ is defined inductively by: fol:syn:frm:defn:terms

1. Every variable is a term.
2. Every constant symbol of $\mathcal{L}$ is a term.
3. If f is an n-place function symbol and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.
4. Nothing else is a term.

A term containing no variables is a *closed term*.

explanation The constant symbols appear in our specification of the language and the terms as a separate category of symbols, but they could instead have been included as zero-place function symbols. We could then do without the second clause in the definition of terms. We just have to understand $f(t_1, \ldots, t_n)$ as just f by itself if $n = 0$.

Definition 2.5 (Formulas). The set of *formulas* $\mathrm{Frm}(\mathcal{L})$ of the language $\mathcal{L}$ is defined inductively as follows: fol:syn:frm:defn:formulas

1. $\bot$ is an atomic formula.
2. $\top$ is an atomic formula.
3. If R is an n-place predicate symbol of $\mathcal{L}$ and $t_1, \ldots, t_n$ are terms of $\mathcal{L}$, then $R(t_1, \ldots, t_n)$ is an atomic formula.

4. If t_1 and t_2 are terms of $\mathcal{L}$, then $=(t_1, t_2)$ is an atomic formula.
5. If φ is a formula, then $\neg\varphi$ is formula.
6. If φ and ψ are formulas, then $(\varphi \wedge \psi)$ is a formula.
7. If φ and ψ are formulas, then $(\varphi \vee \psi)$ is a formula.
8. If φ and ψ are formulas, then $(\varphi \rightarrow \psi)$ is a formula.
9. If φ and ψ are formulas, then $(\varphi \leftrightarrow \psi)$ is a formula.
10. If φ is a formula and x is a variable, then $\forall x\, \varphi$ is a formula.
11. If φ is a formula and x is a variable, then $\exists x\, \varphi$ is a formula.
12. Nothing else is a formula.

The definitions of the set of terms and that of formulas are *inductive definitions*. Essentially, we construct the set of formulas in infinitely many stages. In the initial stage, we pronounce all atomic formulas to be formulas; this corresponds to the first few cases of the definition, i.e., the cases for $\top$, $\bot$, $R(t_1, \dots, t_n)$ and $=(t_1, t_2)$. "Atomic formula" thus means any formula of this form. explanation

The other cases of the definition give rules for constructing new formulas out of formulas already constructed. At the second stage, we can use them to construct formulas out of atomic formulas. At the third stage, we construct new formulas from the atomic formulas and those obtained in the second stage, and so on. A formula is anything that is eventually constructed at such a stage, and nothing else.

By convention, we write $=$ between its arguments and leave out the parentheses: $t_1 = t_2$ is an abbreviation for $=(t_1, t_2)$. Moreover, $\neg{=}(t_1, t_2)$ is abbreviated as $t_1 \neq t_2$. When writing a formula $(\psi * \chi)$ constructed from ψ, χ using a two-place connective $*$, we will often leave out the outermost pair of parentheses and write simply $\psi * \chi$.

Some logic texts require that the variable x must occur in φ in order for $\exists x\, \varphi$ and $\forall x\, \varphi$ to count as formulas. Nothing bad happens if you don't require this, and it makes things easier. intro

If we work in a language for a specific application, we will often write two-place predicate symbols and function symbols between the respective terms, e.g., $t_1 < t_2$ and $(t_1 + t_2)$ in the language of arithmetic and $t_1 \in t_2$ in the language of set theory. The successor function in the language of arithmetic is even written conventionally *after* its argument: t'. Officially, however, these are just conventional abbreviations for $A^2_0(t_1, t_2)$, $f^2_0(t_1, t_2)$, $A^2_0(t_1, t_2)$ and $f^1_0(t)$, respectively.

Definition 2.6 (Syntactic identity). The symbol $\equiv$ expresses syntactic identity between strings of symbols, i.e., $\varphi \equiv \psi$ iff φ and ψ are strings of symbols of the same length and which contain the same symbol in each place.

The $\equiv$ symbol may be flanked by strings obtained by concatenation, e.g., $\varphi \equiv (\psi \lor \chi)$ means: the string of symbols φ is the same string as the one obtained by concatenating an opening parenthesis, the string ψ, the $\lor$ symbol, the string χ, and a closing parenthesis, in this order. If this is the case, then we know that the first symbol of φ is an opening parenthesis, φ contains ψ as a substring (starting at the second symbol), that substring is followed by $\lor$, etc.

As terms and formulas are built up from basic elements via inductive definitions, we can use the following induction principles to prove things about them.

Lemma 2.7 (*Principle of induction on terms*). *Let $\mathcal{L}$ be a first-order language. If some property P holds in all of the following cases, then $P(t)$ for every $t \in \mathrm{Trm}(\mathcal{L})$.* fol:syn:frm: lem:trmind

1. *$P(v)$ for every variable v,*
2. *$P(a)$ for every constant symbol a of $\mathcal{L}$,*
3. *If $t_1, \dots, t_n \in \mathrm{Trm}(\mathcal{L})$, f is an n-place function symbol of $\mathcal{L}$, and $P(t_1), \dots, P(t_n)$, then $P(f(t_1, \dots, t_n))$.*

Problem 2.1. Prove Lemma 2.7.

Lemma 2.8 (*Principle of induction on formulas*). *Let $\mathcal{L}$ be a first-order language. If some property P holds for all the atomic formulas and is such that* fol:syn:frm: thm:frmind

1. *φ is an atomic formula.*
2. *it holds for $\lnot\varphi$ whenever it holds for φ;*
3. *it holds for $(\varphi \land \psi)$ whenever it holds for φ and ψ;*
4. *it holds for $(\varphi \lor \psi)$ whenever it holds for φ and ψ;*
5. *it holds for $(\varphi \rightarrow \psi)$ whenever it holds for φ and ψ;*
6. *it holds for $(\varphi \leftrightarrow \psi)$ whenever it holds for φ and ψ;*
7. *it holds for $\exists x \varphi$ whenever it holds for φ;*
8. *it holds for $\forall x \varphi$ whenever it holds for φ;*

then P holds for all formulas $\varphi \in \mathrm{Frm}(\mathcal{L})$.

2.4 Unique Readability

fol:syn:unq: sec

The way we defined formulas guarantees that every formula has a *unique reading*, i.e., there is essentially only one way of constructing it according to our formation rules for formulas and only one way of "interpreting" it. If this were not so, we would have ambiguous formulas, i.e., formulas that have more than one reading or intepretation—and that is clearly something we want to avoid. But more importantly, without this property, most of the definitions and proofs we are going to give will not go through. explanation

Perhaps the best way to make this clear is to see what would happen if we had given bad rules for forming formulas that would not guarantee unique readability. For instance, we could have forgotten the parentheses in the formation rules for connectives, e.g., we might have allowed this:

> If φ and ψ are formulas, then so is $\varphi \rightarrow \psi$.

Starting from an atomic formula θ, this would allow us to form $\theta \rightarrow \theta$. From this, together with θ, we would get $\theta \rightarrow \theta \rightarrow \theta$. But there are two ways to do this:

1. We take θ to be φ and $\theta \rightarrow \theta$ to be ψ.
2. We take φ to be $\theta \rightarrow \theta$ and ψ is θ.

Correspondingly, there are two ways to "read" the formula $\theta \rightarrow \theta \rightarrow \theta$. It is of the form $\psi \rightarrow \chi$ where ψ is θ and χ is $\theta \rightarrow \theta$, but *it is also* of the form $\psi \rightarrow \chi$ with ψ being $\theta \rightarrow \theta$ and χ being θ.

If this happens, our definitions will not always work. For instance, when we define the main operator of a formula, we say: in a formula of the form $\psi \rightarrow \chi$, the main operator is the indicated occurrence of $\rightarrow$. But if we can match the formula $\theta \rightarrow \theta \rightarrow \theta$ with $\psi \rightarrow \chi$ in the two different ways mentioned above, then in one case we get the first occurrence of $\rightarrow$ as the main operator, and in the second case the second occurrence. But we intend the main operator to be a *function* of the formula, i.e., every formula must have exactly one main operator occurrence.

Lemma 2.9. *The number of left and right parentheses in a formula φ are equal.*

Proof. We prove this by induction on the way φ is constructed. This requires two things: (a) We have to prove first that all atomic formulas have the property in question (the induction basis). (b) Then we have to prove that when we construct new formulas out of given formulas, the new formulas have the property provided the old ones do.

Let $l(\varphi)$ be the number of left parentheses, and $r(\varphi)$ the number of right parentheses in φ, and $l(t)$ and $r(t)$ similarly the number of left and right parentheses in a term t.

Problem 2.2. Prove that for any term t, $l(t) = r(t)$.

1. $\varphi \equiv \bot$: φ has 0 left and 0 right parentheses.
2. $\varphi \equiv \top$: φ has 0 left and 0 right parentheses.
3. $\varphi \equiv R(t_1, \dots, t_n)$: $l(\varphi) = 1 + l(t_1) + \dots + l(t_n) = 1 + r(t_1) + \dots + r(t_n) = r(\varphi)$. Here we make use of the fact, left as an exercise, that $l(t) = r(t)$ for any term t.
4. $\varphi \equiv t_1 = t_2$: $l(\varphi) = l(t_1) + l(t_2) = r(t_1) + r(t_2) = r(\varphi)$.
5. $\varphi \equiv \neg\psi$: By induction hypothesis, $l(\psi) = r(\psi)$. Thus $l(\varphi) = l(\psi) = r(\psi) = r(\varphi)$.
6. $\varphi \equiv (\psi * \chi)$: By induction hypothesis, $l(\psi) = r(\psi)$ and $l(\chi) = r(\chi)$. Thus $l(\varphi) = 1 + l(\psi) + l(\chi) = 1 + r(\psi) + r(\chi) = r(\varphi)$.
7. $\varphi \equiv \forall x\, \psi$: By induction hypothesis, $l(\psi) = r(\psi)$. Thus, $l(\varphi) = l(\psi) = r(\psi) = r(\varphi)$.
8. $\varphi \equiv \exists x\, \psi$: Similarly. □

Definition 2.10 (Proper prefix). A string of symbols ψ is a *proper prefix* of a string of symbols φ if concatenating ψ and a non-empty string of symbols yields φ.

Lemma 2.11. *If φ is a formula, and ψ is a proper prefix of φ, then ψ is not a formula.* *fol:syn:unq: lem:no-prefix*

Proof. Exercise. □

Problem 2.3. Prove Lemma 2.11.

Proposition 2.12. *If φ is an atomic formula, then it satisfies one, and only one of the following conditions.* *fol:syn:unq: prop:unique-atomic*

1. $\varphi \equiv \bot$.
2. $\varphi \equiv \top$.
3. $\varphi \equiv R(t_1, \dots, t_n)$ *where R is an n-place predicate symbol, t_1, …, t_n are terms, and each of R, t_1, …, t_n is uniquely determined.*
4. $\varphi \equiv t_1 = t_2$ *where t_1 and t_2 are uniquely determined terms.*

Proof. Exercise. □

Problem 2.4. Prove Proposition 2.12 (Hint: Formulate and prove a version of Lemma 2.11 for terms.)

Proposition 2.13 (Unique Readability). *Every formula satisfies one, and only one of the following conditions.*

1. *φ is atomic.*
2. *φ is of the form $\neg\psi$.*
3. *φ is of the form $(\psi \land \chi)$.*
4. *φ is of the form $(\psi \lor \chi)$.*
5. *φ is of the form $(\psi \rightarrow \chi)$.*
6. *φ is of the form $(\psi \leftrightarrow \chi)$.*
7. *φ is of the form $\forall x\, \psi$.*
8. *φ is of the form $\exists x\, \psi$.*

Moreover, in each case ψ, or ψ and χ, are uniquely determined. This means that, e.g., there are no different pairs ψ, χ and ψ', χ' so that φ is both of the form $(\psi \rightarrow \chi)$ and $(\psi' \rightarrow \chi')$.

Proof. The formation rules require that if a formula is not atomic, it must start with an opening parenthesis (, $\neg$, or a quantifier. On the other hand, every formula that starts with one of the following symbols must be atomic: a predicate symbol, a function symbol, a constant symbol, $\bot$, $\top$.

So we really only have to show that if φ is of the form $(\psi * \chi)$ and also of the form $(\psi' *' \chi')$, then $\psi \equiv \psi'$, $\chi \equiv \chi'$, and $* = *'$.

So suppose both $\varphi \equiv (\psi * \chi)$ and $\varphi \equiv (\psi' *' \chi')$. Then either $\psi \equiv \psi'$ or not. If it is, clearly $* = *'$ and $\chi \equiv \chi'$, since they then are substrings of φ that begin in the same place and are of the same length. The other case is $\psi \not\equiv \psi'$. Since ψ and ψ' are both substrings of φ that begin at the same place, one must be a proper prefix of the other. But this is impossible by Lemma 2.11. □

2.5 Main operator of a Formula

fol:syn:mai:sec It is often useful to talk about the last operator used in constructing a formula φ. This operator is called the *main operator* of φ. Intuitively, it is the "outermost" operator of φ. For example, the main operator of $\neg\varphi$ is $\neg$, the main operator of $(\varphi \lor \psi)$ is $\lor$, etc. explanation

fol:syn:mai:def:main-op **Definition 2.14 (Main operator).** The *main operator* of a formula φ is defined as follows:

1. φ is atomic: φ has no main operator.
2. $\varphi \equiv \neg\psi$: the main operator of φ is $\neg$.
3. $\varphi \equiv (\psi \land \chi)$: the main operator of φ is $\land$.
4. $\varphi \equiv (\psi \lor \chi)$: the main operator of φ is $\lor$.
5. $\varphi \equiv (\psi \rightarrow \chi)$: the main operator of φ is $\rightarrow$.

6. $\varphi \equiv (\psi \leftrightarrow \chi)$: the main operator of φ is $\leftrightarrow$.

7. $\varphi \equiv \forall x\, \psi$: the main operator of φ is $\forall$.

8. $\varphi \equiv \exists x\, \psi$: the main operator of φ is $\exists$.

In each case, we intend the specific indicated *occurrence* of the main operator in the formula. For instance, since the formula $((\theta \to \alpha) \to (\alpha \to \theta))$ is of the form $(\psi \to \chi)$ where ψ is $(\theta \to \alpha)$ and χ is $(\alpha \to \theta)$, the second occurrence of $\to$ is the main operator.

explanation This is a *recursive* definition of a function which maps all non-atomic formulas to their main operator occurrence. Because of the way formulas are defined inductively, every formula φ satisfies one of the cases in Definition 2.14. This guarantees that for each non-atomic formula φ a main operator exists. Because each formula satisfies only one of these conditions, and because the smaller formulas from which φ is constructed are uniquely determined in each case, the main operator occurrence of φ is unique, and so we have defined a function.

We call formulas by the names in Table 2.1 depending on which symbol their main operator is.

Main operator	Type of formula	Example
none	atomic (formula)	$\bot$, $\top$, $R(t_1, \dots, t_n)$, $t_1 = t_2$
$\neg$	negation	$\neg\varphi$
$\land$	conjunction	$(\varphi \land \psi)$
$\lor$	disjunction	$(\varphi \lor \psi)$
$\to$	conditional	$(\varphi \to \psi)$
$\leftrightarrow$	biconditional	$(\varphi \leftrightarrow \psi)$
$\forall$	universal (formula)	$\forall x\, \varphi$
$\exists$	existential (formula)	$\exists x\, \varphi$

Table 2.1: Main operator and names of formulas

fol:syn:mai:tab:main-op

2.6 Subformulas

explanation It is often useful to talk about the formulas that “make up” a given formula. We call these its *subformulas*. Any formula counts as a subformula of itself; a subformula of φ other than φ itself is a *proper subformula*. fol:syn:sbf:sec

Definition 2.15 (Immediate Subformula). If φ is a formula, the *immediate subformulas* of φ are defined inductively as follows:

1. Atomic formulas have no immediate subformulas.

2. $\varphi \equiv \neg\psi$: The only immediate subformula of φ is ψ.

3. $\varphi \equiv (\psi * \chi)$: The immediate subformulas of φ are ψ and χ ($*$ is any one of the two-place connectives).

4. $\varphi \equiv \forall x\, \psi$: The only immediate subformula of φ is ψ.

5. $\varphi \equiv \exists x\, \psi$: The only immediate subformula of φ is ψ.

Definition 2.16 (Proper Subformula). If φ is a formula, the *proper subformulas* of φ are defined recursively as follows:

1. Atomic formulas have no proper subformulas.
2. $\varphi \equiv \neg\psi$: The proper subformulas of φ are ψ together with all proper subformulas of ψ.
3. $\varphi \equiv (\psi * \chi)$: The proper subformulas of φ are ψ, χ, together with all proper subformulas of ψ and those of χ.
4. $\varphi \equiv \forall x\, \psi$: The proper subformulas of φ are ψ together with all proper subformulas of ψ.
5. $\varphi \equiv \exists x\, \psi$: The proper subformulas of φ are ψ together with all proper subformulas of ψ.

Definition 2.17 (Subformula). The subformulas of φ are φ itself together with all its proper subformulas.

Note the subtle difference in how we have defined immediate subformulas and proper subformulas. In the first case, we have directly defined the immediate subformulas of a formula φ for each possible form of φ. It is an explicit definition by cases, and the cases mirror the inductive definition of the set of formulas. In the second case, we have also mirrored the way the set of all formulas is defined, but in each case we have also included the proper subformulas of the smaller formulas ψ, χ in addition to these formulas themselves. This makes the definition *recursive*. In general, a definition of a function on an inductively defined set (in our case, formulas) is recursive if the cases in the definition of the function make use of the function itself. To be well defined, we must make sure, however, that we only ever use the values of the function for arguments that come "before" the one we are defining—in our case, when defining "proper subformula" for $(\psi * \chi)$ we only use the proper subformulas of the "earlier" formulas ψ and χ. explanation

fol:syn:sbf: prop:subfrm-trans **Proposition 2.18.** *Suppose ψ is a subformula of φ and χ is a subformula of ψ. Then χ is a subformula of φ. In other words, the subformula relation is transitive.*

Problem 2.5. Prove Proposition 2.18.

fol:syn:sbf: prop:count-subfrms **Proposition 2.19.** *Suppose φ is a formula with n connectives and quantifiers. Then φ has at most $2n + 1$ subformulas.*

Problem 2.6. Prove Proposition 2.19.

2.7 Formation Sequences

Defining formulas via an inductive definition, and the complementary technique of proving properties of formulas via induction, is an elegant and efficient approach. However, it can also be useful to consider a more bottom-up, step-by-step approach to the construction of formulas, which we do here using the notion of a *formation sequence*. To show how terms and formulas can be introduced in this way without needing to refer to their inductive definitions, we first introduce the notion of an arbitrary string of symbols drawn from some language $\mathcal{L}$. fol:syn:fseq:sec

Definition 2.20 (Strings). Suppose $\mathcal{L}$ is a first-order language. An $\mathcal{L}$-*string* is a finite sequence of symbols of $\mathcal{L}$. Where the language $\mathcal{L}$ is clearly fixed by the context, we will often refer to a $\mathcal{L}$-string as a *string* simpliciter. fol:syn:fseq:defn:string

Example 2.21. For any first-order language $\mathcal{L}$, all $\mathcal{L}$-formulas are $\mathcal{L}$-strings, but not conversely. For example,

$$)(\mathsf{v}_0 \to \exists$$

is an $\mathcal{L}$-string but not an $\mathcal{L}$-formula.

Definition 2.22 (Formation sequences for terms). A finite sequence of $\mathcal{L}$-strings $\langle t_0, \dots, t_n \rangle$ is a *formation sequence* for a term t if $t \equiv t_n$ and for all $i \leq n$, either t_i is a variable or a constant symbol, or $\mathcal{L}$ contains a k-ary function symbol f and there exist $m_0, \dots, m_k < i$ such that $t_i \equiv f(t_{m_0}, \dots, t_{m_k})$. fol:syn:fseq:defn:fseq-trm

Example 2.23. The sequence

$$\langle \mathsf{c}_0, \mathsf{v}_0, \mathsf{f}_0^2(\mathsf{c}_0, \mathsf{v}_0), \mathsf{f}_0^1(\mathsf{f}_0^2(\mathsf{c}_0, \mathsf{v}_0)) \rangle$$

is a formation sequence for the term $\mathsf{f}_0^1(\mathsf{f}_0^2(\mathsf{c}_0, \mathsf{v}_0))$, as is

$$\langle \mathsf{v}_0, \mathsf{c}_0, \mathsf{f}_0^2(\mathsf{c}_0, \mathsf{v}_0), \mathsf{f}_0^1(\mathsf{f}_0^2(\mathsf{c}_0, \mathsf{v}_0)) \rangle.$$

Definition 2.24 (Formation sequences for formulas). A finite sequence of $\mathcal{L}$-strings $\langle \varphi_0, \dots, \varphi_n \rangle$ is a *formation sequence* for φ if $\varphi \equiv \varphi_n$ and for all $i \leq n$, either φ_i is an atomic formula or there exist $j, k < i$ and a variable x such that one of the following holds: fol:syn:fseq:defn:fseq-frm

1. $\varphi_i \equiv \neg\varphi_j$.
2. $\varphi_i \equiv (\varphi_j \land \varphi_k)$.
3. $\varphi_i \equiv (\varphi_j \lor \varphi_k)$.
4. $\varphi_i \equiv (\varphi_j \to \varphi_k)$.
5. $\varphi_i \equiv (\varphi_j \leftrightarrow \varphi_k)$.

6. $\varphi_i \equiv \forall x\, \varphi_j$.
7. $\varphi_i \equiv \exists x\, \varphi_j$.

Example 2.25.

$$\langle A^1_0(v_0), A^1_1(c_1), (A^1_1(c_1) \land A^1_0(v_0)), \exists v_0\,(A^1_1(c_1) \land A^1_0(v_0))\rangle$$

is a formation sequence of $\exists v_0\,(A^1_1(c_1) \land A^1_0(v_0))$, as is

$$\langle A^1_0(v_0), A^1_1(c_1), (A^1_1(c_1) \land A^1_0(v_0)), A^1_1(c_1), \\ \forall v_1\, A^1_0(v_0), \exists v_0\,(A^1_1(c_1) \land A^1_0(v_0))\rangle.$$

As can be seen from the second example, formation sequences may contain "junk": formulas which are redundant or do not contribute to the construction.

fol:syn:fseq: prop:formed

Proposition 2.26. *Every formula φ in $\mathrm{Frm}(\mathcal{L})$ has a formation sequence.*

Proof. Suppose φ is atomic. Then the sequence $\langle\varphi\rangle$ is a formation sequence for φ. Now suppose that ψ and χ have formation sequences $\langle\psi_0, \dots, \psi_n\rangle$ and $\langle\chi_0, \dots, \chi_m\rangle$ respectively.

1. If $\varphi \equiv \lnot\psi$, then $\langle\psi_0, \dots, \psi_n, \lnot\psi_n\rangle$ is a formation sequence for φ.
2. If $\varphi \equiv (\psi \land \chi)$, then $\langle\psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \land \chi_m)\rangle$ is a formation sequence for φ.
3. If $\varphi \equiv (\psi \lor \chi)$, then $\langle\psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \lor \chi_m)\rangle$ is a formation sequence for φ.
4. If $\varphi \equiv (\psi \rightarrow \chi)$, then $\langle\psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \rightarrow \chi_m)\rangle$ is a formation sequence for φ.
5. If $\varphi \equiv (\psi \leftrightarrow \chi)$, then $\langle\psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \leftrightarrow \chi_m)\rangle$ is a formation sequence for φ.
6. If $\varphi \equiv \forall x\, \psi$, then $\langle\psi_0, \dots, \psi_n, \forall x\, \psi_n\rangle$ is a formation sequence for φ.
7. If $\varphi \equiv \exists x\, \psi$, then $\langle\psi_0, \dots, \psi_n, \exists x\, \psi_n\rangle$ is a formation sequence for φ.

By the principle of induction on formulas, every formula has a formation sequence. □

We can also prove the converse. This is important because it shows that our two ways of defining formulas are equivalent: they give the same results. It also means that we can prove theorems about formulas by using ordinary induction on the length of formation sequences.

fol:syn:fseq: lem:fseq-init

Lemma 2.27. *Suppose that $\langle\varphi_0, \dots, \varphi_n\rangle$ is a formation sequence for φ_n, and that $k \le n$. Then $\langle\varphi_0, \dots, \varphi_k\rangle$ is a formation sequence for φ_k.*

Proof. Exercise. □

Problem 2.7. Prove Lemma 2.27.

Theorem 2.28. $\mathrm{Frm}(\mathcal{L})$ *is the set of all expressions (strings of symbols) in the language* $\mathcal{L}$ *with a formation sequence.* *fol:syn:fseq: thm:fseq-frm-equiv*

Proof. Let F be the set of all strings of symbols in the language $\mathcal{L}$ that have a formation sequence. We have seen in Proposition 2.26 that $\mathrm{Frm}(\mathcal{L}) \subseteq F$, so now we prove the converse.

Suppose φ has a formation sequence $\langle \varphi_0, \dots, \varphi_n \rangle$. We prove that $\varphi \in \mathrm{Frm}(\mathcal{L})$ by strong induction on n. Our induction hypothesis is that every string of symbols with a formation sequence of length $m < n$ is in $\mathrm{Frm}(\mathcal{L})$. By the definition of a formation sequence, either φ_n is atomic or there must exist $j, k < n$ such that one of the following is the case:

1. $\varphi_i \equiv \neg\varphi_j$.
2. $\varphi_i \equiv (\varphi_j \land \varphi_k)$.
3. $\varphi_i \equiv (\varphi_j \lor \varphi_k)$.
4. $\varphi_i \equiv (\varphi_j \rightarrow \varphi_k)$.
5. $\varphi_i \equiv (\varphi_j \leftrightarrow \varphi_k)$.
6. $\varphi_i \equiv \forall x\, \varphi_j$.
7. $\varphi_i \equiv \exists x\, \varphi_j$.

Now we reason by cases. If φ_n is atomic then $\varphi_n \in \mathrm{Frm}(\mathcal{L}_0)$. Suppose instead that $\varphi \equiv (\varphi_j \land \varphi_k)$. By Lemma 2.27, $\langle \varphi_0, \dots, \varphi_j \rangle$ and $\langle \varphi_0, \dots, \varphi_k \rangle$ are formation sequences for φ_j and φ_k, respectively. Since these are proper initial subsequences of the formation sequence for φ, they both have length less than n. Therefore by the induction hypothesis, φ_j and φ_k are in $\mathrm{Frm}(\mathcal{L}_0)$, and by the definition of a formula, so is $(\varphi_j \land \varphi_k)$. The other cases follow by parallel reasoning. □

Formation sequences for terms have similar properties to those for formulas.

Proposition 2.29. $\mathrm{Trm}(\mathcal{L})$ *is the set of all expressions* t *in the language* $\mathcal{L}$ *such that there exists a (term) formation sequence fo* t. *fol:syn:fseq: prop:fseq-trm-equiv*

Proof. Exercise. □

Problem 2.8. Prove Proposition 2.29. Hint: use a similar strategy to that used in the proof of Theorem 2.28.

There are two types of "junk" that can appear in formation sequences: repeated elements, and elements that are irrelevant to the construction of the formation or term. We can eliminate both by looking at minimal formation sequences.

fol:syn:fseq: defn:minimal-fseq **Definition 2.30 (Minimal formation sequences).** A formation sequence $\langle \varphi_0, \dots, \varphi_n \rangle$ for φ is a *minimal formation sequence* for φ if for every other formation sequence s for φ, the length of s is greater than or equal to $n + 1$.

fol:syn:fseq: prop:subformula-equivs **Proposition 2.31.** *The following are equivalent:*

1. *ψ is a sub-formula of φ.*
2. *ψ occurs in every formation sequence of φ.*
3. *ψ occurs in a minimal formation sequence of φ.*

Proof. Exercise. □

Problem 2.9. Prove Proposition 2.31.

Historical Remarks Formation sequences were introduced by Raymond Smullyan in his textbook *First-Order Logic* (Smullyan, 1968). Additional properties of formation sequences were established by Zuckerman (1973).

2.8 Free Variables and Sentences

fol:syn:fvs: sec

fol:syn:fvs: defn:free-occ **Definition 2.32 (Free occurrences of a variable).** The *free* occurrences of a variable in a formula are defined inductively as follows:

1. φ is atomic: all variable occurrences in φ are free.
2. $\varphi \equiv \neg\psi$: the free variable occurrences of φ are exactly those of ψ.
3. $\varphi \equiv (\psi * \chi)$: the free variable occurrences of φ are those in ψ together with those in χ.
4. $\varphi \equiv \forall x\, \psi$: the free variable occurrences in φ are all of those in ψ except for occurrences of x.
5. $\varphi \equiv \exists x\, \psi$: the free variable occurrences in φ are all of those in ψ except for occurrences of x.

Definition 2.33 (Bound Variables). An occurrence of a variable in a formula φ is *bound* if it is not free.

Problem 2.10. Give an inductive definition of the bound variable occurrences along the lines of Definition 2.32.

Definition 2.34 (Scope). If $\forall x\,\psi$ is an occurrence of a subformula in a formula φ, then the corresponding occurrence of ψ in φ is called the *scope* of the corresponding occurrence of $\forall x$. Similarly for $\exists x$.

If ψ is the scope of a quantifier occurrence $\forall x$ or $\exists x$ in φ, then the free occurrences of x in ψ are bound in $\forall x\,\psi$ and $\exists x\,\psi$. We say that these occurrences are *bound by* the mentioned quantifier occurrence.

Example 2.35. Consider the following formula:

$$\exists v_0\ \underbrace{A^2_0(v_0, v_1)}_{\psi}$$

ψ represents the scope of $\exists v_0$. The quantifier binds the occurrence of v_0 in ψ, but does not bind the occurrence of v_1. So v_1 is a free variable in this case.

We can now see how this might work in a more complicated formula φ:

$$\forall v_0\ \underbrace{(A^1_0(v_0) \to A^2_0(v_0, v_1))}_{\psi} \to \exists v_1\ \underbrace{(A^2_1(v_0, v_1) \lor \forall v_0\ \overbrace{\neg A^1_1(v_0)}^{\theta})}_{\chi}$$

ψ is the scope of the first $\forall v_0$, χ is the scope of $\exists v_1$, and θ is the scope of the second $\forall v_0$. The first $\forall v_0$ binds the occurrences of v_0 in ψ, $\exists v_1$ binds the occurrence of v_1 in χ, and the second $\forall v_0$ binds the occurrence of v_0 in θ. The first occurrence of v_1 and the fourth occurrence of v_0 are free in φ. The last occurrence of v_0 is free in θ, but bound in χ and φ.

Definition 2.36 (Sentence). A formula φ is a *sentence* iff it contains no free occurrences of variables.

2.9 Substitution

fol:syn:sub:sec

Definition 2.37 (Substitution in a term). We define $s[t/x]$, the result of *substituting* t for every occurrence of x in s, recursively:

1. $s \equiv c$: $s[t/x]$ is just s.
2. $s \equiv y$: $s[t/x]$ is also just s, provided y is a variable and $y \not\equiv x$.
3. $s \equiv x$: $s[t/x]$ is t.
4. $s \equiv f(t_1, \dots, t_n)$: $s[t/x]$ is $f(t_1[t/x], \dots, t_n[t/x])$.

Definition 2.38. A term t is *free for* x in φ if none of the free occurrences of x in φ occur in the scope of a quantifier that binds a variable in t.

Example 2.39.

1. v_8 is free for v_1 in $\exists v_3 A^2_4(v_3, v_1)$
2. $f^2_1(v_1, v_2)$ is *not* free for v_0 in $\forall v_2 A^2_4(v_0, v_2)$

Definition 2.40 (Substitution in a formula). If φ is a formula, x is a variable, and t is a term free for x in φ, then $\varphi[t/x]$ is the result of substituting t for all free occurrences of x in φ.

1. $\varphi \equiv \bot$: $\varphi[t/x]$ is $\bot$.
2. $\varphi \equiv \top$: $\varphi[t/x]$ is $\top$.
3. $\varphi \equiv P(t_1, \dots, t_n)$: $\varphi[t/x]$ is $P(t_1[t/x], \dots, t_n[t/x])$.
4. $\varphi \equiv t_1 = t_2$: $\varphi[t/x]$ is $t_1[t/x] = t_2[t/x]$.
5. $\varphi \equiv \lnot\psi$: $\varphi[t/x]$ is $\lnot\psi[t/x]$.
6. $\varphi \equiv (\psi \land \chi)$: $\varphi[t/x]$ is $(\psi[t/x] \land \chi[t/x])$.
7. $\varphi \equiv (\psi \lor \chi)$: $\varphi[t/x]$ is $(\psi[t/x] \lor \chi[t/x])$.
8. $\varphi \equiv (\psi \to \chi)$: $\varphi[t/x]$ is $(\psi[t/x] \to \chi[t/x])$.
9. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\varphi[t/x]$ is $(\psi[t/x] \leftrightarrow \chi[t/x])$.
10. $\varphi \equiv \forall y\, \psi$: $\varphi[t/x]$ is $\forall y\, \psi[t/x]$, provided y is a variable other than x; otherwise $\varphi[t/x]$ is just φ.
11. $\varphi \equiv \exists y\, \psi$: $\varphi[t/x]$ is $\exists y\, \psi[t/x]$, provided y is a variable other than x; otherwise $\varphi[t/x]$ is just φ.

Note that substitution may be vacuous: If x does not occur in φ at all, then $\varphi[t/x]$ is just φ. explanation

The restriction that t must be free for x in φ is necessary to exclude cases like the following. If $\varphi \equiv \exists y\, x < y$ and $t \equiv y$, then $\varphi[t/x]$ would be $\exists y\, y < y$. In this case the free variable y is "captured" by the quantifier $\exists y$ upon substitution, and that is undesirable. For instance, we would like it to be the case that whenever $\forall x\, \psi$ holds, so does $\psi[t/x]$. But consider $\forall x\, \exists y\, x < y$ (here ψ is $\exists y\, x < y$). It is a sentence that is true about, e.g., the natural numbers: for every number x there is a number y greater than it. If we allowed y as a possible substitution for x, we would end up with $\psi[y/x] \equiv \exists y\, y < y$, which is false. We prevent this by requiring that none of the free variables in t would end up being bound by a quantifier in φ.

We often use the following convention to avoid cumbersome notation: If φ is a formula which may contain the variable x free, we also write $\varphi(x)$ to indicate this. When it is clear which φ and x we have in mind, and t is a term (assumed to be free for x in $\varphi(x)$), then we write $\varphi(t)$ as short for $\varphi[t/x]$. So for instance, we might say, "we call $\varphi(t)$ an instance of $\forall x\, \varphi(x)$." By this we mean that if φ is any formula, x a variable, and t a term that's free for x in φ, then $\varphi[t/x]$ is an instance of $\forall x\, \varphi$.

Chapter 3

Semantics of First-Order Logic

3.1 Introduction

fol:syn:its:sec

Giving the meaning of expressions is the domain of semantics. The central concept in semantics is that of satisfaction in a structure. A structure gives meaning to the building blocks of the language: a domain is a non-empty set of objects. The quantifiers are interpreted as ranging over this domain, constant symbols are assigned elements in the domain, function symbols are assigned functions from the domain to itself, and predicate symbols are assigned relations on the domain. The domain together with assignments to the basic vocabulary constitutes a structure. Variables may appear in formulas, and in order to give a semantics, we also have to assign elements of the domain to them—this is a variable assignment. The satisfaction relation, finally, brings these together. A formula may be satisfied in a structure $\mathfrak{M}$ relative to a variable assignment s, written as $\mathfrak{M}, s \vDash \varphi$. This relation is also defined by induction on the structure of φ, using the truth tables for the logical connectives to define, say, satisfaction of $(\varphi \land \psi)$ in terms of satisfaction (or not) of φ and ψ. It then turns out that the variable assignment is irrelevant if the formula φ is a sentence, i.e., has no free variables, and so we can talk of sentences being simply satisfied (or not) in structures.

On the basis of the satisfaction relation $\mathfrak{M} \vDash \varphi$ for sentences we can then define the basic semantic notions of validity, entailment, and satisfiability. A sentence is valid, $\vDash \varphi$, if every structure satisfies it. It is entailed by a set of sentences, $\Gamma \vDash \varphi$, if every structure that satisfies all the sentences in Γ also satisfies φ. And a set of sentences is satisfiable if some structure satisfies all sentences in it at the same time. Because formulas are inductively defined, and satisfaction is in turn defined by induction on the structure of formulas, we can use induction to prove properties of our semantics and to relate the semantic notions defined.

3.2 Structures for First-order Languages

fol:syn:str:sec First-order languages are, by themselves, *uninterpreted:* the constant symbols, function symbols, and predicate symbols have no specific meaning attached to them. Meanings are given by specifying a *structure*. It specifies the *domain*, i.e., the objects which the constant symbols pick out, the function symbols operate on, and the quantifiers range over. In addition, it specifies which constant symbols pick out which objects, how a function symbol maps objects to objects, and which objects the predicate symbols apply to. Structures are the basis for *semantic* notions in logic, e.g., the notion of consequence, validity, satisfiability. They are variously called "structures," "interpretations," or "models" in the literature. explanation

Definition 3.1 (Structures). A *structure* $\mathfrak{M}$, for a language $\mathcal{L}$ of first-order logic consists of the following elements:

1. *Domain:* a non-empty set, $|\mathfrak{M}|$
2. *Interpretation of constant symbols:* for each constant symbol c of $\mathcal{L}$, an element $c^{\mathfrak{M}} \in |\mathfrak{M}|$
3. *Interpretation of predicate symbols:* for each n-place predicate symbol R of $\mathcal{L}$ (other than $=$), an n-place relation $R^{\mathfrak{M}} \subseteq |\mathfrak{M}|^n$
4. *Interpretation of function symbols:* for each n-place function symbol f of $\mathcal{L}$, an n-place function $f^{\mathfrak{M}} \colon |\mathfrak{M}|^n \to |\mathfrak{M}|$

Example 3.2. A structure $\mathfrak{M}$ for the language of arithmetic consists of a set, an element of $|\mathfrak{M}|$, $\mathsf{o}^{\mathfrak{M}}$, as interpretation of the constant symbol o, a one-place function $\prime^{\mathfrak{M}} \colon |\mathfrak{M}| \to |\mathfrak{M}|$, two two-place functions $+^{\mathfrak{M}}$ and $\times^{\mathfrak{M}}$, both $|\mathfrak{M}|^2 \to |\mathfrak{M}|$, and a two-place relation $<^{\mathfrak{M}} \subseteq |\mathfrak{M}|^2$.

An obvious example of such a structure is the following:

1. $|\mathfrak{N}| = \mathbb{N}$
2. $\mathsf{o}^{\mathfrak{N}} = 0$
3. $\prime^{\mathfrak{N}}(n) = n + 1$ for all $n \in \mathbb{N}$
4. $+^{\mathfrak{N}}(n, m) = n + m$ for all $n, m \in \mathbb{N}$
5. $\times^{\mathfrak{N}}(n, m) = n \cdot m$ for all $n, m \in \mathbb{N}$
6. $<^{\mathfrak{N}} = \{\langle n, m \rangle : n \in \mathbb{N}, m \in \mathbb{N}, n < m\}$

The structure $\mathfrak{N}$ for $\mathcal{L}_A$ so defined is called the *standard model of arithmetic*, because it interprets the non-logical constants of $\mathcal{L}_A$ exactly how you would expect.

However, there are many other possible structures for $\mathcal{L}_A$. For instance, we might take as the domain the set $\mathbb{Z}$ of integers instead of $\mathbb{N}$, and define the interpretations of o, $\prime$, $+$, $\times$, $<$ accordingly. But we can also define structures for $\mathcal{L}_A$ which have nothing even remotely to do with numbers.

Example 3.3. A structure $\mathfrak{M}$ for the language $\mathcal{L}_Z$ of set theory requires just a set and a single-two place relation. So technically, e.g., the set of people plus the relation "x is older than y" could be used as a structure for $\mathcal{L}_Z$, as well as $\mathbb{N}$ together with $n \geq m$ for $n, m \in \mathbb{N}$.

A particularly interesting structure for $\mathcal{L}_Z$ in which the elements of the domain are actually sets, and the interpretation of $\in$ actually is the relation "x is an element of y" is the structure $\mathfrak{HF}$ of *hereditarily finite sets*:

1. $|\mathfrak{HF}| = \emptyset \cup \wp(\emptyset) \cup \wp(\wp(\emptyset)) \cup \wp(\wp(\wp(\emptyset))) \cup \dots$;
2. $\in^{\mathfrak{HF}} = \{\langle x, y\rangle : x, y \in |\mathfrak{HF}|, x \in y\}$.

digression The stipulations we make as to what counts as a structure impact our logic. For example, the choice to prevent empty domains ensures, given the usual account of satisfaction (or truth) for quantified sentences, that $\exists x\, (\varphi(x) \lor \lnot\varphi(x))$ is valid—that is, a logical truth. And the stipulation that all constant symbols must refer to an object in the domain ensures that the existential generalization is a sound pattern of inference: $\varphi(a)$, therefore $\exists x\, \varphi(x)$. If we allowed names to refer outside the domain, or to not refer, then we would be on our way to a *free logic*, in which existential generalization requires an additional premise: $\varphi(a)$ and $\exists x\, x = a$, therefore $\exists x\, \varphi(x)$.

3.3 Covered Structures for First-order Languages

explanation Recall that a term is *closed* if it contains no variables. fol:syn:cov:sec

Definition 3.4 (Value of closed terms). If t is a closed term of the language $\mathcal{L}$ and $\mathfrak{M}$ is a structure for $\mathcal{L}$, the *value* $\mathrm{Val}^{\mathfrak{M}}(t)$ is defined as follows:

1. If t is just the constant symbol c, then $\mathrm{Val}^{\mathfrak{M}}(c) = c^{\mathfrak{M}}$.
2. If t is of the form $f(t_1, \dots, t_n)$, then
$$\mathrm{Val}^{\mathfrak{M}}(t) = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}(t_1), \dots, \mathrm{Val}^{\mathfrak{M}}(t_n)).$$

Definition 3.5 (Covered structure). A structure is *covered* if every element of the domain is the value of some closed term.

Example 3.6. Let $\mathcal{L}$ be the language with constant symbols zero, one, two, …, the binary predicate symbol $<$, and the binary function symbols $+$ and $\times$. Then a structure $\mathfrak{M}$ for $\mathcal{L}$ is the one with domain $|\mathfrak{M}| = \{0, 1, 2, \dots\}$ and assignments $\mathit{zero}^{\mathfrak{M}} = 0$, $\mathit{one}^{\mathfrak{M}} = 1$, $\mathit{two}^{\mathfrak{M}} = 2$, and so forth. For the binary relation symbol $<$, the set $<^{\mathfrak{M}}$ is the set of all pairs $\langle c_1, c_2\rangle \in |\mathfrak{M}|^2$ such that c_1 is less than c_2: for example, $\langle 1, 3\rangle \in <^{\mathfrak{M}}$ but $\langle 2, 2\rangle \notin <^{\mathfrak{M}}$. For the binary function symbol $+$, define $+^{\mathfrak{M}}$ in the usual way—for example, $+^{\mathfrak{M}}(2, 3)$ maps to 5, and similarly for the binary function symbol $\times$. Hence, the value of

four is just 4, and the value of $\times(\mathit{two}, +(\mathit{three}, \mathit{zero}))$ (or in infix notation, $\mathit{two} \times (\mathit{three} + \mathit{zero})$) is

$$\begin{aligned}
\mathrm{Val}^{\mathfrak{M}}(\times(\mathit{two}, +(\mathit{three}, \mathit{zero})) = \\
&= \times^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}(\mathit{two}), \mathrm{Val}^{\mathfrak{M}}(+(\mathit{three}, \mathit{zero}))) \\
&= \times^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}(\mathit{two}), +^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}(\mathit{three}), \mathrm{Val}^{\mathfrak{M}}(\mathit{zero}))) \\
&= \times^{\mathfrak{M}}(\mathit{two}^{\mathfrak{M}}, +^{\mathfrak{M}}(\mathit{three}^{\mathfrak{M}}, \mathit{zero}^{\mathfrak{M}})) \\
&= \times^{\mathfrak{M}}(2, +^{\mathfrak{M}}(3, 0)) \\
&= \times^{\mathfrak{M}}(2, 3) \\
&= 6
\end{aligned}$$

Problem 3.1. Is $\mathfrak{N}$, the standard model of arithmetic, covered? Explain.

3.4 Satisfaction of a Formula in a Structure

fol:syn:sat:sec

The basic notion that relates expressions such as terms and formulas, on the one hand, and structures on the other, are those of *value* of a term and *satisfaction* of a formula. Informally, the value of a term is an element of a structure—if the term is just a constant, its value is the object assigned to the constant by the structure, and if it is built up using function symbols, the value is computed from the values of constants and the functions assigned to the functions in the term. A formula is *satisfied* in a structure if the interpretation given to the predicates makes the formula true in the domain of the structure. This notion of satisfaction is specified inductively: the specification of the structure directly states when atomic formulas are satisfied, and we define when a complex formula is satisfied depending on the main connective or quantifier and whether or not the immediate subformulas are satisfied. explanation

The case of the quantifiers here is a bit tricky, as the immediate subformula of a quantified formula has a free variable, and structures don't specify the values of variables. In order to deal with this difficulty, we also introduce *variable assignments* and define satisfaction not with respect to a structure alone, but with respect to a structure plus a variable assignment.

Definition 3.7 (Variable Assignment). A *variable assignment* s for a structure $\mathfrak{M}$ is a function which maps each variable to an element of $|\mathfrak{M}|$, i.e., $s\colon \mathrm{Var} \to |\mathfrak{M}|$.

A structure assigns a value to each constant symbol, and a variable assignment to each variable. But we want to use terms built up from them to also name elements of the domain. For this we define the value of terms inductively. For constant symbols and variables the value is just as the structure or the variable assignment specifies it; for more complex terms it is computed recursively using the functions the structure assigns to the function symbols. explanation

Definition 3.8 (Value of Terms). If t is a term of the language $\mathcal{L}$, $\mathfrak{M}$ is a structure for $\mathcal{L}$, and s is a variable assignment for $\mathfrak{M}$, the *value* $\mathrm{Val}_s^{\mathfrak{M}}(t)$ is defined as follows:

1. $t \equiv c$: $\mathrm{Val}_s^{\mathfrak{M}}(t) = c^{\mathfrak{M}}$.
2. $t \equiv x$: $\mathrm{Val}_s^{\mathfrak{M}}(t) = s(x)$.
3. $t \equiv f(t_1, \dots, t_n)$:
$$\mathrm{Val}_s^{\mathfrak{M}}(t) = f^{\mathfrak{M}}(\mathrm{Val}_s^{\mathfrak{M}}(t_1), \dots, \mathrm{Val}_s^{\mathfrak{M}}(t_n)).$$

Definition 3.9 (x-Variant). If s is a variable assignment for a structure $\mathfrak{M}$, then any variable assignment s' for $\mathfrak{M}$ which differs from s at most in what it assigns to x is called an *x-variant* of s. If s' is an x-variant of s we write $s' \sim_x s$.

explanation

Note that an x-variant of an assignment s does not *have* to assign something different to x. In fact, every assignment counts as an x-variant of itself.

Definition 3.10. If s is a variable assignment for a structure $\mathfrak{M}$ and $m \in |\mathfrak{M}|$, then the assignment $s[m/x]$ is the variable assignment defined by
$$s[m/x](y) = \begin{cases} m & \text{if } y \equiv x \\ s(y) & \text{otherwise.} \end{cases}$$

In other words, $s[m/x]$ is the particular x-variant of s which assigns the domain element m to x, and assigns the same things to variables other than x that s does.

Definition 3.11 (Satisfaction). Satisfaction of a formula φ in a structure $\mathfrak{M}$ relative to a variable assignment s, in symbols: $\mathfrak{M}, s \vDash \varphi$, is defined recursively as follows. (We write $\mathfrak{M}, s \nvDash \varphi$ to mean "not $\mathfrak{M}, s \vDash \varphi$.") fol:syn:sat: defn:satisfaction

1. $\varphi \equiv \bot$: $\mathfrak{M}, s \nvDash \varphi$.
2. $\varphi \equiv \top$: $\mathfrak{M}, s \vDash \varphi$.
3. $\varphi \equiv R(t_1, \dots, t_n)$: $\mathfrak{M}, s \vDash \varphi$ iff $\langle \mathrm{Val}_s^{\mathfrak{M}}(t_1), \dots, \mathrm{Val}_s^{\mathfrak{M}}(t_n)\rangle \in R^{\mathfrak{M}}$.
4. $\varphi \equiv t_1 = t_2$: $\mathfrak{M}, s \vDash \varphi$ iff $\mathrm{Val}_s^{\mathfrak{M}}(t_1) = \mathrm{Val}_s^{\mathfrak{M}}(t_2)$.
5. $\varphi \equiv \lnot\psi$: $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}, s \nvDash \psi$.
6. $\varphi \equiv (\psi \land \chi)$: $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}, s \vDash \psi$ and $\mathfrak{M}, s \vDash \chi$.
7. $\varphi \equiv (\psi \lor \chi)$: $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}, s \vDash \psi$ or $\mathfrak{M}, s \vDash \chi$ (or both).
8. $\varphi \equiv (\psi \to \chi)$: $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}, s \nvDash \psi$ or $\mathfrak{M}, s \vDash \chi$ (or both).

9. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, s \vDash \varphi$ iff either both $\mathfrak{M}, s \vDash \psi$ and $\mathfrak{M}, s \vDash \chi$, or neither $\mathfrak{M}, s \vDash \psi$ nor $\mathfrak{M}, s \vDash \chi$.

10. $\varphi \equiv \forall x\, \psi$: $\mathfrak{M}, s \vDash \varphi$ iff for every element $m \in |\mathfrak{M}|$, $\mathfrak{M}, s[m/x] \vDash \psi$.

11. $\varphi \equiv \exists x\, \psi$: $\mathfrak{M}, s \vDash \varphi$ iff for at least one element $m \in |\mathfrak{M}|$, $\mathfrak{M}, s[m/x] \vDash \psi$.

The variable assignments are important in the last two clauses. We cannot define satisfaction of $\forall x\, \psi(x)$ by "for all $m \in |\mathfrak{M}|$, $\mathfrak{M} \vDash \psi(m)$." We cannot define satisfaction of $\exists x\, \psi(x)$ by "for at least one $m \in |\mathfrak{M}|$, $\mathfrak{M} \vDash \psi(m)$." The reason is that if $m \in |\mathfrak{M}|$, it is not a symbol of the language, and so $\psi(m)$ is not a formula (that is, $\psi[m/x]$ is undefined). We also cannot assume that we have constant symbols or terms available that name every element of $\mathfrak{M}$, since there is nothing in the definition of structures that requires it. In the standard language, the set of constant symbols is denumerable, so if $|\mathfrak{M}|$ is not enumerable there aren't even enough constant symbols to name every object. explanation

We solve this problem by introducing variable assignments, which allow us to link variables directly with elements of the domain. Then instead of saying that, e.g., $\exists x\, \psi(x)$ is satisfied in $\mathfrak{M}$ iff for at least one $m \in |\mathfrak{M}|$, we say it is satisfied in $\mathfrak{M}$ *relative to* s iff $\psi(x)$ is satisfied relative to $s[m/x]$ for at least one $m \in |\mathfrak{M}|$.

Example 3.12. Let $\mathcal{L} = \{a, b, f, R\}$ where a and b are constant symbols, f is a two-place function symbol, and R is a two-place predicate symbol. Consider the structure $\mathfrak{M}$ defined by:

1. $|\mathfrak{M}| = \{1, 2, 3, 4\}$
2. $a^{\mathfrak{M}} = 1$
3. $b^{\mathfrak{M}} = 2$
4. $f^{\mathfrak{M}}(x, y) = x + y$ if $x + y \leq 3$ and $= 3$ otherwise.
5. $R^{\mathfrak{M}} = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle, \langle 2, 4\rangle\}$

The function $s(x) = 1$ that assigns $1 \in |\mathfrak{M}|$ to every variable is a variable assignment for $\mathfrak{M}$.

Then

$$\mathrm{Val}^{\mathfrak{M}}_s(f(a, b)) = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_s(a), \mathrm{Val}^{\mathfrak{M}}_s(b)).$$

Since a and b are constant symbols, $\mathrm{Val}^{\mathfrak{M}}_s(a) = a^{\mathfrak{M}} = 1$ and $\mathrm{Val}^{\mathfrak{M}}_s(b) = b^{\mathfrak{M}} = 2$. So

$$\mathrm{Val}^{\mathfrak{M}}_s(f(a, b)) = f^{\mathfrak{M}}(1, 2) = 1 + 2 = 3.$$

To compute the value of $f(f(a, b), a)$ we have to consider

$$\mathrm{Val}^{\mathfrak{M}}_s(f(f(a, b), a)) = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_s(f(a, b)), \mathrm{Val}^{\mathfrak{M}}_s(a)) = f^{\mathfrak{M}}(3, 1) = 3,$$

since $3 + 1 > 3$. Since $s(x) = 1$ and $\mathrm{Val}^{\mathfrak{M}}_s(x) = s(x)$, we also have

$$\mathrm{Val}^{\mathfrak{M}}_s(f(f(a,b),x)) = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_s(f(a,b)), \mathrm{Val}^{\mathfrak{M}}_s(x)) = f^{\mathfrak{M}}(3,1) = 3,$$

An atomic formula $R(t_1, t_2)$ is satisfied if the tuple of values of its arguments, i.e., $\langle \mathrm{Val}^{\mathfrak{M}}_s(t_1), \mathrm{Val}^{\mathfrak{M}}_s(t_2)\rangle$, is an element of $R^{\mathfrak{M}}$. So, e.g., we have $\mathfrak{M}, s \vDash R(b, f(a,b))$ since $\langle \mathrm{Val}^{\mathfrak{M}}(b), \mathrm{Val}^{\mathfrak{M}}(f(a,b))\rangle = \langle 2, 3\rangle \in R^{\mathfrak{M}}$, but $\mathfrak{M}, s \nvDash R(x, f(a,b))$ since $\langle 1, 3\rangle \notin R^{\mathfrak{M}}[s]$.

To determine if a non-atomic formula φ is satisfied, you apply the clauses in the inductive definition that applies to the main connective. For instance, the main connective in $R(a,a) \to (R(b,x) \lor R(x,b))$ is the $\to$, and

$$\begin{aligned}&\mathfrak{M}, s \vDash R(a,a) \to (R(b,x) \lor R(x,b)) \text{ iff}\\ &\qquad \mathfrak{M}, s \nvDash R(a,a) \text{ or } \mathfrak{M}, s \vDash R(b,x) \lor R(x,b)\end{aligned}$$

Since $\mathfrak{M}, s \vDash R(a,a)$ (because $\langle 1, 1\rangle \in R^{\mathfrak{M}}$) we can't yet determine the answer and must first figure out if $\mathfrak{M}, s \vDash R(b,x) \lor R(x,b)$:

$$\begin{aligned}&\mathfrak{M}, s \vDash R(b,x) \lor R(x,b) \text{ iff}\\ &\qquad \mathfrak{M}, s \vDash R(b,x) \text{ or } \mathfrak{M}, s \vDash R(x,b)\end{aligned}$$

And this is the case, since $\mathfrak{M}, s \vDash R(x,b)$ (because $\langle 1, 2\rangle \in R^{\mathfrak{M}}$).

Recall that an x-variant of s is a variable assignment that differs from s at most in what it assigns to x. For every element of $|\mathfrak{M}|$, there is an x-variant of s:

$$\begin{aligned} s_1 &= s[1/x], & s_2 &= s[2/x],\\ s_3 &= s[3/x], & s_4 &= s[4/x].\end{aligned}$$

So, e.g., $s_2(x) = 2$ and $s_2(y) = s(y) = 1$ for all variables y other than x. These are all the x-variants of s for the structure $\mathfrak{M}$, since $|\mathfrak{M}| = \{1, 2, 3, 4\}$. Note, in particular, that $s_1 = s$ (s is always an x-variant of itself).

To determine if an existentially quantified formula $\exists x\, \varphi(x)$ is satisfied, we have to determine if $\mathfrak{M}, s[m/x] \vDash \varphi(x)$ for at least one $m \in |\mathfrak{M}|$. So,

$$\mathfrak{M}, s \vDash \exists x\, (R(b,x) \lor R(x,b)),$$

since $\mathfrak{M}, s[1/x] \vDash R(b,x) \lor R(x,b)$ ($s[3/x]$ would also fit the bill). But,

$$\mathfrak{M}, s \nvDash \exists x\, (R(b,x) \land R(x,b))$$

since, whichever $m \in |\mathfrak{M}|$ we pick, $\mathfrak{M}, s[m/x] \nvDash R(b,x) \land R(x,b)$.

To determine if a universally quantified formula $\forall x\, \varphi(x)$ is satisfied, we have to determine if $\mathfrak{M}, s[m/x] \vDash \varphi(x)$ for all $m \in |\mathfrak{M}|$. So,

$$\mathfrak{M}, s \vDash \forall x\, (R(x,a) \to R(a,x)),$$

since $\mathfrak{M}, s[m/x] \vDash R(x, a) \to R(a, x)$ for all $m \in |\mathfrak{M}|$. For $m = 1$, we have $\mathfrak{M}, s[1/x] \vDash R(a, x)$ so the consequent is true; for $m = 2$, 3, and 4, we have $\mathfrak{M}, s[m/x] \nvDash R(x, a)$, so the antecedent is false. But,

$$\mathfrak{M}, s \nvDash \forall x\, (R(a, x) \to R(x, a))$$

since $\mathfrak{M}, s[2/x] \nvDash R(a, x) \to R(x, a)$ (because $\mathfrak{M}, s[2/x] \vDash R(a, x)$ and $\mathfrak{M}, s[2/x] \nvDash R(x, a)$).

For a more complicated case, consider

$$\forall x\, (R(a, x) \to \exists y\, R(x, y)).$$

Since $\mathfrak{M}, s[3/x] \nvDash R(a, x)$ and $\mathfrak{M}, s[4/x] \nvDash R(a, x)$, the interesting cases where we have to worry about the consequent of the conditional are only $m = 1$ and $= 2$. Does $\mathfrak{M}, s[1/x] \vDash \exists y\, R(x, y)$ hold? It does if there is at least one $n \in |\mathfrak{M}|$ so that $\mathfrak{M}, s[1/x][n/y] \vDash R(x, y)$. In fact, if we take $n = 1$, we have $s[1/x][n/y] = s[1/y] = s$. Since $s(x) = 1$, $s(y) = 1$, and $\langle 1, 1 \rangle \in R^{\mathfrak{M}}$, the answer is yes.

To determine if $\mathfrak{M}, s[2/x] \vDash \exists y\, R(x, y)$, we have to look at the variable assignments $s[2/x][n/y]$. Here, for $n = 1$, this assignment is $s_2 = s[2/x]$, which does not satisfy $R(x, y)$ ($s_2(x) = 2$, $s_2(y) = 1$, and $\langle 2, 1 \rangle \notin R^{\mathfrak{M}}$). However, consider $s[2/x][3/y] = s_2[3/y]$. $\mathfrak{M}, s_2[3/y] \vDash R(x, y)$ since $\langle 2, 3 \rangle \in R^{\mathfrak{M}}$, and so $\mathfrak{M}, s_2 \vDash \exists y\, R(x, y)$.

So, for all $n \in |\mathfrak{M}|$, either $\mathfrak{M}, s[m/x] \nvDash R(a, x)$ (if $m = 3$, 4) or $\mathfrak{M}, s[m/x] \vDash \exists y\, R(x, y)$ (if $m = 1$, 2), and so

$$\mathfrak{M}, s \vDash \forall x\, (R(a, x) \to \exists y\, R(x, y)).$$

On the other hand,

$$\mathfrak{M}, s \nvDash \exists x\, (R(a, x) \land \forall y\, R(x, y)).$$

We have $\mathfrak{M}, s[m/x] \vDash R(a, x)$ only for $m = 1$ and $m = 2$. But for both of these values of m, there is in turn an $n \in |\mathfrak{M}|$, namely $n = 4$, so that $\mathfrak{M}, s[m/x][n/y] \nvDash R(x, y)$ and so $\mathfrak{M}, s[m/x] \nvDash \forall y\, R(x, y)$ for $m = 1$ and $m = 2$. In sum, there is no $m \in |\mathfrak{M}|$ such that $\mathfrak{M}, s[m/x] \vDash R(a, x) \land \forall y\, R(x, y)$.

Problem 3.2. Let $\mathcal{L} = \{c, f, A\}$ with one constant symbol, one one-place function symbol and one two-place predicate symbol, and let the structure $\mathfrak{M}$ be given by

1. $|\mathfrak{M}| = \{1, 2, 3\}$
2. $c^{\mathfrak{M}} = 3$
3. $f^{\mathfrak{M}}(1) = 2, f^{\mathfrak{M}}(2) = 3, f^{\mathfrak{M}}(3) = 2$
4. $A^{\mathfrak{M}} = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 3 \rangle\}$

(a) Let $s(v) = 1$ for all variables v. Find out whether

$$\mathfrak{M}, s \vDash \exists x\, (A(f(z), c) \rightarrow \forall y\, (A(y, x) \lor A(f(y), x)))$$

Explain why or why not.

(b) Give a different structure and variable assignment in which the formula is not satisfied.

3.5 Variable Assignments

explanation A variable assignment s provides a value for *every* variable—and there are infinitely many of them. This is of course not necessary. We require variable assignments to assign values to all variables simply because it makes things a lot easier. The value of a term t, and whether or not a formula φ is satisfied in a structure with respect to s, only depend on the assignments s makes to the variables in t and the free variables of φ. This is the content of the next two propositions. To make the idea of "depends on" precise, we show that any two variable assignments that agree on all the variables in t give the same value, and that φ is satisfied relative to one iff it is satisfied relative to the other if two variable assignments agree on all free variables of φ. fol:syn:ass:sec

Proposition 3.13. *If the variables in a term t are among $x_1, \ldots, x_n$, and $s_1(x_i) = s_2(x_i)$ for $i = 1, \ldots, n$, then $\mathrm{Val}^{\mathfrak{M}}_{s_1}(t) = \mathrm{Val}^{\mathfrak{M}}_{s_2}(t)$.* fol:syn:ass:prop:valindep

Proof. By induction on the complexity of t. For the base case, t can be a constant symbol or one of the variables $x_1, \ldots, x_n$. If $t = c$, then $\mathrm{Val}^{\mathfrak{M}}_{s_1}(t) = c^{\mathfrak{M}} = \mathrm{Val}^{\mathfrak{M}}_{s_2}(t)$. If $t = x_i$, $s_1(x_i) = s_2(x_i)$ by the hypothesis of the proposition, and so $\mathrm{Val}^{\mathfrak{M}}_{s_1}(t) = s_1(x_i) = s_2(x_i) = \mathrm{Val}^{\mathfrak{M}}_{s_2}(t)$.

For the inductive step, assume that $t = f(t_1, \ldots, t_k)$ and that the claim holds for $t_1, \ldots, t_k$. Then

$$\begin{aligned} \mathrm{Val}^{\mathfrak{M}}_{s_1}(t) &= \mathrm{Val}^{\mathfrak{M}}_{s_1}(f(t_1, \ldots, t_k)) = \\ &= f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_{s_1}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_k)) \end{aligned}$$

For $j = 1, \ldots, k$, the variables of t_j are among $x_1, \ldots, x_n$. By induction hypothesis, $\mathrm{Val}^{\mathfrak{M}}_{s_1}(t_j) = \mathrm{Val}^{\mathfrak{M}}_{s_2}(t_j)$. So,

$$\begin{aligned} \mathrm{Val}^{\mathfrak{M}}_{s_1}(t) &= \mathrm{Val}^{\mathfrak{M}}_{s_1}(f(t_1, \ldots, t_k)) = \\ &= f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_{s_1}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_k)) = \\ &= f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_{s_2}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}}_{s_2}(t_k)) = \\ &= \mathrm{Val}^{\mathfrak{M}}_{s_2}(f(t_1, \ldots, t_k)) = \mathrm{Val}^{\mathfrak{M}}_{s_2}(t). \end{aligned}$$ □

Proposition 3.14. *If the free variables in φ are among $x_1, \ldots, x_n$, and $s_1(x_i) = s_2(x_i)$ for $i = 1, \ldots, n$, then $\mathfrak{M}, s_1 \vDash \varphi$ iff $\mathfrak{M}, s_2 \vDash \varphi$.* fol:syn:ass:prop:satindep

Proof. We use induction on the complexity of φ. For the base case, where φ is atomic, φ can be: $\top$, $\bot$, $R(t_1, \dots, t_k)$ for a k-place predicate R and terms t_1, $\dots$, t_k, or $t_1 = t_2$ for terms t_1 and t_2.

1. $\varphi \equiv \top$: both $\mathfrak{M}, s_1 \vDash \varphi$ and $\mathfrak{M}, s_2 \vDash \varphi$.

2. $\varphi \equiv \bot$: both $\mathfrak{M}, s_1 \nvDash \varphi$ and $\mathfrak{M}, s_2 \nvDash \varphi$.

3. $\varphi \equiv R(t_1, \dots, t_k)$: let $\mathfrak{M}, s_1 \vDash \varphi$. Then
$$\langle \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_1), \dots, \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_k) \rangle \in R^{\mathfrak{M}}.$$
For $i = 1, \dots, k$, $\mathrm{Val}^{\mathfrak{M}}_{s_1}(t_i) = \mathrm{Val}^{\mathfrak{M}}_{s_2}(t_i)$ by Proposition 3.13. So we also have $\langle \mathrm{Val}^{\mathfrak{M}}_{s_2}(t_i), \dots, \mathrm{Val}^{\mathfrak{M}}_{s_2}(t_k) \rangle \in R^{\mathfrak{M}}$.

4. $\varphi \equiv t_1 = t_2$: suppose $\mathfrak{M}, s_1 \vDash \varphi$. Then $\mathrm{Val}^{\mathfrak{M}}_{s_1}(t_1) = \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_2)$. So,
$$\begin{aligned}
\mathrm{Val}^{\mathfrak{M}}_{s_2}(t_1) &= \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_1) && \text{(by Proposition 3.13)} \\
&= \mathrm{Val}^{\mathfrak{M}}_{s_1}(t_2) && \text{(since } \mathfrak{M}, s_1 \vDash t_1 = t_2\text{)} \\
&= \mathrm{Val}^{\mathfrak{M}}_{s_2}(t_2) && \text{(by Proposition 3.13),}
\end{aligned}$$
so $\mathfrak{M}, s_2 \vDash t_1 = t_2$.

Now assume $\mathfrak{M}, s_1 \vDash \psi$ iff $\mathfrak{M}, s_2 \vDash \psi$ for all formulas ψ less complex than φ. The induction step proceeds by cases determined by the main operator of φ. In each case, we only demonstrate the forward direction of the biconditional; the proof of the reverse direction is symmetrical. In all cases except those for the quantifiers, we apply the induction hypothesis to sub-formulas ψ of φ. The free variables of ψ are among those of φ. Thus, if s_1 and s_2 agree on the free variables of φ, they also agree on those of ψ, and the induction hypothesis applies to ψ.

1. $\varphi \equiv \neg\psi$: if $\mathfrak{M}, s_1 \vDash \varphi$, then $\mathfrak{M}, s_1 \nvDash \psi$, so by the induction hypothesis, $\mathfrak{M}, s_2 \nvDash \psi$, hence $\mathfrak{M}, s_2 \vDash \varphi$.

2. $\varphi \equiv \psi \land \chi$: if $\mathfrak{M}, s_1 \vDash \varphi$, then $\mathfrak{M}, s_1 \vDash \psi$ and $\mathfrak{M}, s_1 \vDash \chi$, so by induction hypothesis, $\mathfrak{M}, s_2 \vDash \psi$ and $\mathfrak{M}, s_2 \vDash \chi$. Hence, $\mathfrak{M}, s_2 \vDash \varphi$.

3. $\varphi \equiv \psi \lor \chi$: if $\mathfrak{M}, s_1 \vDash \varphi$, then $\mathfrak{M}, s_1 \vDash \psi$ or $\mathfrak{M}, s_1 \vDash \chi$. By induction hypothesis, $\mathfrak{M}, s_2 \vDash \psi$ or $\mathfrak{M}, s_2 \vDash \chi$, so $\mathfrak{M}, s_2 \vDash \varphi$.

4. $\varphi \equiv \psi \rightarrow \chi$: if $\mathfrak{M}, s_1 \vDash \varphi$, then $\mathfrak{M}, s_1 \nvDash \psi$ or $\mathfrak{M}, s_1 \vDash \chi$. By the induction hypothesis, $\mathfrak{M}, s_2 \nvDash \psi$ or $\mathfrak{M}, s_2 \vDash \chi$, so $\mathfrak{M}, s_2 \vDash \varphi$.

5. $\varphi \equiv \psi \leftrightarrow \chi$: if $\mathfrak{M}, s_1 \vDash \varphi$, then either $\mathfrak{M}, s_1 \vDash \psi$ and $\mathfrak{M}, s_1 \vDash \chi$, or $\mathfrak{M}, s_1 \nvDash \psi$ and $\mathfrak{M}, s_1 \nvDash \chi$. By the induction hypothesis, either $\mathfrak{M}, s_2 \vDash \psi$ and $\mathfrak{M}, s_2 \vDash \chi$ or $\mathfrak{M}, s_2 \nvDash \psi$ and $\mathfrak{M}, s_2 \nvDash \chi$. In either case, $\mathfrak{M}, s_2 \vDash \varphi$.

6. $\varphi \equiv \exists x\, \psi$: if $\mathfrak{M}, s_1 \vDash \varphi$, there is an $m \in |\mathfrak{M}|$ so that $\mathfrak{M}, s_1[m/x] \vDash \psi$. Let $s_1' = s_1[m/x]$ and $s_2' = s_2[m/x]$. The free variables of ψ are among x_1, …, x_n, and x. $s_1'(x_i) = s_2'(x_i)$, since s_1' and s_2' are x-variants of s_1 and s_2, respectively, and by hypothesis $s_1(x_i) = s_2(x_i)$. $s_1'(x) = s_2'(x) = m$ by the way we have defined s_1' and s_2'. Then the induction hypothesis applies to ψ and s_1', s_2', so $\mathfrak{M}, s_2' \vDash \psi$. Hence, since $s_2' = s_2[m/x]$, there is an $m \in |\mathfrak{M}|$ such that $\mathfrak{M}, s_2[m/x] \vDash \psi$, and so $\mathfrak{M}, s_2 \vDash \varphi$.

7. $\varphi \equiv \forall x\, \psi$: if $\mathfrak{M}, s_1 \vDash \varphi$, then for every $m \in |\mathfrak{M}|$, $\mathfrak{M}, s_1[m/x] \vDash \psi$. We want to show that also, for every $m \in |\mathfrak{M}|$, $\mathfrak{M}, s_2[m/x] \vDash \psi$. So let $m \in |\mathfrak{M}|$ be arbitrary, and consider $s_1' = s[m/x]$ and $s_2' = s[m/x]$. We have that $\mathfrak{M}, s_1' \vDash \psi$. The free variables of ψ are among x_1, …, x_n, and x. $s_1'(x_i) = s_2'(x_i)$, since s_1' and s_2' are x-variants of s_1 and s_2, respectively, and by hypothesis $s_1(x_i) = s_2(x_i)$. $s_1'(x) = s_2'(x) = m$ by the way we have defined s_1' and s_2'. Then the induction hypothesis applies to ψ and s_1', s_2', and we have $\mathfrak{M}, s_2' \vDash \psi$. This applies to every $m \in |\mathfrak{M}|$, i.e., $\mathfrak{M}, s_2[m/x] \vDash \psi$ for all $m \in |\mathfrak{M}|$, so $\mathfrak{M}, s_2 \vDash \varphi$.

By induction, we get that $\mathfrak{M}, s_1 \vDash \varphi$ iff $\mathfrak{M}, s_2 \vDash \varphi$ whenever the free variables in φ are among x_1, …, x_n and $s_1(x_i) = s_2(x_i)$ for $i = 1$, …, n. □

Problem 3.3. Complete the proof of Proposition 3.14.

explanation

Sentences have no free variables, so any two variable assignments assign the same things to all the (zero) free variables of any sentence. The proposition just proved then means that whether or not a sentence is satisfied in a structure relative to a variable assignment is completely independent of the assignment. We'll record this fact. It justifies the definition of satisfaction of a sentence in a structure (without mentioning a variable assignment) that follows.

Corollary 3.15. *If φ is a sentence and s a variable assignment, then $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}, s' \vDash \varphi$ for every variable assignment s'.* fol:syn:ass:cor:sat-sentence

Proof. Let s' be any variable assignment. Since φ is a sentence, it has no free variables, and so every variable assignment s' trivially assigns the same things to all free variables of φ as does s. So the condition of Proposition 3.14 is satisfied, and we have $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}, s' \vDash \varphi$. □

Definition 3.16. If φ is a sentence, we say that a structure $\mathfrak{M}$ *satisfies* φ, $\mathfrak{M} \vDash \varphi$, iff $\mathfrak{M}, s \vDash \varphi$ for all variable assignments s. fol:syn:ass:defn:satisfaction

If $\mathfrak{M} \vDash \varphi$, we also simply say that φ *is true in* $\mathfrak{M}$.

Proposition 3.17. *Let $\mathfrak{M}$ be a structure, φ be a sentence, and s a variable assignment. $\mathfrak{M} \vDash \varphi$ iff $\mathfrak{M}, s \vDash \varphi$.* fol:syn:ass:prop:sentence-sat-true

Proof. Exercise. □

Problem 3.4. Prove Proposition 3.17

fol:syn:ass:prop:sat-quant **Proposition 3.18.** *Suppose $\varphi(x)$ only contains x free, and $\mathfrak{M}$ is a structure. Then:*

1. *$\mathfrak{M} \vDash \exists x\, \varphi(x)$ iff $\mathfrak{M}, s \vDash \varphi(x)$ for at least one variable assignment s.*
2. *$\mathfrak{M} \vDash \forall x\, \varphi(x)$ iff $\mathfrak{M}, s \vDash \varphi(x)$ for all variable assignments s.*

Proof. Exercise. □

Problem 3.5. Prove Proposition 3.18.

Problem 3.6. Suppose $\mathcal{L}$ is a language without function symbols. Given a structure $\mathfrak{M}$, c a constant symbol and $a \in |\mathfrak{M}|$, define $\mathfrak{M}[a/c]$ to be the structure that is just like $\mathfrak{M}$, except that $c^{\mathfrak{M}[a/c]} = a$. Define $\mathfrak{M} \mid\!\vDash \varphi$ for sentences φ by:

1. $\varphi \equiv \bot$: not $\mathfrak{M} \mid\!\vDash \varphi$.
2. $\varphi \equiv \top$: $\mathfrak{M} \mid\!\vDash \varphi$.
3. $\varphi \equiv R(d_1, \dots, d_n)$: $\mathfrak{M} \mid\!\vDash \varphi$ iff $\langle d_1^{\mathfrak{M}}, \dots, d_n^{\mathfrak{M}} \rangle \in R^{\mathfrak{M}}$.
4. $\varphi \equiv d_1 = d_2$: $\mathfrak{M} \mid\!\vDash \varphi$ iff $d_1^{\mathfrak{M}} = d_2^{\mathfrak{M}}$.
5. $\varphi \equiv \lnot\psi$: $\mathfrak{M} \mid\!\vDash \varphi$ iff not $\mathfrak{M} \mid\!\vDash \psi$.
6. $\varphi \equiv (\psi \land \chi)$: $\mathfrak{M} \mid\!\vDash \varphi$ iff $\mathfrak{M} \mid\!\vDash \psi$ and $\mathfrak{M} \mid\!\vDash \chi$.
7. $\varphi \equiv (\psi \lor \chi)$: $\mathfrak{M} \mid\!\vDash \varphi$ iff $\mathfrak{M} \mid\!\vDash \psi$ or $\mathfrak{M} \mid\!\vDash \chi$ (or both).
8. $\varphi \equiv (\psi \to \chi)$: $\mathfrak{M} \mid\!\vDash \varphi$ iff not $\mathfrak{M} \mid\!\vDash \psi$ or $\mathfrak{M} \mid\!\vDash \chi$ (or both).
9. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M} \mid\!\vDash \varphi$ iff either both $\mathfrak{M} \mid\!\vDash \psi$ and $\mathfrak{M} \mid\!\vDash \chi$, or neither $\mathfrak{M} \mid\!\vDash \psi$ nor $\mathfrak{M} \mid\!\vDash \chi$.
10. $\varphi \equiv \forall x\, \psi$: $\mathfrak{M} \mid\!\vDash \varphi$ iff for all $a \in |\mathfrak{M}|$, $\mathfrak{M}[a/c] \mid\!\vDash \psi[c/x]$, if c does not occur in ψ.
11. $\varphi \equiv \exists x\, \psi$: $\mathfrak{M} \mid\!\vDash \varphi$ iff there is an $a \in |\mathfrak{M}|$ such that $\mathfrak{M}[a/c] \mid\!\vDash \psi[c/x]$, if c does not occur in ψ.

Let x_1, …, x_n be all free variables in φ, c_1, …, c_n constant symbols not in φ, a_1, …, $a_n \in |\mathfrak{M}|$, and $s(x_i) = a_i$.

Show that $\mathfrak{M}, s \vDash \varphi$ iff $\mathfrak{M}[a_1/c_1, \dots, a_n/c_n] \mid\!\vDash \varphi[c_1/x_1] \dots [c_n/x_n]$.

(This problem shows that it is possible to give a semantics for first-order logic that makes do without variable assignments.)

Problem 3.7. Suppose that f is a function symbol not in $\varphi(x, y)$. Show that there is a structure $\mathfrak{M}$ such that $\mathfrak{M} \vDash \forall x\, \exists y\, \varphi(x, y)$ iff there is an $\mathfrak{M}'$ such that $\mathfrak{M}' \vDash \forall x\, \varphi(x, f(x))$.

(This problem is a special case of what's known as Skolem's Theorem; $\forall x\, \varphi(x, f(x))$ is called a *Skolem normal form* of $\forall x\, \exists y\, \varphi(x, y)$.)

3.6 Extensionality

explanation Extensionality, sometimes called relevance, can be expressed informally as follows: the only factors that bear upon the satisfaction of formula φ in a structure $\mathfrak{M}$ relative to a variable assignment s, are the size of the domain and the assignments made by $\mathfrak{M}$ and s to the elements of the language that actually appear in φ. fol:syn:ext:sec

One immediate consequence of extensionality is that where two structures $\mathfrak{M}$ and $\mathfrak{M}'$ agree on all the elements of the language appearing in a sentence φ and have the same domain, $\mathfrak{M}$ and $\mathfrak{M}'$ must also agree on whether or not φ itself is true.

Proposition 3.19 (Extensionality). *Let φ be a formula, and $\mathfrak{M}_1$ and $\mathfrak{M}_2$ be structures with $|\mathfrak{M}_1| = |\mathfrak{M}_2|$, and s a variable assignment on $|\mathfrak{M}_1| = |\mathfrak{M}_2|$. If $c^{\mathfrak{M}_1} = c^{\mathfrak{M}_2}$, $R^{\mathfrak{M}_1} = R^{\mathfrak{M}_2}$, and $f^{\mathfrak{M}_1} = f^{\mathfrak{M}_2}$ for every constant symbol c, relation symbol R, and function symbol f occurring in φ, then $\mathfrak{M}_1, s \vDash \varphi$ iff $\mathfrak{M}_2, s \vDash \varphi$.* *fol:syn:ext:prop:extensionality*

Proof. First prove (by induction on t) that for every term, $\mathrm{Val}_s^{\mathfrak{M}_1}(t) = \mathrm{Val}_s^{\mathfrak{M}_2}(t)$. Then prove the proposition by induction on φ, making use of the claim just proved for the induction basis (where φ is atomic). □

Problem 3.8. Carry out the proof of Proposition 3.19 in detail.

Corollary 3.20 (Extensionality for Sentences). *Let φ be a sentence and $\mathfrak{M}_1$, $\mathfrak{M}_2$ as in Proposition 3.19. Then $\mathfrak{M}_1 \vDash \varphi$ iff $\mathfrak{M}_2 \vDash \varphi$.* *fol:syn:ext:cor:extensionality-sent*

Proof. Follows from Proposition 3.19 by Corollary 3.15. □

Moreover, the value of a term, and whether or not a structure satisfies a formula, only depend on the values of its subterms.

Proposition 3.21. *Let $\mathfrak{M}$ be a structure, t and t' terms, and s a variable assignment. Then $\mathrm{Val}_s^{\mathfrak{M}}(t[t'/x]) = \mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}_s^{\mathfrak{M}}(t')/x]}(t)$.* *fol:syn:ext:prop:ext-terms*

Proof. By induction on t.

1. If t is a constant, say, $t \equiv c$, then $t[t'/x] = c$, and $\mathrm{Val}_s^{\mathfrak{M}}(c) = c^{\mathfrak{M}} = \mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}_s^{\mathfrak{M}}(t')/x]}(c)$.

2. If t is a variable other than x, say, $t \equiv y$, then $t[t'/x] = y$, and $\mathrm{Val}_s^{\mathfrak{M}}(y) = \mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}_s^{\mathfrak{M}}(t')/x]}(y)$ since $s \sim_x s[\mathrm{Val}_s^{\mathfrak{M}}(t')/x]$.

3. If $t \equiv x$, then $t[t'/x] = t'$. But $\mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}_s^{\mathfrak{M}}(t')/x]}(x) = \mathrm{Val}_s^{\mathfrak{M}}(t')$ by definition of $s[\mathrm{Val}_s^{\mathfrak{M}}(t')/x]$.

4. If $t \equiv f(t_1, \dots, t_n)$ then we have:

$$\begin{aligned}
&\mathrm{Val}^{\mathfrak{M}}_s(t[t'/x]) = \\
&\qquad = \mathrm{Val}^{\mathfrak{M}}_s(f(t_1[t'/x], \dots, t_n[t'/x])) \\
&\qquad\qquad \text{by definition of } t[t'/x] \\
&\qquad = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_s(t_1[t'/x]), \dots, \mathrm{Val}^{\mathfrak{M}}_s(t_n[t'/x])) \\
&\qquad\qquad \text{by definition of } \mathrm{Val}^{\mathfrak{M}}_s(f(\dots)) \\
&\qquad = f^{\mathfrak{M}}(\mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]}(t_1), \dots, \mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]}(t_n)) \\
&\qquad\qquad \text{by induction hypothesis} \\
&\qquad = \mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]}(t) \text{ by definition of } \mathrm{Val}^{\mathfrak{M}}_{s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x]}(f(\dots)) \qquad \square
\end{aligned}$$

fol:syn:ext:prop:ext-formulas **Proposition 3.22.** *Let $\mathfrak{M}$ be a structure, φ a formula, t' a term, and s a variable assignment. Then $\mathfrak{M}, s \vDash \varphi[t'/x]$ iff $\mathfrak{M}, s[\mathrm{Val}^{\mathfrak{M}}_s(t')/x] \vDash \varphi$.*

Proof. Exercise. □

Problem 3.9. Prove Proposition 3.22

The point of Propositions 3.21 and 3.22 is the following. Suppose we have a term t or a formula φ and some term t', and we want to know the value of $t[t'/x]$ or whether or not $\varphi[t'/x]$ is satisfied in a structure $\mathfrak{M}$ relative to a variable assignment s. Then we can either perform the substitution first and then consider the value or satisfaction relative to $\mathfrak{M}$ and s, or we can first determine the value $m = \mathrm{Val}^{\mathfrak{M}}_s(t')$ of t' in $\mathfrak{M}$ relative to s, change the variable assignment to $s[m/x]$ and then consider the value of t in $\mathfrak{M}$ and $s[m/x]$, or whether $\mathfrak{M}, s[m/x] \vDash \varphi$. Propositions 3.21 and 3.22 guarantee that the answer will be the same, whichever way we do it. explanation

3.7 Semantic Notions

fol:syn:sem:sec Given the definition of structures for first-order languages, we can define some basic semantic properties of and relationships between sentences. The simplest of these is the notion of *validity* of a sentence. A sentence is valid if it is satisfied in every structure. Valid sentences are those that are satisfied regardless of how the non-logical symbols in it are interpreted. Valid sentences are therefore also called *logical truths*—they are true, i.e., satisfied, in any structure and hence their truth depends only on the logical symbols occurring in them and their syntactic structure, but not on the non-logical symbols or their interpretation. explanation

Definition 3.23 (Validity). A sentence φ is *valid*, $\vDash \varphi$, iff $\mathfrak{M} \vDash \varphi$ for every structure $\mathfrak{M}$.

Definition 3.24 (Entailment). A set of sentences Γ *entails* a sentence φ, $\Gamma \vDash \varphi$, iff for every structure $\mathfrak{M}$ with $\mathfrak{M} \vDash \Gamma$, $\mathfrak{M} \vDash \varphi$.

Definition 3.25 (Satisfiability). A set of sentences Γ is *satisfiable* if $\mathfrak{M} \vDash \Gamma$ for some structure $\mathfrak{M}$. If Γ is not satisfiable it is called *unsatisfiable.*

Proposition 3.26. *A sentence φ is valid iff $\Gamma \vDash \varphi$ for every set of sentences Γ.*

Proof. For the forward direction, let φ be valid, and let Γ be a set of sentences. Let $\mathfrak{M}$ be a structure so that $\mathfrak{M} \vDash \Gamma$. Since φ is valid, $\mathfrak{M} \vDash \varphi$, hence $\Gamma \vDash \varphi$.

For the contrapositive of the reverse direction, let φ be invalid, so there is a structure $\mathfrak{M}$ with $\mathfrak{M} \nvDash \varphi$. When $\Gamma = \{\top\}$, since $\top$ is valid, $\mathfrak{M} \vDash \Gamma$. Hence, there is a structure $\mathfrak{M}$ so that $\mathfrak{M} \vDash \Gamma$ but $\mathfrak{M} \nvDash \varphi$, hence Γ does not entail φ. □

Proposition 3.27. $\Gamma \vDash \varphi$ *iff* $\Gamma \cup \{\neg\varphi\}$ *is unsatisfiable.* *fol:syn:sem:prop:entails-unsat*

Proof. For the forward direction, suppose $\Gamma \vDash \varphi$ and suppose to the contrary that there is a structure $\mathfrak{M}$ so that $\mathfrak{M} \vDash \Gamma \cup \{\neg\varphi\}$. Since $\mathfrak{M} \vDash \Gamma$ and $\Gamma \vDash \varphi$, $\mathfrak{M} \vDash \varphi$. Also, since $\mathfrak{M} \vDash \Gamma \cup \{\neg\varphi\}$, $\mathfrak{M} \vDash \neg\varphi$, so we have both $\mathfrak{M} \vDash \varphi$ and $\mathfrak{M} \nvDash \varphi$, a contradiction. Hence, there can be no such structure $\mathfrak{M}$, so $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable.

For the reverse direction, suppose $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable. So for every structure $\mathfrak{M}$, either $\mathfrak{M} \nvDash \Gamma$ or $\mathfrak{M} \vDash \varphi$. Hence, for every structure $\mathfrak{M}$ with $\mathfrak{M} \vDash \Gamma$, $\mathfrak{M} \vDash \varphi$, so $\Gamma \vDash \varphi$. □

Problem 3.10.

1. Show that $\Gamma \vDash \bot$ iff Γ is unsatisfiable.
2. Show that $\Gamma \cup \{\varphi\} \vDash \bot$ iff $\Gamma \vDash \neg\varphi$.
3. Suppose c does not occur in φ or Γ. Show that $\Gamma \vDash \forall x\, \varphi$ iff $\Gamma \vDash \varphi[c/x]$.

Proposition 3.28. *If $\Gamma \subseteq \Gamma'$ and $\Gamma \vDash \varphi$, then $\Gamma' \vDash \varphi$.*

Proof. Suppose that $\Gamma \subseteq \Gamma'$ and $\Gamma \vDash \varphi$. Let $\mathfrak{M}$ be a structure such that $\mathfrak{M} \vDash \Gamma'$; then $\mathfrak{M} \vDash \Gamma$, and since $\Gamma \vDash \varphi$, we get that $\mathfrak{M} \vDash \varphi$. Hence, whenever $\mathfrak{M} \vDash \Gamma'$, $\mathfrak{M} \vDash \varphi$, so $\Gamma' \vDash \varphi$. □

Theorem 3.29 (Semantic Deduction Theorem). $\Gamma \cup \{\varphi\} \vDash \psi$ *iff* $\Gamma \vDash \varphi \to \psi$. *fol:syn:sem:thm:sem-deduction*

Proof. For the forward direction, let $\Gamma \cup \{\varphi\} \vDash \psi$ and let $\mathfrak{M}$ be a structure so that $\mathfrak{M} \vDash \Gamma$. If $\mathfrak{M} \vDash \varphi$, then $\mathfrak{M} \vDash \Gamma \cup \{\varphi\}$, so since $\Gamma \cup \{\varphi\}$ entails ψ, we get $\mathfrak{M} \vDash \psi$. Therefore, $\mathfrak{M} \vDash \varphi \to \psi$, so $\Gamma \vDash \varphi \to \psi$.

For the reverse direction, let $\Gamma \vDash \varphi \to \psi$ and $\mathfrak{M}$ be a structure so that $\mathfrak{M} \vDash \Gamma \cup \{\varphi\}$. Then $\mathfrak{M} \vDash \Gamma$, so $\mathfrak{M} \vDash \varphi \to \psi$, and since $\mathfrak{M} \vDash \varphi$, $\mathfrak{M} \vDash \psi$. Hence, whenever $\mathfrak{M} \vDash \Gamma \cup \{\varphi\}$, $\mathfrak{M} \vDash \psi$, so $\Gamma \cup \{\varphi\} \vDash \psi$. □

fol:syn:sem:prop:quant-terms **Proposition 3.30.** *Let $\mathfrak{M}$ be a structure, and $\varphi(x)$ a formula with one free variable x, and t a closed term. Then:*

1. $\varphi(t) \vDash \exists x\, \varphi(x)$
2. $\forall x\, \varphi(x) \vDash \varphi(t)$

Proof.

1. Suppose $\mathfrak{M} \vDash \varphi(t)$. Let s be a variable assignment with $s(x) = \mathrm{Val}^{\mathfrak{M}}(t)$. Then $\mathfrak{M}, s \vDash \varphi(t)$ since $\varphi(t)$ is a sentence. By Proposition 3.22, $\mathfrak{M}, s \vDash \varphi(x)$. By Proposition 3.18, $\mathfrak{M} \vDash \exists x\, \varphi(x)$.

2. Suppose $\mathfrak{M} \vDash \forall x\, \varphi(x)$. Let s be a variable assignment with $s(x) = \mathrm{Val}^{\mathfrak{M}}(t)$. By Proposition 3.18, $\mathfrak{M}, s \vDash \varphi(x)$. By Proposition 3.22, $\mathfrak{M}, s \vDash \varphi(t)$. By Proposition 3.17, $\mathfrak{M} \vDash \varphi(t)$ since $\varphi(t)$ is a sentence. □

Problem 3.11. Complete the proof of Proposition 3.30.

Chapter 4

Theories and Their Models

4.1 Introduction

explanation The development of the axiomatic method is a significant achievement in the history of science, and is of special importance in the history of mathematics. An axiomatic development of a field involves the clarification of many questions: What is the field about? What are the most fundamental concepts? How are they related? Can all the concepts of the field be defined in terms of these fundamental concepts? What laws do, and must, these concepts obey? fol:mat:int:sec

The axiomatic method and logic were made for each other. Formal logic provides the tools for formulating axiomatic theories, for proving theorems from the axioms of the theory in a precisely specified way, for studying the properties of all systems satisfying the axioms in a systematic way.

Definition 4.1. A set of sentences Γ is *closed* iff, whenever $\Gamma \vDash \varphi$ then $\varphi \in \Gamma$. The *closure* of a set of sentences Γ is $\{\varphi : \Gamma \vDash \varphi\}$.

We say that Γ is *axiomatized by* a set of sentences Δ if Γ is the closure of Δ.

explanation We can think of an axiomatic theory as the set of sentences that is axiomatized by its set of axioms Δ. In other words, when we have a first-order language which contains non-logical symbols for the primitives of the axiomatically developed science we wish to study, together with a set of sentences that express the fundamental laws of the science, we can think of the theory as represented by all the sentences in this language that are entailed by the axioms. This ranges from simple examples with only a single primitive and simple axioms, such as the theory of partial orders, to complex theories such as Newtonian mechanics.

The important logical facts that make this formal approach to the axiomatic method so important are the following. Suppose Γ is an axiom system for a theory, i.e., a set of sentences.

1. We can state precisely when an axiom system captures an intended class of structures. That is, if we are interested in a certain class of struc-

tures, we will successfully capture that class by an axiom system Γ iff the structures are exactly those $\mathfrak{M}$ such that $\mathfrak{M} \vDash \Gamma$.

2. We may fail in this respect because there are $\mathfrak{M}$ such that $\mathfrak{M} \vDash \Gamma$, but $\mathfrak{M}$ is not one of the structures we intend. This may lead us to add axioms which are not true in $\mathfrak{M}$.

3. If we are successful at least in the respect that Γ is true in all the intended structures, then a sentence φ is true in all intended structures whenever $\Gamma \vDash \varphi$. Thus we can use logical tools (such as derivation methods) to show that sentences are true in all intended structures simply by showing that they are entailed by the axioms.

4. Sometimes we don't have intended structures in mind, but instead start from the axioms themselves: we begin with some primitives that we want to satisfy certain laws which we codify in an axiom system. One thing that we would like to verify right away is that the axioms do not contradict each other: if they do, there can be no concepts that obey these laws, and we have tried to set up an incoherent theory. We can verify that this doesn't happen by finding a model of Γ. And if there are models of our theory, we can use logical methods to investigate them, and we can also use logical methods to construct models.

5. The independence of the axioms is likewise an important question. It may happen that one of the axioms is actually a consequence of the others, and so is redundant. We can prove that an axiom φ in Γ is redundant by proving $\Gamma \setminus \{\varphi\} \vDash \varphi$. We can also prove that an axiom is not redundant by showing that $(\Gamma \setminus \{\varphi\}) \cup \{\neg\varphi\}$ is satisfiable. For instance, this is how it was shown that the parallel postulate is independent of the other axioms of geometry.

6. Another important question is that of definability of concepts in a theory: The choice of the language determines what the models of a theory consist of. But not every aspect of a theory must be represented separately in its models. For instance, every ordering $\leq$ determines a corresponding strict ordering $<$—given one, we can define the other. So it is not necessary that a model of a theory involving such an order must *also* contain the corresponding strict ordering. When is it the case, in general, that one relation can be defined in terms of others? When is it impossible to define a relation in terms of others (and hence must add it to the primitives of the language)?

4.2 Expressing Properties of Structures

fol:mat:exs:sec It is often useful and important to express conditions on functions and relations, or more generally, that the functions and relations in a structure satisfy these conditions. For instance, we would like to have ways of distinguishing those explanation structures for a language which "capture" what we want the predicate symbols to "mean" from those that do not. Of course we're completely free to specify which structures we "intend," e.g., we can specify that the interpretation of the predicate symbol $\leq$ must be an ordering, or that we are only interested in interpretations of $\mathcal{L}$ in which the domain consists of sets and $\in$ is interpreted by the "is an element of" relation. But can we do this with sentences of the language? In other words, which conditions on a structure $\mathfrak{M}$ can we express by a sentence (or perhaps a set of sentences) in the language of $\mathfrak{M}$? There are some conditions that we will not be able to express. For instance, there is no sentence of $\mathcal{L}_A$ which is only true in a structure $\mathfrak{M}$ if $|\mathfrak{M}| = \mathbb{N}$. We cannot express "the domain contains only natural numbers." But there are "structural properties" of structures that we perhaps can express. Which properties of structures can we express by sentences? Or, to put it another way, which collections of structures can we describe as those making a sentence (or set of sentences) true?

Definition 4.2 (Model of a set). Let Γ be a set of sentences in a language $\mathcal{L}$. We say that a structure $\mathfrak{M}$ *is a model of* Γ if $\mathfrak{M} \vDash \varphi$ for all $\varphi \in \Gamma$.

Example 4.3. The sentence $\forall x\, x \leq x$ is true in $\mathfrak{M}$ iff $\leq^{\mathfrak{M}}$ is a reflexive relation. The sentence $\forall x\, \forall y\, ((x \leq y \land y \leq x) \rightarrow x = y)$ is true in $\mathfrak{M}$ iff $\leq^{\mathfrak{M}}$ is anti-symmetric. The sentence $\forall x\, \forall y\, \forall z\, ((x \leq y \land y \leq z) \rightarrow x \leq z)$ is true in $\mathfrak{M}$ iff $\leq^{\mathfrak{M}}$ is transitive. Thus, the models of

$$\begin{aligned} \{ \quad & \forall x\, x \leq x, \\ & \forall x\, \forall y\, ((x \leq y \land y \leq x) \rightarrow x = y), \\ & \forall x\, \forall y\, \forall z\, ((x \leq y \land y \leq z) \rightarrow x \leq z) \quad \} \end{aligned}$$

are exactly those structures in which $\leq^{\mathfrak{M}}$ is reflexive, anti-symmetric, and transitive, i.e., a partial order. Hence, we can take them as axioms for the *first-order theory of partial orders.*

4.3 Examples of First-Order Theories

fol:mat:the:sec

Example 4.4. The theory of strict linear orders in the language $\mathcal{L}_<$ is axiomatized by the set

$$\begin{aligned} \{ \quad & \forall x\, \neg x < x, \\ & \forall x\, \forall y\, ((x < y \lor y < x) \lor x = y), \\ & \forall x\, \forall y\, \forall z\, ((x < y \land y < z) \rightarrow x < z) \quad \} \end{aligned}$$

It completely captures the intended structures: every strict linear order is a model of this axiom system, and vice versa, if R is a linear order on a set X, then the structure $\mathfrak{M}$ with $|\mathfrak{M}| = X$ and $<^{\mathfrak{M}} = R$ is a model of this theory.

Example 4.5. The theory of groups in the language $\mathsf{1}$ (constant symbol), $\cdot$ (two-place function symbol) is axiomatized by

$$\begin{aligned}
&\forall x\,(x \cdot \mathsf{1}) = x\\
&\forall x\,\forall y\,\forall z\,(x \cdot (y \cdot z)) = ((x \cdot y) \cdot z)\\
&\forall x\,\exists y\,(x \cdot y) = \mathsf{1}
\end{aligned}$$

Example 4.6. The theory of Peano arithmetic is axiomatized by the following sentences in the language of arithmetic $\mathcal{L}_A$.

$$\begin{aligned}
&\forall x\,\forall y\,(x' = y' \to x = y)\\
&\forall x\,\mathsf{0} \neq x'\\
&\forall x\,(x + \mathsf{0}) = x\\
&\forall x\,\forall y\,(x + y') = (x + y)'\\
&\forall x\,(x \times \mathsf{0}) = \mathsf{0}\\
&\forall x\,\forall y\,(x \times y') = ((x \times y) + x)\\
&\forall x\,\forall y\,(x < y \leftrightarrow \exists z\,(z' + x) = y)
\end{aligned}$$

plus all sentences of the form

$$(\varphi(\mathsf{0}) \land \forall x\,(\varphi(x) \to \varphi(x'))) \to \forall x\,\varphi(x)$$

Since there are infinitely many sentences of the latter form, this axiom system is infinite. The latter form is called the *induction schema*. (Actually, the induction schema is a bit more complicated than we let on here.)

The last axiom is an *explicit definition* of $<$.

Example 4.7. The theory of pure sets plays an important role in the foundations (and in the philosophy) of mathematics. A set is pure if all its elements are also pure sets. The empty set counts therefore as pure, but a set that has something as an element that is not a set would not be pure. So the pure sets are those that are formed just from the empty set and no "urelements," i.e., objects that are not themselves sets.

The following might be considered as an axiom system for a theory of pure sets:

$$\begin{aligned}
&\exists x\,\neg\exists y\,y \in x\\
&\forall x\,\forall y\,(\forall z(z \in x \leftrightarrow z \in y) \to x = y)\\
&\forall x\,\forall y\,\exists z\,\forall u\,(u \in z \leftrightarrow (u = x \lor u = y))\\
&\forall x\,\exists y\,\forall z\,(z \in y \leftrightarrow \exists u\,(z \in u \land u \in x))
\end{aligned}$$

plus all sentences of the form

$$\exists x\,\forall y\,(y \in x \leftrightarrow \varphi(y))$$

The first axiom says that there is a set with no elements (i.e., $\emptyset$ exists); the second says that sets are extensional; the third that for any sets X and Y, the set $\{X, Y\}$ exists; the fourth that for any set X, the set $\cup X$ exists, where $\cup X$ is the union of all the elements of X.

The sentences mentioned last are collectively called the *naive comprehension scheme*. It essentially says that for every $\varphi(x)$, the set $\{x : \varphi(x)\}$ exists—so at first glance a true, useful, and perhaps even necessary axiom. It is called "naive" because, as it turns out, it makes this theory unsatisfiable: if you take $\varphi(y)$ to be $\neg y \in y$, you get the sentence

$$\exists x\, \forall y\, (y \in x \leftrightarrow \neg y \in y)$$

and this sentence is not satisfied in any structure.

Example 4.8. In the area of *mereology*, the relation of *parthood* is a fundamental relation. Just like theories of sets, there are theories of parthood that axiomatize various conceptions (sometimes conflicting) of this relation.

The language of mereology contains a single two-place predicate symbol P, and $\mathsf{P}(x, y)$ "means" that x is a part of y. When we have this interpretation in mind, a structure for this language is called a *parthood structure*. Of course, not every structure for a single two-place predicate will really deserve this name. To have a chance of capturing "parthood," $\mathsf{P}^{\mathfrak{M}}$ must satisfy some conditions, which we can lay down as axioms for a theory of parthood. For instance, parthood is a partial order on objects: every object is a part (albeit an *improper* part) of itself; no two different objects can be parts of each other; a part of a part of an object is itself part of that object. Note that in this sense "is a part of" resembles "is a subset of," but does not resemble "is an element of" which is neither reflexive nor transitive.

$$\begin{aligned}
&\forall x\, \mathsf{P}(x, x)\\
&\forall x\, \forall y\, ((\mathsf{P}(x, y) \land \mathsf{P}(y, x)) \to x = y)\\
&\forall x\, \forall y\, \forall z\, ((\mathsf{P}(x, y) \land \mathsf{P}(y, z)) \to \mathsf{P}(x, z))
\end{aligned}$$

Moreover, any two objects have a mereological sum (an object that has these two objects as parts, and is minimal in this respect).

$$\forall x\, \forall y\, \exists z\, \forall u\, (\mathsf{P}(z, u) \leftrightarrow (\mathsf{P}(x, u) \land \mathsf{P}(y, u)))$$

These are only some of the basic principles of parthood considered by metaphysicians. Further principles, however, quickly become hard to formulate or write down without first introducing some defined relations. For instance, most metaphysicians interested in mereology also view the following as a valid principle: whenever an object x has a proper part y, it also has a part z that has no parts in common with y, and so that the fusion of y and z is x.

fol:mat:exr:sec

4.4 Expressing Relations in a Structure

One main use formulas can be put to is to express properties and relations in a structure $\mathfrak{M}$ in terms of the primitives of the language $\mathcal{L}$ of $\mathfrak{M}$. By this we mean the following: the domain of $\mathfrak{M}$ is a set of objects. The constant symbols, function symbols, and predicate symbols are interpreted in $\mathfrak{M}$ by some objects in$|\mathfrak{M}|$, functions on $|\mathfrak{M}|$, and relations on $|\mathfrak{M}|$. For instance, if A^2_0 is in $\mathcal{L}$, then $\mathfrak{M}$ assigns to it a relation $R = A^{2\,\mathfrak{M}}_0$. Then the formula $A^2_0(v_1, v_2)$ *expresses* that very relation, in the following sense: if a variable assignment s maps v_1 to $a \in |\mathfrak{M}|$ and v_2 to $b \in |\mathfrak{M}|$, then explanation

$$Rab \quad \text{iff} \quad \mathfrak{M}, s \vDash A^2_0(v_1, v_2).$$

Note that we have to involve variable assignments here: we can't just say "Rab iff $\mathfrak{M} \vDash A^2_0(a, b)$" because a and b are not symbols of our language: they are elements of $|\mathfrak{M}|$.

Since we don't just have atomic formulas, but can combine them using the logical connectives and the quantifiers, more complex formulas can define other relations which aren't directly built into $\mathfrak{M}$. We're interested in how to do that, and specifically, which relations we can define in a structure.

Definition 4.9. Let $\varphi(v_1, \dots, v_n)$ be a formula of $\mathcal{L}$ in which only v_1,..., v_n occur free, and let $\mathfrak{M}$ be a structure for $\mathcal{L}$. $\varphi(v_1, \dots, v_n)$ *expresses the relation* $R \subseteq |\mathfrak{M}|^n$ iff

$$Ra_1 \dots a_n \quad \text{iff} \quad \mathfrak{M}, s \vDash \varphi(v_1, \dots, v_n)$$

for any variable assignment s with $s(v_i) = a_i$ $(i = 1, \dots, n)$.

Example 4.10. In the standard model of arithmetic $\mathfrak{N}$, the formula $v_1 < v_2 \lor v_1 = v_2$ expresses the $\leq$ relation on $\mathbb{N}$. The formula $v_2 = v_1'$ expresses the successor relation, i.e., the relation $R \subseteq \mathbb{N}^2$ where Rnm holds if m is the successor of n. The formula $v_1 = v_2'$ expresses the predecessor relation. The formulas $\exists v_3\,(v_3 \neq 0 \land v_2 = (v_1 + v_3))$ and $\exists v_3\,(v_1 + v_3{}') = v_2$ both express the $<$ relation. This means that the predicate symbol $<$ is actually superfluous in the language of arithmetic; it can be defined.

This idea is not just interesting in specific structures, but generally whenever we use a language to describe an intended model or models, i.e., when we consider theories. These theories often only contain a few predicate symbols as basic symbols, but in the domain they are used to describe often many other relations play an important role. If these other relations can be systematically expressed by the relations that interpret the basic predicate symbols of the language, we say we can *define* them in the language. explanation

Problem 4.1. Find formulas in $\mathcal{L}_A$ which define the following relations:

1. n is between i and j;

2. n evenly divides m (i.e., m is a multiple of n);

3. n is a prime number (i.e., no number other than 1 and n evenly divides n).

Problem 4.2. Suppose the formula $\varphi(v_1, v_2)$ expresses the relation $R \subseteq |\mathfrak{M}|^2$ in a structure $\mathfrak{M}$. Find formulas that express the following relations:

1. the inverse R^{-1} of R;

2. the relative product $R \mid R$;

Can you find a way to express R^+, the transitive closure of R?

Problem 4.3. Let $\mathcal{L}$ be the language containing a 2-place predicate symbol $<$ only (no other constant symbols, function symbols or predicate symbols—except of course $=$). Let $\mathfrak{N}$ be the structure such that $|\mathfrak{N}| = \mathbb{N}$, and $<^{\mathfrak{N}} = \{\langle n, m\rangle : n < m\}$. Prove the following:

1. $\{0\}$ is definable in $\mathfrak{N}$;

2. $\{1\}$ is definable in $\mathfrak{N}$;

3. $\{2\}$ is definable in $\mathfrak{N}$;

4. for each $n \in \mathbb{N}$, the set $\{n\}$ is definable in $\mathfrak{N}$;

5. every finite subset of $|\mathfrak{N}|$ is definable in $\mathfrak{N}$;

6. every co-finite subset of $|\mathfrak{N}|$ is definable in $\mathfrak{N}$ (where $X \subseteq \mathbb{N}$ is co-finite iff $\mathbb{N} \setminus X$ is finite).

4.5 The Theory of Sets

Almost all of mathematics can be developed in the theory of sets. Developing mathematics in this theory involves a number of things. First, it requires a set of axioms for the relation $\in$. A number of different axiom systems have been developed, sometimes with conflicting properties of $\in$. The axiom system known as **ZFC**, Zermelo-Fraenkel set theory with the axiom of choice stands out: it is by far the most widely used and studied, because it turns out that its axioms suffice to prove almost all the things mathematicians expect to be able to prove. But before that can be established, it first is necessary to make clear how we can even *express* all the things mathematicians would like to express. For starters, the language contains no constant symbols or function symbols, so it seems at first glance unclear that we can talk about particular sets (such as $\emptyset$ or $\mathbb{N}$), can talk about operations on sets (such as $X \cup Y$ and $\wp(X)$), let alone other constructions which involve things other than sets, such as relations and functions. fol:mat:set:sec

To begin with, "is an element of" is not the only relation we are interested in: "is a subset of" seems almost as important. But we can *define* "is a subset of" in terms of "is an element of." To do this, we have to find a formula $\varphi(x, y)$ in the language of set theory which is satisfied by a pair of sets $\langle X, Y\rangle$ iff $X \subseteq Y$. But X is a subset of Y just in case all elements of X are also elements of Y. So we can define $\subseteq$ by the formula

$$\forall z\,(z \in x \to z \in y)$$

Now, whenever we want to use the relation $\subseteq$ in a formula, we could instead use that formula (with x and y suitably replaced, and the bound variable z renamed if necessary). For instance, extensionality of sets means that if any sets x and y are contained in each other, then x and y must be the same set. This can be expressed by $\forall x\,\forall y\,((x \subseteq y \land y \subseteq x) \to x = y)$, or, if we replace $\subseteq$ by the above definition, by

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

This is in fact one of the axioms of **ZFC**, the "axiom of extensionality."

There is no constant symbol for $\emptyset$, but we can express "x is empty" by $\lnot\exists y\,y \in x$. Then "$\emptyset$ exists" becomes the sentence $\exists x\,\lnot\exists y\,y \in x$. This is another axiom of **ZFC**. (Note that the axiom of extensionality implies that there is only one empty set.) Whenever we want to talk about $\emptyset$ in the language of set theory, we would write this as "there is a set that's empty and ..." As an example, to express the fact that $\emptyset$ is a subset of every set, we could write

$$\exists x\,(\lnot\exists y\,y \in x \land \forall z\,x \subseteq z)$$

where, of course, $x \subseteq z$ would in turn have to be replaced by its definition.

To talk about operations on sets, such as $X \cup Y$ and $\wp(X)$, we have to use a similar trick. There are no function symbols in the language of set theory, but we can express the functional relations $X \cup Y = Z$ and $\wp(X) = Y$ by

$$\begin{aligned}&\forall u\,((u \in x \lor u \in y) \leftrightarrow u \in z)\\ &\forall u\,(u \subseteq x \leftrightarrow u \in y)\end{aligned}$$

since the elements of $X \cup Y$ are exactly the sets that are either elements of X or elements of Y, and the elements of $\wp(X)$ are exactly the subsets of X. However, this doesn't allow us to use $x \cup y$ or $\wp(x)$ as if they were terms: we can only use the entire formulas that define the relations $X \cup Y = Z$ and $\wp(X) = Y$. In fact, we do not know that these relations are ever satisfied, i.e., we do not know that unions and power sets always exist. For instance, the sentence $\forall x\,\exists y\,\wp(x) = y$ is another axiom of **ZFC** (the power set axiom).

Now what about talk of ordered pairs or functions? Here we have to explain how we can think of ordered pairs and functions as special kinds of sets. One way to define the ordered pair $\langle x, y\rangle$ is as the set $\{\{x\}, \{x, y\}\}$. But like before, we cannot introduce a function symbol that names this set; we can only define the relation $\langle x, y\rangle = z$, i.e., $\{\{x\}, \{x, y\}\} = z$:

$$\forall u\,(u \in z \leftrightarrow (\forall v\,(v \in u \leftrightarrow v = x) \lor \forall v\,(v \in u \leftrightarrow (v = x \lor v = y))))$$

This says that the elements u of z are exactly those sets which either have x as its only element or have x and y as its only elements (in other words, those sets that are either identical to $\{x\}$ or identical to $\{x, y\}$). Once we have this, we can say further things, e.g., that $X \times Y = Z$:

$$\forall z\,(z \in Z \leftrightarrow \exists x\,\exists y\,(x \in X \land y \in Y \land \langle x, y\rangle = z))$$

A function $f\colon X \to Y$ can be thought of as the relation $f(x) = y$, i.e., as the set of pairs $\{\langle x, y\rangle : f(x) = y\}$. We can then say that a set f is a function from X to Y if (a) it is a relation $\subseteq X \times Y$, (b) it is total, i.e., for all $x \in X$ there is some $y \in Y$ such that $\langle x, y\rangle \in f$ and (c) it is functional, i.e., whenever $\langle x, y\rangle, \langle x, y'\rangle \in f$, $y = y'$ (because values of functions must be unique). So "f is a function from X to Y" can be written as:

$$\begin{aligned}
&\forall u\,(u \in f \to \exists x\,\exists y\,(x \in X \land y \in Y \land \langle x, y\rangle = u)) \land \\
&\forall x\,(x \in X \to (\exists y\,(y \in Y \land \mathrm{maps}(f, x, y)) \land \\
&\qquad\qquad (\forall y\,\forall y'\,((\mathrm{maps}(f, x, y) \land \mathrm{maps}(f, x, y')) \to y = y')))
\end{aligned}$$

where $\mathrm{maps}(f, x, y)$ abbreviates $\exists v\,(v \in f \land \langle x, y\rangle = v)$ (this formula expresses "$f(x) = y$").

It is now also not hard to express that $f\colon X \to Y$ is injective, for instance:

$$\begin{aligned}
&f\colon X \to Y \land \forall x\,\forall x'\,((x \in X \land x' \in X \land \\
&\qquad\qquad \exists y\,(\mathrm{maps}(f, x, y) \land \mathrm{maps}(f, x', y))) \to x = x')
\end{aligned}$$

A function $f\colon X \to Y$ is injective iff, whenever f maps $x, x' \in X$ to a single y, $x = x'$. If we abbreviate this formula as $\mathrm{inj}(f, X, Y)$, we're already in a position to state in the language of set theory something as non-trivial as Cantor's theorem: there is no injective function from $\wp(X)$ to X:

$$\forall X\,\forall Y\,(\wp(X) = Y \to \lnot\exists f\,\mathrm{inj}(f, Y, X))$$

One might think that set theory requires another axiom that guarantees the existence of a set for every defining property. If $\varphi(x)$ is a formula of set theory with the variable x free, we can consider the sentence

$$\exists y\,\forall x\,(x \in y \leftrightarrow \varphi(x)).$$

This sentence states that there is a set y whose elements are all and only those x that satisfy $\varphi(x)$. This schema is called the "comprehension principle." It looks very useful; unfortunately it is inconsistent. Take $\varphi(x) \equiv \lnot x \in x$, then the comprehension principle states

$$\exists y\,\forall x\,(x \in y \leftrightarrow x \notin x),$$

i.e., it states the existence of a set of all sets that are not elements of themselves. No such set can exist—this is Russell's Paradox. **ZFC**, in fact, contains a restricted—and consistent—version of this principle, the separation principle:

$$\forall z\,\exists y\,\forall x\,(x \in y \leftrightarrow (x \in z \land \varphi(x))).$$

Problem 4.4. Show that the comprehension principle is inconsistent by giving a derivation that shows

$$\exists y\, \forall x\, (x \in y \leftrightarrow x \notin x) \vdash \bot.$$

It may help to first show $(A \to \neg A) \land (\neg A \to A) \vdash \bot$.

4.6 Expressing the Size of Structures

fol:mat:siz:sec There are some properties of structures we can express even without using the non-logical symbols of a language. For instance, there are sentences which are true in a structure iff the domain of the structure has at least, at most, or exactly a certain number n of elements. explanation

Proposition 4.11. *The sentence*

$$\begin{aligned}\varphi_{\geq n} \equiv \exists x_1\, \exists x_2\, \dots \exists x_n \\ (x_1 \neq x_2 \land x_1 \neq x_3 \land x_1 \neq x_4 \land \dots \land x_1 \neq x_n \land \\ x_2 \neq x_3 \land x_2 \neq x_4 \land \dots \land x_2 \neq x_n \land \\ \vdots \\ x_{n-1} \neq x_n)\end{aligned}$$

is true in a structure $\mathfrak{M}$ iff $|\mathfrak{M}|$ contains at least n elements. Consequently, $\mathfrak{M} \vDash \neg\varphi_{\geq n+1}$ iff $|\mathfrak{M}|$ contains at most n elements.

Proposition 4.12. *The sentence*

$$\begin{aligned}\varphi_{= n} \equiv \exists x_1\, \exists x_2\, \dots \exists x_n \\ (x_1 \neq x_2 \land x_1 \neq x_3 \land x_1 \neq x_4 \land \dots \land x_1 \neq x_n \land \\ x_2 \neq x_3 \land x_2 \neq x_4 \land \dots \land x_2 \neq x_n \land \\ \vdots \\ x_{n-1} \neq x_n \land \\ \forall y\, (y = x_1 \lor \dots \lor y = x_n))\end{aligned}$$

is true in a structure $\mathfrak{M}$ iff $|\mathfrak{M}|$ contains exactly n elements.

Proposition 4.13. *A structure is infinite iff it is a model of*

$$\{\varphi_{\geq 1}, \varphi_{\geq 2}, \varphi_{\geq 3}, \dots\}.$$

There is no single purely logical sentence which is true in $\mathfrak{M}$ iff $|\mathfrak{M}|$ is infinite. However, one can give sentences with non-logical predicate symbols which only have infinite models (although not every infinite structure is a model of them). The property of being a finite structure, and the property of being a non-enumerable structure cannot even be expressed with an infinite set of sentences. These facts follow from the compactness and Löwenheim-Skolem theorems.

Chapter 5

Derivation Systems

This chapter collects general material on derivation systems. A textbook using a specific system can insert the introduction section plus the relevant survey section at the beginning of the chapter introducing that system.

5.1 Introduction

Logics commonly have both a semantics and a derivation system. The semantics concerns concepts such as truth, satisfiability, validity, and entailment. The purpose of derivation systems is to provide a purely syntactic method of establishing entailment and validity. They are purely syntactic in the sense that a derivation in such a system is a finite syntactic object, usually a sequence (or other finite arrangement) of sentences or formulas. Good derivation systems have the property that any given sequence or arrangement of sentences or formulas can be verified mechanically to be "correct." fol:prf:int:sec

The simplest (and historically first) derivation systems for first-order logic were *axiomatic*. A sequence of formulas counts as a derivation in such a system if each individual formula in it is either among a fixed set of "axioms" or follows from formulas coming before it in the sequence by one of a fixed number of "inference rules"—and it can be mechanically verified if a formula is an axiom and whether it follows correctly from other formulas by one of the inference rules. Axiomatic derivation systems are easy to describe—and also easy to handle meta-theoretically—but derivations in them are hard to read and understand, and are also hard to produce.

Other derivation systems have been developed with the aim of making it easier to construct derivations or easier to understand derivations once they are complete. Examples are natural deduction, truth trees, also known as tableaux proofs, and the sequent calculus. Some derivation systems are designed especially with mechanization in mind, e.g., the resolution method is easy to implement in software (but its derivations are essentially impossible to understand). Most of these other derivation systems represent derivations as trees of formulas rather than sequences. This makes it easier to see which parts of a derivation depend on which other parts.

So for a given logic, such as first-order logic, the different derivation systems will give different explications of what it is for a sentence to be a *theorem* and what it means for a sentence to be derivable from some others. However that is done (via axiomatic derivations, natural deductions, sequent derivations, truth trees, resolution refutations), we want these relations to match the semantic notions of validity and entailment. Let's write $\vdash \varphi$ for "φ is a theorem" and "$\Gamma \vdash \varphi$" for "φ is derivable from Γ." However $\vdash$ is defined, we want it to match up with $\vDash$, that is:

1. $\vdash \varphi$ if and only if $\vDash \varphi$
2. $\Gamma \vdash \varphi$ if and only if $\Gamma \vDash \varphi$

The "only if" direction of the above is called *soundness*. A derivation system is sound if derivability guarantees entailment (or validity). Every decent derivation system has to be sound; unsound derivation systems are not useful at all. After all, the entire purpose of a derivation is to provide a syntactic guarantee of validity or entailment. We'll prove soundness for the derivation systems we present.

The converse "if" direction is also important: it is called *completeness*. A complete derivation system is strong enough to show that φ is a theorem whenever φ is valid, and that $\Gamma \vdash \varphi$ whenever $\Gamma \vDash \varphi$. Completeness is harder to establish, and some logics have no complete derivation systems. First-order logic does. Kurt Gödel was the first one to prove completeness for a derivation system of first-order logic in his 1929 dissertation.

Another concept that is connected to derivation systems is that of *consistency*. A set of sentences is called inconsistent if anything whatsoever can be derived from it, and consistent otherwise. Inconsistency is the syntactic counterpart to unsatisfiablity: like unsatisfiable sets, inconsistent sets of sentences do not make good theories, they are defective in a fundamental way. Consistent sets of sentences may not be true or useful, but at least they pass that minimal threshold of logical usefulness. For different derivation systems the specific definition of consistency of sets of sentences might differ, but like $\vdash$, we want consistency to coincide with its semantic counterpart, satisfiability. We want it to always be the case that Γ is consistent if and only if it is satisfiable. Here, the "if" direction amounts to completeness (consistency guarantees satisfiability), and the "only if" direction amounts to soundness (satisfiability guarantees consistency). In fact, for classical first-order logic, the two versions of soundness and completeness are equivalent.

5.2 The Sequent Calculus

fol:prf:seq:sec

While many derivation systems operate with arrangements of sentences, the sequent calculus operates with *sequents*. A sequent is an expression of the form

$$\varphi_1, \ldots, \varphi_m \Rightarrow \psi_1, \ldots, \psi_m,$$

that is a pair of sequences of sentences, separated by the sequent symbol $\Rightarrow$. Either sequence may be empty. A derivation in the sequent calculus is a tree of sequents, where the topmost sequents are of a special form (they are called "initial sequents" or "axioms") and every other sequent follows from the sequents immediately above it by one of the rules of inference. The rules of inference either manipulate the sentences in the sequents (adding, removing, or rearranging them on either the left or the right), or they introduce a complex formula in the conclusion of the rule. For instance, the $\land$L rule allows the inference from $\varphi, \Gamma \Rightarrow \Delta$ to $\varphi \land \psi, \Gamma \Rightarrow \Delta$, and the $\rightarrow$R allows the inference from $\varphi, \Gamma \Rightarrow \Delta, \psi$ to $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$, for any Γ, Δ, φ, and ψ. (In particular, Γ and Δ may be empty.)

The $\vdash$ relation based on the sequent calculus is defined as follows: $\Gamma \vdash \varphi$ iff there is some sequence Γ_0 such that every φ in Γ_0 is in Γ and there is a derivation with the sequent $\Gamma_0 \Rightarrow \varphi$ at its root. φ is a theorem in the sequent calculus if the sequent $\Rightarrow \varphi$ has a derivation. For instance, here is a derivation that shows that $\vdash (\varphi \land \psi) \rightarrow \varphi$:

$$\cfrac{\cfrac{\varphi \Rightarrow \varphi}{\varphi \land \psi \Rightarrow \varphi}\,\land\text{L}}{\Rightarrow (\varphi \land \psi) \rightarrow \varphi}\,\rightarrow\text{R}$$

A set Γ is inconsistent in the sequent calculus if there is a derivation of $\Gamma_0 \Rightarrow$ (where every $\varphi \in \Gamma_0$ is in Γ and the right side of the sequent is empty). Using the rule WR, any sentence can be derived from an inconsistent set.

The sequent calculus was invented in the 1930s by Gerhard Gentzen. Because of its systematic and symmetric design, it is a very useful formalism for developing a theory of derivations. It is relatively easy to find derivations in the sequent calculus, but these derivations are often hard to read and their connection to proofs are sometimes not easy to see. It has proved to be a very elegant approach to derivation systems, however, and many logics have sequent calculus systems.

5.3 Natural Deduction

Natural deduction is a derivation system intended to mirror actual reasoning (especially the kind of regimented reasoning employed by mathematicians). fol:prf:ntd:sec Actual reasoning proceeds by a number of "natural" patterns. For instance, proof by cases allows us to establish a conclusion on the basis of a disjunctive premise, by establishing that the conclusion follows from either of the disjuncts. Indirect proof allows us to establish a conclusion by showing that its negation leads to a contradiction. Conditional proof establishes a conditional claim "if ... then ..." by showing that the consequent follows from the antecedent.

Natural deduction is a formalization of some of these natural inferences. Each of the logical connectives and quantifiers comes with two rules, an introduction and an elimination rule, and they each correspond to one such natural inference pattern. For instance, $\rightarrow$Intro corresponds to conditional proof, and $\lor$Elim to proof by cases. A particularly simple rule is $\land$Elim which allows the inference from $\varphi \land \psi$ to φ (or ψ).

One feature that distinguishes natural deduction from other derivation systems is its use of assumptions. A derivation in natural deduction is a tree of formulas. A single formula stands at the root of the tree of formulas, and the "leaves" of the tree are formulas from which the conclusion is derived. In natural deduction, some leaf formulas play a role inside the derivation but are "used up" by the time the derivation reaches the conclusion. This corresponds to the practice, in actual reasoning, of introducing hypotheses which only remain in effect for a short while. For instance, in a proof by cases, we assume the truth of each of the disjuncts; in conditional proof, we assume the truth of the antecedent; in indirect proof, we assume the truth of the negation of the conclusion. This way of introducing hypothetical assumptions and then doing away with them in the service of establishing an intermediate step is a hallmark of natural deduction. The formulas at the leaves of a natural deduction derivation are called assumptions, and some of the rules of inference may "discharge" them. For instance, if we have a derivation of ψ from some assumptions which include φ, then the $\rightarrow$Intro rule allows us to infer $\varphi \rightarrow \psi$ and discharge any assumption of the form φ. (To keep track of which assumptions are discharged at which inferences, we label the inference and the assumptions it discharges with a number.) The assumptions that remain undischarged at the end of the derivation are together sufficient for the truth of the conclusion, and so a derivation establishes that its undischarged assumptions entail its conclusion.

The relation $\Gamma \vdash \varphi$ based on natural deduction holds iff there is a derivation in which φ is the last sentence in the tree, and every leaf which is undischarged is in Γ. φ is a theorem in natural deduction iff there is a derivation in which φ is the last sentence and all assumptions are discharged. For instance, here is a derivation that shows that $\vdash (\varphi \land \psi) \rightarrow \varphi$:

$$\cfrac{1\quad\cfrac{[\varphi \land \psi]^1}{\varphi}\,\land\text{Elim}}{(\varphi \land \psi) \rightarrow \varphi}\,\rightarrow\text{Intro}$$

The label 1 indicates that the assumption $\varphi \land \psi$ is discharged at the $\rightarrow$Intro inference.

A set Γ is inconsistent iff $\Gamma \vdash \bot$ in natural deduction. The rule $\bot_I$ makes it so that from an inconsistent set, any sentence can be derived.

Natural deduction systems were developed by Gerhard Gentzen and Stanisław Jaśkowski in the 1930s, and later developed by Dag Prawitz and Frederic Fitch. Because its inferences mirror natural methods of proof, it is favored by philosophers. The versions developed by Fitch are often used in introductory logic textbooks. In the philosophy of logic, the rules of natural deduction have sometimes been taken to give the meanings of the logical operators ("proof-theoretic semantics").

5.4 Tableaux

While many derivation systems operate with arrangements of sentences, tableaux operate with signed formulas. A signed formula is a pair consisting of a truth value sign ($\mathbb{T}$ or $\mathbb{F}$) and a sentence fol:prf:tab:sec

$$\mathbb{T}\,\varphi \text{ or } \mathbb{F}\,\varphi.$$

A tableau consists of signed formulas arranged in a downward-branching tree. It begins with a number of *assumptions* and continues with signed formulas which result from one of the signed formulas above it by applying one of the rules of inference. Each rule allows us to add one or more signed formulas to the end of a branch, or two signed formulas side by side—in this case a branch splits into two, with the two added signed formulas forming the ends of the two branches.

A rule applied to a complex signed formula results in the addition of signed formulas which are immediate sub-formulas. They come in pairs, one rule for each of the two signs. For instance, the $\wedge\mathbb{T}$ rule applies to $\mathbb{T}\,\varphi \wedge \psi$, and allows the addition of both the two signed formulas $\mathbb{T}\,\varphi$ and $\mathbb{T}\,\psi$ to the end of any branch containing $\mathbb{T}\,\varphi \wedge \psi$, and the rule $\varphi \wedge \psi\mathbb{F}$ allows a branch to be split by adding $\mathbb{F}\,\varphi$ and $\mathbb{F}\,\psi$ side-by-side. A tableau is closed if every one of its branches contains a matching pair of signed formulas $\mathbb{T}\,\varphi$ and $\mathbb{F}\,\varphi$.

The $\vdash$ relation based on tableaux is defined as follows: $\Gamma \vdash \varphi$ iff there is some finite set $\Gamma_0 = \{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ such that there is a closed tableau for the assumptions

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}$$

For instance, here is a closed tableau that shows that $\vdash (\varphi \wedge \psi) \rightarrow \varphi$:

1.	$\mathbb{F}\,(\varphi \wedge \psi) \rightarrow \varphi$	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$	$\rightarrow\mathbb{F}\,1$
3.	$\mathbb{F}\,\varphi$	$\rightarrow\mathbb{F}\,1$
4.	$\mathbb{T}\,\varphi$	$\rightarrow\mathbb{T}\,2$
5.	$\mathbb{T}\,\psi$	$\rightarrow\mathbb{T}\,2$
	$\otimes$	

A set Γ is inconsistent in the tableau calculus if there is a closed tableau for assumptions

$$\{\mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}$$

for some $\psi_i \in \Gamma$.

Tableaux were invented in the 1950s independently by Evert Beth and Jaakko Hintikka, and simplified and popularized by Raymond Smullyan. They are very easy to use, since constructing a tableau is a very systematic procedure. Because of the systematic nature of tableaux, they also lend themselves to implementation by computer. However, a tableau is often hard to read and their connection to proofs are sometimes not easy to see. The approach is also quite general, and many different logics have tableau systems. Tableaux also help us to find structures that satisfy given (sets of) sentences: if the set is satisfiable, it won't have a closed tableau, i.e., any tableau will have an open branch. The satisfying structure can be "read off" an open branch, provided every rule it is possible to apply has been applied on that branch. There is also a very close connection to the sequent calculus: essentially, a closed tableau is a condensed derivation in the sequent calculus, written upside-down.

5.5 Axiomatic Derivations

fol:prf:axd:sec Axiomatic derivations are the oldest and simplest logical derivation systems. Its derivations are simply sequences of sentences. A sequence of sentences counts as a correct derivation if every sentence φ in it satisfies one of the following conditions:

1. φ is an axiom, or
2. φ is an element of a given set Γ of sentences, or
3. φ is justified by a rule of inference.

To be an axiom, φ has to have the form of one of a number of fixed sentence schemas. There are many sets of axiom schemas that provide a satisfactory (sound and complete) derivation system for first-order logic. Some are organized according to the connectives they govern, e.g., the schemas

$$\varphi \to (\psi \to \varphi) \qquad \psi \to (\psi \lor \chi) \qquad (\psi \land \chi) \to \psi$$

are common axioms that govern $\to$, $\lor$ and $\land$. Some axiom systems aim at a minimal number of axioms. Depending on the connectives that are taken as primitives, it is even possible to find axiom systems that consist of a single axiom.

A rule of inference is a conditional statement that gives a sufficient condition for a sentence in a derivation to be justified. Modus ponens is one very common such rule: it says that if φ and $\varphi \to \psi$ are already justified, then ψ is justified. This means that a line in a derivation containing the sentence ψ is justified, provided that both φ and $\varphi \to \psi$ (for some sentence φ) appear in the derivation before ψ.

The $\vdash$ relation based on axiomatic derivations is defined as follows: $\Gamma \vdash \varphi$ iff there is a derivation with the sentence φ as its last formula (and Γ is taken as the set of sentences in that derivation which are justified by (2) above). φ is a theorem if φ has a derivation where Γ is empty, i.e., every sentence in the derivation is justfied either by (1) or (3). For instance, here is a derivation that shows that $\vdash \varphi \to (\psi \to (\psi \lor \varphi))$:

1. $\psi \to (\psi \lor \varphi)$
2. $(\psi \to (\psi \lor \varphi)) \to (\varphi \to (\psi \to (\psi \lor \varphi)))$
3. $\varphi \to (\psi \to (\psi \lor \varphi))$

The sentence on line 1 is of the form of the axiom $\varphi \to (\varphi \lor \psi)$ (with the roles of φ and ψ reversed). The sentence on line 2 is of the form of the axiom $\varphi \to (\psi \to \varphi)$. Thus, both lines are justified. Line 3 is justified by modus ponens: if we abbreviate it as θ, then line 2 has the form $\chi \to \theta$, where χ is $\psi \to (\psi \lor \varphi)$, i.e., line 1.

A set Γ is inconsistent if $\Gamma \vdash \bot$. A complete axiom system will also prove that $\bot \to \varphi$ for any φ, and so if Γ is inconsistent, then $\Gamma \vdash \varphi$ for any φ.

Systems of axiomatic derivations for logic were first given by Gottlob Frege in his 1879 *Begriffsschrift*, which for this reason is often considered the first work of modern logic. They were perfected in Alfred North Whitehead and Bertrand Russell's *Principia Mathematica* and by David Hilbert and his students in the 1920s. They are thus often called "Frege systems" or "Hilbert systems." They are very versatile in that it is often easy to find an axiomatic system for a logic. Because derivations have a very simple structure and only one or two inference rules, it is also relatively easy to prove things *about* them. However, they are very hard to use in practice, i.e., it is difficult to find and write proofs.

Chapter 6

The Sequent Calculus

> This chapter presents Gentzen's standard sequent calculus LK for classical first-order logic. It could use more examples and exercises. To include or exclude material relevant to the sequent calculus as a proof system, use the "prfLK" tag.

6.1 Rules and Derivations

fol:seq:rul:sec For the following, let $\Gamma, \Delta, \Pi, \Lambda$ represent finite sequences of sentences.

Definition 6.1 (Sequent). A *sequent* is an expression of the form

$$\Gamma \Rightarrow \Delta$$

where Γ and Δ are finite (possibly empty) sequences of sentences of the language $\mathcal{L}$. Γ is called the *antecedent*, while Δ is the *succedent*.

The intuitive idea behind a sequent is: if all of the sentences in the antecedent hold, then at least one of the sentences in the succedent holds. That is, if $\Gamma = \langle \varphi_1, \dots, \varphi_m \rangle$ and $\Delta = \langle \psi_1, \dots, \psi_n \rangle$, then $\Gamma \Rightarrow \Delta$ holds iff explanation

$$(\varphi_1 \wedge \dots \wedge \varphi_m) \rightarrow (\psi_1 \vee \dots \vee \psi_n)$$

holds. There are two special cases: where Γ is empty and when Δ is empty. When Γ is empty, i.e., $m = 0$, $\Rightarrow \Delta$ holds iff $\psi_1 \vee \dots \vee \psi_n$ holds. When Δ is empty, i.e., $n = 0$, $\Gamma \Rightarrow$ holds iff $\neg(\varphi_1 \wedge \dots \wedge \varphi_m)$ does. We say a sequent is valid iff the corresponding sentence is valid.

If Γ is a sequence of sentences, we write Γ, φ for the result of appending φ to the right end of Γ (and φ, Γ for the result of appending φ to the left end of Γ). If Δ is a sequence of sentences also, then Γ, Δ is the concatenation of the two sequences.

Definition 6.2 (Initial Sequent). An *initial sequent* is a sequent of one of the following forms:

1. $\varphi \Rightarrow \varphi$
2. $\Rightarrow \top$
3. $\bot \Rightarrow$

for any sentence φ in the language.

Derivations in the sequent calculus are certain trees of sequents, where the topmost sequents are initial sequents, and if a sequent stands below one or two other sequents, it must follow correctly by a rule of inference. The rules for **LK** are divided into two main types: *logical* rules and *structural* rules. The logical rules are named for the main operator of the sentence containing φ and/or ψ in the lower sequent. Each one comes in two versions, one for inferring a sequent with the sentence containing the logical operator on the left, and one with the sentence on the right.

6.2 Propositional Rules

Rules for ¬ fol:seq:prl:sec

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

Rules for ∧

$$\frac{\varphi, \Gamma \Rightarrow \Delta}{\varphi \land \psi, \Gamma \Rightarrow \Delta}\,\land\mathrm{L}$$

$$\frac{\psi, \Gamma \Rightarrow \Delta}{\varphi \land \psi, \Gamma \Rightarrow \Delta}\,\land\mathrm{L}$$

$$\frac{\Gamma \Rightarrow \Delta, \varphi \qquad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \land \psi}\,\land\mathrm{R}$$

Rules for ∨

$$\frac{\varphi, \Gamma \Rightarrow \Delta \qquad \psi, \Gamma \Rightarrow \Delta}{\varphi \lor \psi, \Gamma \Rightarrow \Delta}\,\lor\mathrm{L}$$

$$\frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \varphi \lor \psi}\,\lor\mathrm{R}$$

$$\frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \lor \psi}\,\lor\mathrm{R}$$

Rules for $\to$

$$\frac{\Gamma \Rightarrow \Delta, \varphi \qquad \psi, \Pi \Rightarrow \Lambda}{\varphi \to \psi, \Gamma, \Pi \Rightarrow \Delta, \Lambda} \to\text{L} \qquad \frac{\varphi, \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \to \psi} \to\text{R}$$

6.3 Quantifier Rules

fol:seq:qrl:sec

Rules for $\forall$

$$\frac{\varphi(t), \Gamma \Rightarrow \Delta}{\forall x\, \varphi(x), \Gamma \Rightarrow \Delta} \forall\text{L} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi(a)}{\Gamma \Rightarrow \Delta, \forall x\, \varphi(x)} \forall\text{R}$$

In $\forall$L, t is a closed term (i.e., one without variables). In $\forall$R, a is a constant symbol which must not occur anywhere in the lower sequent of the $\forall$R rule. We call a the *eigenvariable* of the $\forall$R inference.[1]

Rules for $\exists$

$$\frac{\varphi(a), \Gamma \Rightarrow \Delta}{\exists x\, \varphi(x), \Gamma \Rightarrow \Delta} \exists\text{L} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi(t)}{\Gamma \Rightarrow \Delta, \exists x\, \varphi(x)} \exists\text{R}$$

Again, t is a closed term, and a is a constant symbol which does not occur in the lower sequent of the $\exists$L rule. We call a the *eigenvariable* of the $\exists$L inference.

The condition that an eigenvariable not occur in the lower sequent of the $\forall$R or $\exists$L inference is called the *eigenvariable condition*.

Recall the convention that when φ is a formula with the variable x free, we indicate this by writing $\varphi(x)$. In the same context, $\varphi(t)$ then is short for $\varphi[t/x]$. So we could also write the $\exists$R rule as: explanation

$$\frac{\Gamma \Rightarrow \Delta, \varphi[t/x]}{\Gamma \Rightarrow \Delta, \exists x\, \varphi} \exists\text{R}$$

Note that t may already occur in φ, e.g., φ might be $P(t, x)$. Thus, inferring $\Gamma \Rightarrow \Delta, \exists x\, P(t, x)$ from $\Gamma \Rightarrow \Delta, P(t, t)$ is a correct application of $\exists$R—you may "replace" one or more, and not necessarily all, occurrences of t in the premise by the bound variable x. However, the eigenvariable conditions in $\forall$R

[1] We use the term "eigenvariable" even though a in the above rule is a constant symbol. This has historical reasons.

and $\exists$L require that the constant symbol a does not occur in φ. So, you cannot correctly infer $\Gamma \Rightarrow \Delta, \forall x\, P(a, x)$ from $\Gamma \Rightarrow \Delta, P(a, a)$ using $\forall$R.

explanation

In $\exists$R and $\forall$L there are no restrictions on the term t. On the other hand, in the $\exists$L and $\forall$R rules, the eigenvariable condition requires that the constant symbol a does not occur anywhere outside of $\varphi(a)$ in the upper sequent. It is necessary to ensure that the system is sound, i.e., only derives sequents that are valid. Without this condition, the following would be allowed:

$$\dfrac{\dfrac{\varphi(a) \Rightarrow \varphi(a)}{\exists x\, \varphi(x) \Rightarrow \varphi(a)}\, {*}\exists\text{L}}{\exists x\, \varphi(x) \Rightarrow \forall x\, \varphi(x)}\, \forall\text{R} \qquad \dfrac{\dfrac{\varphi(a) \Rightarrow \varphi(a)}{\varphi(a) \Rightarrow \forall x\, \varphi(x)}\, {*}\forall\text{R}}{\exists x\, \varphi(x) \Rightarrow \forall x\, \varphi(x)}\, \exists\text{L}$$

However, $\exists x\, \varphi(x) \Rightarrow \forall x\, \varphi(x)$ is not valid.

6.4 Structural Rules

We also need a few rules that allow us to rearrange sentences in the left and right side of a sequent. Since the logical rules require that the sentences in the premise which the rule acts upon stand either to the far left or to the far right, we need an "exchange" rule that allows us to move sentences to the right position. It's also important sometimes to be able to combine two identical sentences into one, and to add a sentence on either side. fol:seq:srl:sec

Weakening

$$\dfrac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\, \text{WL} \qquad \dfrac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \varphi}\, \text{WR}$$

Contraction

$$\dfrac{\varphi, \varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\, \text{CL} \qquad \dfrac{\Gamma \Rightarrow \Delta, \varphi, \varphi}{\Gamma \Rightarrow \Delta, \varphi}\, \text{CR}$$

Exchange

$$\dfrac{\Gamma, \varphi, \psi, \Pi \Rightarrow \Delta}{\Gamma, \psi, \varphi, \Pi \Rightarrow \Delta}\, \text{XL} \qquad \dfrac{\Gamma \Rightarrow \Delta, \varphi, \psi, \Lambda}{\Gamma \Rightarrow \Delta, \psi, \varphi, \Lambda}\, \text{XR}$$

A series of weakening, contraction, and exchange inferences will often be indicated by double inference lines.

The following rule, called "cut," is not strictly speaking necessary, but makes it a lot easier to reuse and combine derivations.

$$\frac{\Gamma \Rightarrow \Delta, \varphi \qquad \varphi, \Pi \Rightarrow \Lambda}{\Gamma, \Pi \Rightarrow \Delta, \Lambda}\ \text{Cut}$$

6.5 Derivations

fol:seq:der:sec We've said what an initial sequent looks like, and we've given the rules of inference. Derivations in the sequent calculus are inductively generated from these: each derivation either is an initial sequent on its own, or consists of one or two derivations followed by an inference. explanation

Definition 6.3 (LK derivation). An **LK**-*derivation* of a sequent S is a finite tree of sequents satisfying the following conditions:

1. The topmost sequents of the tree are initial sequents.
2. The bottommost sequent of the tree is S.
3. Every sequent in the tree except S is a premise of a correct application of an inference rule whose conclusion stands directly below that sequent in the tree.

We then say that S is the *end-sequent* of the derivation and that S is *derivable in* **LK** (or **LK**-derivable).

Example 6.4. Every initial sequent, e.g., $\chi \Rightarrow \chi$ is a derivation. We can obtain a new derivation from this by applying, say, the WL rule,

$$\frac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\ \text{WL}$$

The rule, however, is meant to be general: we can replace the φ in the rule with any sentence, e.g., also with θ. If the premise matches our initial sequent $\chi \Rightarrow \chi$, that means that both Γ and Δ are just χ, and the conclusion would then be $\theta, \chi \Rightarrow \chi$. So, the following is a derivation:

$$\frac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi}\ \text{WL}$$

We can now apply another rule, say XL, which allows us to switch two sentences on the left. So, the following is also a correct derivation:

$$\cfrac{\cfrac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi}\ \text{WL}}{\chi, \theta \Rightarrow \chi}\ \text{XL}$$

In this application of the rule, which was given as

$$\frac{\Gamma, \varphi, \psi, \Pi \Rightarrow \Delta}{\Gamma, \psi, \varphi, \Pi \Rightarrow \Delta,}\ \text{XL}$$

both Γ and Π were empty, Δ is χ, and the roles of φ and ψ are played by θ and χ, respectively. In much the same way, we also see that

$$\frac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta}\ \text{WL}$$

is a derivation. Now we can take these two derivations, and combine them using $\wedge$R. That rule was

$$\frac{\Gamma \Rightarrow \Delta, \varphi \qquad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}\ \wedge\text{R}$$

In our case, the premises must match the last sequents of the derivations ending in the premises. That means that Γ is χ, θ, Δ is empty, φ is χ and ψ is θ. So the conclusion, if the inference should be correct, is $\chi, \theta \Rightarrow \chi \wedge \theta$.

$$\frac{\dfrac{\dfrac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi}\ \text{WL}}{\chi, \theta \Rightarrow \chi}\ \text{XL} \qquad \dfrac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta}\ \text{WL}}{\chi, \theta \Rightarrow \chi \wedge \theta}\ \wedge\text{R}$$

Of course, we can also reverse the premises, then φ would be θ and ψ would be χ.

$$\frac{\dfrac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta}\ \text{WL} \qquad \dfrac{\dfrac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi}\ \text{WL}}{\chi, \theta \Rightarrow \chi}\ \text{XL}}{\chi, \theta \Rightarrow \theta \wedge \chi}\ \wedge\text{R}$$

6.6 Examples of Derivations

fol:seq:pro:sec

Example 6.5. Give an **LK**-derivation for the sequent $\varphi \wedge \psi \Rightarrow \varphi$.

We begin by writing the desired end-sequent at the bottom of the derivation.

$$\frac{}{\varphi \wedge \psi \Rightarrow \varphi}$$

Next, we need to figure out what kind of inference could have a lower sequent of this form. This could be a structural rule, but it is a good idea to start by looking for a logical rule. The only logical connective occurring in the lower sequent is $\wedge$, so we're looking for an $\wedge$ rule, and since the $\wedge$ symbol occurs in the antecedent, we're looking at the $\wedge$L rule.

$$\frac{}{\varphi \wedge \psi \Rightarrow \varphi}\ \wedge\text{L}$$

There are two options for what could have been the upper sequent of the $\wedge$L inference: we could have an upper sequent of $\varphi \Rightarrow \varphi$, or of $\psi \Rightarrow \varphi$. Clearly, $\varphi \Rightarrow \varphi$ is an initial sequent (which is a good thing), while $\psi \Rightarrow \varphi$ is not derivable in general. We fill in the upper sequent:

$$\dfrac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi}\ \wedge\text{L}$$

We now have a correct **LK**-derivation of the sequent $\varphi \wedge \psi \Rightarrow \varphi$.

Example 6.6. Give an **LK**-derivation for the sequent $\neg\varphi \vee \psi \Rightarrow \varphi \to \psi$.

Begin by writing the desired end-sequent at the bottom of the derivation.

$$\dfrac{}{\neg\varphi \vee \psi \Rightarrow \varphi \to \psi}$$

To find a logical rule that could give us this end-sequent, we look at the logical connectives in the end-sequent: $\neg$, $\vee$, and $\to$. We only care at the moment about $\vee$ and $\to$ because they are main operators of sentences in the end-sequent, while $\neg$ is inside the scope of another connective, so we will take care of it later. Our options for logical rules for the final inference are therefore the $\vee$L rule and the $\to$R rule. We could pick either rule, really, but let's pick the $\to$R rule (if for no reason other than it allows us to put off splitting into two branches). According to the form of $\to$R inferences which can yield the lower sequent, this must look like:

$$\dfrac{\dfrac{}{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}}{\neg\varphi \vee \psi \Rightarrow \varphi \to \psi}\ \to\text{R}$$

If we move $\neg\varphi \vee \psi$ to the outside of the antecedent, we can apply the $\vee$L rule. According to the schema, this must split into two upper sequents as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\varphi, \varphi \Rightarrow \psi} \qquad \dfrac{}{\psi, \varphi \Rightarrow \psi}}{\neg\varphi \vee \psi, \varphi \Rightarrow \psi}\ \vee\text{L}}{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}\ \text{XR}}{\neg\varphi \vee \psi \Rightarrow \varphi \to \psi}\ \to\text{R}$$

Remember that we are trying to wind our way up to initial sequents; we seem to be pretty close! The right branch is just one weakening and one exchange away from an initial sequent and then it is done:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\varphi, \varphi \Rightarrow \psi} \qquad \dfrac{\dfrac{\psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \psi}\ \text{WL}}{\psi, \varphi \Rightarrow \psi}\ \text{XL}}{\neg\varphi \vee \psi, \varphi \Rightarrow \psi}\ \vee\text{L}}{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}\ \text{XR}}{\neg\varphi \vee \psi \Rightarrow \varphi \to \psi}\ \to\text{R}$$

Now looking at the left branch, the only logical connective in any sentence is the $\neg$ symbol in the antecedent sentences, so we're looking at an instance of the $\neg$L rule.

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$\varphi \Rightarrow \psi, \varphi$}
\RightLabel{$\neg$L}
\UnaryInfC{$\neg\varphi, \varphi \Rightarrow \psi$}
\AxiomC{$\psi \Rightarrow \psi$}
\RightLabel{WL}
\UnaryInfC{$\varphi, \psi \Rightarrow \psi$}
\RightLabel{XL}
\UnaryInfC{$\psi, \varphi \Rightarrow \psi$}
\RightLabel{$\lor$L}
\BinaryInfC{$\neg\varphi \lor \psi, \varphi \Rightarrow \psi$}
\RightLabel{XR}
\UnaryInfC{$\varphi, \neg\varphi \lor \psi \Rightarrow \psi$}
\RightLabel{$\to$R}
\UnaryInfC{$\neg\varphi \lor \psi \Rightarrow \varphi \to \psi$}
\end{prooftree}
$$

Similarly to how we finished off the right branch, we are just one weakening and one exchange away from finishing off this left branch as well.

$$
\begin{prooftree}
\AxiomC{$\varphi \Rightarrow \varphi$}
\RightLabel{WR}
\UnaryInfC{$\varphi \Rightarrow \varphi, \psi$}
\RightLabel{XR}
\UnaryInfC{$\varphi \Rightarrow \psi, \varphi$}
\RightLabel{$\neg$L}
\UnaryInfC{$\neg\varphi, \varphi \Rightarrow \psi$}
\AxiomC{$\psi \Rightarrow \psi$}
\RightLabel{WL}
\UnaryInfC{$\varphi, \psi \Rightarrow \psi$}
\RightLabel{XL}
\UnaryInfC{$\psi, \varphi \Rightarrow \psi$}
\RightLabel{$\lor$L}
\BinaryInfC{$\neg\varphi \lor \psi, \varphi \Rightarrow \psi$}
\RightLabel{XR}
\UnaryInfC{$\varphi, \neg\varphi \lor \psi \Rightarrow \psi$}
\RightLabel{$\to$R}
\UnaryInfC{$\neg\varphi \lor \psi \Rightarrow \varphi \to \psi$}
\end{prooftree}
$$

Example 6.7. Give an **LK**-derivation of the sequent $\neg\varphi \lor \neg\psi \Rightarrow \neg(\varphi \land \psi)$

Using the techniques from above, we start by writing the desired end-sequent at the bottom.

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$\neg\varphi \lor \neg\psi \Rightarrow \neg(\varphi \land \psi)$}
\end{prooftree}
$$

The available main connectives of sentences in the end-sequent are the $\lor$ symbol and the $\neg$ symbol. It would work to apply either the $\lor$L or the $\neg$R rule here, but we start with the $\neg$R rule because it avoids splitting up into two branches for a moment:

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$\varphi \land \psi, \neg\varphi \lor \neg\psi \Rightarrow$}
\RightLabel{$\neg$R}
\UnaryInfC{$\neg\varphi \lor \neg\psi \Rightarrow \neg(\varphi \land \psi)$}
\end{prooftree}
$$

Now we have a choice of whether to look at the $\land$L or the $\lor$L rule. Let's see what happens when we apply the $\land$L rule: we have a choice to start with either the sequent $\varphi, \neg\varphi \lor \psi \Rightarrow$ or the sequent $\psi, \neg\varphi \lor \psi \Rightarrow$. Since the derivation is symmetric with regards to φ and ψ, let's go with the former:

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$\varphi, \neg\varphi \lor \neg\psi \Rightarrow$}
\RightLabel{$\land$L}
\UnaryInfC{$\varphi \land \psi, \neg\varphi \lor \neg\psi \Rightarrow$}
\RightLabel{$\neg$R}
\UnaryInfC{$\neg\varphi \lor \neg\psi \Rightarrow \neg(\varphi \land \psi)$}
\end{prooftree}
$$

Continuing to fill in the derivation, we see that we run into a problem:

$$
\begin{prooftree}
\AxiomC{$\varphi \Rightarrow \varphi$}
\RightLabel{$\neg$L}
\UnaryInfC{$\neg\varphi, \varphi \Rightarrow$}
\AxiomC{}
\RightLabel{?}
\UnaryInfC{$\varphi \Rightarrow \psi$}
\RightLabel{$\neg$L}
\UnaryInfC{$\neg\psi, \varphi \Rightarrow$}
\RightLabel{$\lor$L}
\BinaryInfC{$\neg\varphi \lor \neg\psi, \varphi \Rightarrow$}
\RightLabel{XL}
\UnaryInfC{$\varphi, \neg\varphi \lor \neg\psi \Rightarrow$}
\RightLabel{$\land$L}
\UnaryInfC{$\varphi \land \psi, \neg\varphi \lor \neg\psi \Rightarrow$}
\RightLabel{$\neg$R}
\UnaryInfC{$\neg\varphi \lor \neg\psi \Rightarrow \neg(\varphi \land \psi)$}
\end{prooftree}
$$

The top of the right branch cannot be reduced any further, and it cannot be brought by way of structural inferences to an initial sequent, so this is not the right path to take. So clearly, it was a mistake to apply the $\land$L rule above. Going back to what we had before and carrying out the $\lor$L rule instead, we get

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$\neg\varphi, \varphi \land \psi \Rightarrow$}
\AxiomC{}
\UnaryInfC{$\neg\psi, \varphi \land \psi \Rightarrow$}
\RightLabel{$\lor$L}
\BinaryInfC{$\neg\varphi \lor \neg\psi, \varphi \land \psi \Rightarrow$}
\RightLabel{XL}
\UnaryInfC{$\varphi \land \psi, \neg\varphi \lor \neg\psi \Rightarrow$}
\RightLabel{$\neg$R}
\UnaryInfC{$\neg\varphi \lor \neg\psi \Rightarrow \neg(\varphi \land \psi)$}
\end{prooftree}
$$

Completing each branch as we've done before, we get

$$
\begin{prooftree}
\AxiomC{$\varphi \Rightarrow \varphi$}
\RightLabel{$\land$L}
\UnaryInfC{$\varphi \land \psi \Rightarrow \varphi$}
\RightLabel{$\neg$L}
\UnaryInfC{$\neg\varphi, \varphi \land \psi \Rightarrow$}
\AxiomC{$\psi \Rightarrow \psi$}
\RightLabel{$\land$L}
\UnaryInfC{$\varphi \land \psi \Rightarrow \psi$}
\RightLabel{$\neg$L}
\UnaryInfC{$\neg\psi, \varphi \land \psi \Rightarrow$}
\RightLabel{$\lor$L}
\BinaryInfC{$\neg\varphi \lor \neg\psi, \varphi \land \psi \Rightarrow$}
\RightLabel{XL}
\UnaryInfC{$\varphi \land \psi, \neg\varphi \lor \neg\psi \Rightarrow$}
\RightLabel{$\neg$R}
\UnaryInfC{$\neg\varphi \lor \neg\psi \Rightarrow \neg(\varphi \land \psi)$}
\end{prooftree}
$$

(We could have carried out the $\land$ rules lower than the $\neg$ rules in these steps and still obtained a correct derivation).

Example 6.8. So far we haven't used the contraction rule, but it is sometimes required. Here's an example where that happens. Suppose we want to prove $\Rightarrow \varphi \lor \neg\varphi$. Applying $\lor$R backwards would give us one of these two derivations:

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$\Rightarrow \varphi$}
\RightLabel{$\lor$R}
\UnaryInfC{$\Rightarrow \varphi \lor \neg\varphi$}
\end{prooftree}
\qquad
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$\varphi \Rightarrow$}
\RightLabel{$\neg$R}
\UnaryInfC{$\Rightarrow \neg\varphi$}
\RightLabel{$\lor$R}
\UnaryInfC{$\Rightarrow \varphi \lor \neg\varphi$}
\end{prooftree}
$$

Neither of these of course ends in an initial sequent. The trick is to realize that the contraction rule allows us to combine two copies of a sentence into one—and when we're searching for a proof, i.e., going from bottom to top, we can keep a copy of $\varphi \lor \neg\varphi$ in the premise, e.g.,

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$\Rightarrow \varphi \lor \neg\varphi, \varphi$}
\RightLabel{$\lor$R}
\UnaryInfC{$\Rightarrow \varphi \lor \neg\varphi, \varphi \lor \neg\varphi$}
\RightLabel{CR}
\UnaryInfC{$\Rightarrow \varphi \lor \neg\varphi$}
\end{prooftree}
$$

Now we can apply $\lor$R a second time, and also get $\lnot\varphi$, which leads to a complete derivation.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \lnot\varphi}\;\lnot\text{R}}{\Rightarrow \varphi, \varphi \lor \lnot\varphi}\;\lor\text{R}}{\Rightarrow \varphi \lor \lnot\varphi, \varphi}\;\text{XR}}{\Rightarrow \varphi \lor \lnot\varphi, \varphi \lor \lnot\varphi}\;\lor\text{R}}{\Rightarrow \varphi \lor \lnot\varphi}\;\text{CR}$$

Problem 6.1. Give derivations of the following sequents:

1. $\varphi \land (\psi \land \chi) \Rightarrow (\varphi \land \psi) \land \chi$.
2. $\varphi \lor (\psi \lor \chi) \Rightarrow (\varphi \lor \psi) \lor \chi$.
3. $\varphi \to (\psi \to \chi) \Rightarrow \psi \to (\varphi \to \chi)$.
4. $\varphi \Rightarrow \lnot\lnot\varphi$.

Problem 6.2. Give derivations of the following sequents:

1. $(\varphi \lor \psi) \to \chi \Rightarrow \varphi \to \chi$.
2. $(\varphi \to \chi) \land (\psi \to \chi) \Rightarrow (\varphi \lor \psi) \to \chi$.
3. $\Rightarrow \lnot(\varphi \land \lnot\varphi)$.
4. $\psi \to \varphi \Rightarrow \lnot\varphi \to \lnot\psi$.
5. $\Rightarrow (\varphi \to \lnot\varphi) \to \lnot\varphi$.
6. $\Rightarrow \lnot(\varphi \to \psi) \to \lnot\psi$.
7. $\varphi \to \chi \Rightarrow \lnot(\varphi \land \lnot\chi)$.
8. $\varphi \land \lnot\chi \Rightarrow \lnot(\varphi \to \chi)$.
9. $\varphi \lor \psi, \lnot\psi \Rightarrow \varphi$.
10. $\lnot\varphi \lor \lnot\psi \Rightarrow \lnot(\varphi \land \psi)$.
11. $\Rightarrow (\lnot\varphi \land \lnot\psi) \to \lnot(\varphi \lor \psi)$.
12. $\Rightarrow \lnot(\varphi \lor \psi) \to (\lnot\varphi \land \lnot\psi)$.

Problem 6.3. Give derivations of the following sequents:

1. $\lnot(\varphi \to \psi) \Rightarrow \varphi$.
2. $\lnot(\varphi \land \psi) \Rightarrow \lnot\varphi \lor \lnot\psi$.
3. $\varphi \to \psi \Rightarrow \lnot\varphi \lor \psi$.

4. $\Rightarrow \neg\neg\varphi \rightarrow \varphi$.
5. $\varphi \rightarrow \psi, \neg\varphi \rightarrow \psi \Rightarrow \psi$.
6. $(\varphi \wedge \psi) \rightarrow \chi \Rightarrow (\varphi \rightarrow \chi) \vee (\psi \rightarrow \chi)$.
7. $(\varphi \rightarrow \psi) \rightarrow \varphi \Rightarrow \varphi$.
8. $\Rightarrow (\varphi \rightarrow \psi) \vee (\psi \rightarrow \chi)$.

(These all require the CR rule.)

6.7 Derivations with Quantifiers

fol:seq:prq:sec

Example 6.9. Give an **LK**-derivation of the sequent $\exists x\, \neg\varphi(x) \Rightarrow \neg\forall x\, \varphi(x)$.

When dealing with quantifiers, we have to make sure not to violate the eigenvariable condition, and sometimes this requires us to play around with the order of carrying out certain inferences. In general, it helps to try and take care of rules subject to the eigenvariable condition first (they will be lower down in the finished proof). Also, it is a good idea to try and look ahead and try to guess what the initial sequent might look like. In our case, it will have to be something like $\varphi(a) \Rightarrow \varphi(a)$. That means that when we are "reversing" the quantifier rules, we will have to pick the same term—what we will call a—for both the $\forall$ and the $\exists$ rule. If we picked different terms for each rule, we would end up with something like $\varphi(a) \Rightarrow \varphi(b)$, which, of course, is not derivable.

Starting as usual, we write

$$\overline{\exists x\, \neg\varphi(x) \;\Rightarrow\; \neg\forall x\, \varphi(x)}$$

We could either carry out the $\exists$L rule or the $\neg$R rule. Since the $\exists$L rule is subject to the eigenvariable condition, it's a good idea to take care of it sooner rather than later, so we'll do that one first.

$$\dfrac{\overline{\neg\varphi(a) \;\Rightarrow\; \neg\forall x\, \varphi(x)}}{\exists x\, \neg\varphi(x) \;\Rightarrow\; \neg\forall x\, \varphi(x)}\;\exists\mathrm{L}$$

Applying the $\neg$L and $\neg$R rules backwards, we get

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{\forall x\, \varphi(x) \;\Rightarrow\; \varphi(a)}}{\neg\varphi(a), \forall x\, \varphi(x) \;\Rightarrow\;}\;\neg\mathrm{L}}{\forall x\, \varphi(x), \neg\varphi(a) \;\Rightarrow\;}\;\mathrm{XL}}{\neg\varphi(a) \;\Rightarrow\; \neg\forall x \varphi(x)}\;\neg\mathrm{R}}{\exists x \neg\varphi(x) \;\Rightarrow\; \neg\forall x \varphi(x)}\;\exists\mathrm{L}$$

At this point, our only option is to carry out the $\forall$L rule. Since this rule is not subject to the eigenvariable restriction, we're in the clear. Remember, we want to try and obtain an initial sequent (of the form $\varphi(a) \Rightarrow \varphi(a)$), so we should choose a as our argument for φ when we apply the rule.

$$
\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\varphi(a) \Rightarrow \varphi(a)}{\forall x\, \varphi(x) \Rightarrow \varphi(a)}\;\forall\text{L}}{\neg\varphi(a), \forall x\, \varphi(x) \Rightarrow}\;\neg\text{L}}{\forall x\, \varphi(x), \neg\varphi(a) \Rightarrow}\;\text{XL}}{\neg\varphi(a) \Rightarrow \neg\forall x\, \varphi(x)}\;\neg\text{R}}{\exists x\, \neg\varphi(x) \Rightarrow \neg\forall x\, \varphi(x)}\;\exists\text{L}
$$

It is important, especially when dealing with quantifiers, to double check at this point that the eigenvariable condition has not been violated. Since the only rule we applied that is subject to the eigenvariable condition was $\exists$L, and the eigenvariable a does not occur in its lower sequent (the end-sequent), this is a correct derivation.

Problem 6.4. Give derivations of the following sequents:

1. $\Rightarrow (\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \forall z\, (\varphi(z) \land \psi(z))$.
2. $\Rightarrow (\exists x\, \varphi(x) \lor \exists y\, \psi(y)) \to \exists z\, (\varphi(z) \lor \psi(z))$.
3. $\forall x\, (\varphi(x) \to \psi) \Rightarrow \exists y\, \varphi(y) \to \psi$.
4. $\forall x\, \neg\varphi(x) \Rightarrow \neg\exists x\, \varphi(x)$.
5. $\Rightarrow \neg\exists x\, \varphi(x) \to \forall x\, \neg\varphi(x)$.
6. $\Rightarrow \neg\exists x\, \forall y\, ((\varphi(x, y) \to \neg\varphi(y, y)) \land (\neg\varphi(y, y) \to \varphi(x, y)))$.

Problem 6.5. Give derivations of the following sequents:

1. $\Rightarrow \neg\forall x\, \varphi(x) \to \exists x\, \neg\varphi(x)$.
2. $(\forall x\, \varphi(x) \to \psi) \Rightarrow \exists y\, (\varphi(y) \to \psi)$.
3. $\Rightarrow \exists x\, (\varphi(x) \to \forall y\, \varphi(y))$.

(These all require the CR rule.)

This section collects the definitions of the provability relation and consistency for natural deduction.

6.8 Proof-Theoretic Notions

explanation Just as we've defined a number of important semantic notions (validity, entailment, satisfiabilty), we now define corresponding *proof-theoretic notions.* These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the derivability or non-derivability of certain sequents. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorem.* fol:seq:ptn:sec

Definition 6.10 (Theorems). A sentence φ is a *theorem* if there is a derivation in **LK** of the sequent $\Rightarrow \varphi$. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definition 6.11 (Derivability). A sentence φ is *derivable from* a set of sentences Γ, $\Gamma \vdash \varphi$, iff there is a finite subset $\Gamma_0 \subseteq \Gamma$ and a sequence Γ_0' of the sentences in Γ_0 such that **LK** derives $\Gamma_0' \Rightarrow \varphi$. If φ is not derivable from Γ we write $\Gamma \nvdash \varphi$.

Because of the contraction, weakening, and exchange rules, the order and number of sentences in Γ_0' does not matter: if a sequent $\Gamma_0' \Rightarrow \varphi$ is derivable, then so is $\Gamma_0'' \Rightarrow \varphi$ for any Γ_0'' that contains the same sentences as Γ_0'. For instance, if $\Gamma_0 = \{\psi, \chi\}$ then both $\Gamma_0' = \langle \psi, \psi, \chi \rangle$ and $\Gamma_0'' = \langle \chi, \chi, \psi \rangle$ are sequences containing just the sentences in Γ_0. If a sequent containing one is derivable, so is the other, e.g.:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \psi, \psi, \chi \Rightarrow \varphi\end{array}}{\psi, \chi \Rightarrow \varphi}\,\mathrm{CL}}{\chi, \psi \Rightarrow \varphi}\,\mathrm{XL}}{\chi, \chi, \psi \Rightarrow \varphi}\,\mathrm{WL}$$

From now on we'll say that if Γ_0 is a finite set of sentences then $\Gamma_0 \Rightarrow \varphi$ is any sequent where the antecedent is a sequence of sentences in Γ_0 and tacitly include contractions, exchanges, and weakenings if necessary.

Definition 6.12 (Consistency). A set of sentences Γ is *inconsistent* iff there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that **LK** derives $\Gamma_0 \Rightarrow$. If Γ is not inconsistent, i.e., if for every finite $\Gamma_0 \subseteq \Gamma$, **LK** does not derive $\Gamma_0 \Rightarrow$, we say it is *consistent*.

fol:seq:ptn: prop:reflexivity **Proposition 6.13 (Reflexivity).** *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.*

Proof. The initial sequent $\varphi \Rightarrow \varphi$ is derivable, and $\{\varphi\} \subseteq \Gamma$. □

fol:seq:ptn: prop:monotonicity **Proposition 6.14 (Monotonicity).** *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.*

Proof. Suppose $\Gamma \vdash \varphi$, i.e., there is a finite $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \Rightarrow \varphi$ is derivable. Since $\Gamma \subseteq \Delta$, then Γ_0 is also a finite subset of Δ. The derivation of $\Gamma_0 \Rightarrow \varphi$ thus also shows $\Delta \vdash \varphi$. □

fol:seq:ptn: prop:transitivity **Proposition 6.15 (Transitivity).** *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.*

Proof. If $\Gamma \vdash \varphi$, there is a finite $\Gamma_0 \subseteq \Gamma$ and a derivation π_0 of $\Gamma_0 \Rightarrow \varphi$. If $\{\varphi\} \cup \Delta \vdash \psi$, then for some finite subset $\Delta_0 \subseteq \Delta$, there is a derivation π_1 of $\varphi, \Delta_0 \Rightarrow \psi$. Consider the following derivation:

$$\cfrac{\begin{array}{cc} \vdots\,\pi_0 & \vdots\,\pi_1 \\ \Gamma_0 \Rightarrow \varphi & \varphi, \Delta_0 \Rightarrow \psi \end{array}}{\Gamma_0, \Delta_0 \Rightarrow \psi}\ \text{Cut}$$

Since $\Gamma_0 \cup \Delta_0 \subseteq \Gamma \cup \Delta$, this shows $\Gamma \cup \Delta \vdash \psi$. □

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \dots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i, then $\Gamma \vdash \psi$.

Proposition 6.16. *Γ is inconsistent iff $\Gamma \vdash \varphi$ for every sentence φ.* *fol:seq:ptn: prop:incons*

Proof. Exercise. □

Problem 6.6. Prove Proposition 6.16

Proposition 6.17 (Compactness). *fol:seq:ptn: prop:proves-compact*

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Proof.

1. If $\Gamma \vdash \varphi$, then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that the sequent $\Gamma_0 \Rightarrow \varphi$ has a derivation. Consequently, $\Gamma_0 \vdash \varphi$.
2. If Γ is inconsistent, there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that **LK** derives $\Gamma_0 \Rightarrow$. But then Γ_0 is a finite subset of Γ that is inconsistent. □

6.9 Derivability and Consistency

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem. fol:seq:prv: sec

Proposition 6.18. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.* *fol:seq:prv: prop:provability-contr*

Proof. There are finite Γ_0 and $\Gamma_1 \subseteq \Gamma$ such that **LK** derives $\Gamma_0 \Rightarrow \varphi$ and $\varphi, \Gamma_1 \Rightarrow$. Let the **LK**-derivation of $\Gamma_0 \Rightarrow \varphi$ be π_0 and the **LK**-derivation of $\Gamma_1, \varphi \Rightarrow$ be π_1. We can then derive

$$\cfrac{\begin{array}{cc} \vdots\,\pi_0 & \vdots\,\pi_1 \\ \Gamma_0 \Rightarrow \varphi & \varphi, \Gamma_1 \Rightarrow \end{array}}{\Gamma_0, \Gamma_1 \Rightarrow}\ \text{Cut}$$

Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$, hence Γ is inconsistent. □

fol:seq:prv: prop:prov-incons **Proposition 6.19.** *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.*

Proof. First suppose $\Gamma \vdash \varphi$, i.e., there is a derivation π_0 of $\Gamma \Rightarrow \varphi$. By adding a $\neg$L rule, we obtain a derivation of $\neg\varphi, \Gamma \Rightarrow$, i.e., $\Gamma \cup \{\neg\varphi\}$ is inconsistent.

If $\Gamma \cup \{\neg\varphi\}$ is inconsistent, there is a derivation π_1 of $\neg\varphi, \Gamma \Rightarrow$. The following is a derivation of $\Gamma \Rightarrow \varphi$:

$$\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \neg\varphi}\,\neg\mathrm{R} \qquad \begin{array}{c}\vdots\,\pi_1\\ \neg\varphi, \Gamma \Rightarrow\end{array}}{\Gamma \Rightarrow \varphi}\,\mathrm{Cut}$$

□

Problem 6.7. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

fol:seq:prv: prop:explicit-inc **Proposition 6.20.** *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.*

Proof. Suppose $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$. Then there is a derivation π of a sequent $\Gamma_0 \Rightarrow \varphi$. The sequent $\neg\varphi, \Gamma_0 \Rightarrow$ is also derivable:

$$\dfrac{\begin{array}{c}\vdots\,\pi\\ \Gamma_0 \Rightarrow \varphi\end{array} \qquad \dfrac{\dfrac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow}\,\neg\mathrm{L}}{\varphi, \neg\varphi \Rightarrow}\,\mathrm{XL}}{\Gamma, \neg\varphi \Rightarrow}\,\mathrm{Cut}$$

Since $\neg\varphi \in \Gamma$ and $\Gamma_0 \subseteq \Gamma$, this shows that Γ is inconsistent. □

fol:seq:prv: prop:provability-exhaustive **Proposition 6.21.** *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

Proof. There are finite sets $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$ and **LK**-derivations π_0 and π_1 of $\varphi, \Gamma_0 \Rightarrow$ and $\neg\varphi, \Gamma_1 \Rightarrow$, respectively. We can then derive

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\pi_0\\ \varphi, \Gamma_0 \Rightarrow\end{array}}{\Gamma_0 \Rightarrow \neg\varphi}\,\neg\mathrm{R} \qquad \begin{array}{c}\vdots\,\pi_1\\ \neg\varphi, \Gamma_1 \Rightarrow\end{array}}{\Gamma_0, \Gamma_1 \Rightarrow}\,\mathrm{Cut}$$

Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$. Hence Γ is inconsistent. □

6.10 Derivability and the Propositional Connectives

explanation We establish that the derivability relation $\vdash$ of the sequent calculus is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \to \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem. fol:seq:ppr:sec

Proposition 6.22. fol:seq:ppr:prop:provability-land

1. *Both* $\varphi \wedge \psi \vdash \varphi$ *and* $\varphi \wedge \psi \vdash \psi$. fol:seq:ppr:prop:provability-land-left
2. $\varphi, \psi \vdash \varphi \wedge \psi$. fol:seq:ppr:prop:provability-land-right

Proof. 1. Both sequents $\varphi \wedge \psi \Rightarrow \varphi$ and $\varphi \wedge \psi \Rightarrow \psi$ are derivable:

$$\dfrac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi}\wedge\mathrm{L} \qquad \dfrac{\psi \Rightarrow \psi}{\varphi \wedge \psi \Rightarrow \psi}\wedge\mathrm{L}$$

2. Here is a derivation of the sequent $\varphi, \psi \Rightarrow \varphi \wedge \psi$:

$$\dfrac{\varphi \Rightarrow \varphi \qquad \psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \varphi \wedge \psi}\wedge\mathrm{R}$$

□

Proposition 6.23. fol:seq:ppr:prop:provability-lor

1. $\varphi \vee \psi, \neg\varphi, \neg\psi$ *is inconsistent.*
2. *Both* $\varphi \vdash \varphi \vee \psi$ *and* $\psi \vdash \varphi \vee \psi$.

Proof. 1. We give a derivation of the sequent $\varphi \vee \psi, \neg\varphi, \neg\psi \Rightarrow$:

$$\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow}\neg\mathrm{L}}{\varphi, \neg\varphi, \neg\psi \Rightarrow} \qquad \dfrac{\dfrac{\psi \Rightarrow \psi}{\neg\psi, \psi \Rightarrow}\neg\mathrm{L}}{\psi, \neg\varphi, \neg\psi \Rightarrow}}{\varphi \vee \psi, \neg\varphi, \neg\psi \Rightarrow}\vee\mathrm{L}$$

(Recall that double inference lines indicate several weakening, contraction, and exchange inferences.)

2. Both sequents $\varphi \Rightarrow \varphi \vee \psi$ and $\psi \Rightarrow \varphi \vee \psi$ have derivations:

$$\dfrac{\varphi \Rightarrow \varphi}{\varphi \Rightarrow \varphi \vee \psi}\vee\mathrm{R} \qquad \dfrac{\psi \Rightarrow \psi}{\psi \Rightarrow \varphi \vee \psi}\vee\mathrm{R}$$

□

Proposition 6.24. fol:seq:ppr:prop:provability-lif

1. $\varphi, \varphi \to \psi \vdash \psi$. fol:seq:ppr:prop:provability-lif-left

fol:seq:ppr:
prop:provability-lif-right

2. *Both* $\neg\varphi \vdash \varphi \rightarrow \psi$ *and* $\psi \vdash \varphi \rightarrow \psi$.

Proof. 1. The sequent $\varphi \rightarrow \psi, \varphi \Rightarrow \psi$ is derivable:

$$\frac{\varphi \Rightarrow \varphi \qquad \psi \Rightarrow \psi}{\varphi \rightarrow \psi, \varphi \Rightarrow \psi} \rightarrow\text{L}$$

2. Both sequents $\neg\varphi \Rightarrow \varphi \rightarrow \psi$ and $\psi \Rightarrow \varphi \rightarrow \psi$ are derivable:

$$\cfrac{\cfrac{\cfrac{\cfrac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow} \neg\text{L}}{\varphi, \neg\varphi \Rightarrow} \text{XL}}{\varphi, \neg\varphi \Rightarrow \psi} \text{WR}}{\neg\varphi \Rightarrow \varphi \rightarrow \psi} \rightarrow\text{R} \qquad \cfrac{\cfrac{\psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \psi} \text{WL}}{\psi \Rightarrow \varphi \rightarrow \psi} \rightarrow\text{R}$$

□

6.11 Derivability and the Quantifiers

fol:seq:qpr:
sec

The completeness theorem also requires that the sequent calculus rules rules yield the facts about $\vdash$ established in this section. explanation

fol:seq:qpr:
thm:strong-generalization

Theorem 6.25. *If c is a constant not occurring in Γ or $\varphi(x)$ and $\Gamma \vdash \varphi(c)$, then $\Gamma \vdash \forall x\, \varphi(x)$.*

Proof. Let π_0 be an **LK**-derivation of $\Gamma_0 \Rightarrow \varphi(c)$ for some finite $\Gamma_0 \subseteq \Gamma$. By adding a $\forall$R inference, we obtain a derivation of $\Gamma_0 \Rightarrow \forall x\, \varphi(x)$, since c does not occur in Γ or $\varphi(x)$ and thus the eigenvariable condition is satisfied. □

fol:seq:qpr:
prop:provability-quantifiers

Proposition 6.26.

1. $\varphi(t) \vdash \exists x\, \varphi(x)$.

2. $\forall x\, \varphi(x) \vdash \varphi(t)$.

Proof. 1. The sequent $\varphi(t) \Rightarrow \exists x\, \varphi(x)$ is derivable:

$$\frac{\varphi(t) \Rightarrow \varphi(t)}{\varphi(t) \Rightarrow \exists x\, \varphi(x)} \exists\text{R}$$

2. The sequent $\forall x\, \varphi(x) \Rightarrow \varphi(t)$ is derivable:

$$\frac{\varphi(t) \Rightarrow \varphi(t)}{\forall x\, \varphi(x) \Rightarrow \varphi(t)} \forall\text{L}$$

□

6.12 Soundness

explanation A derivation system, such as the sequent calculus, is *sound* if it cannot derive things that do not actually hold. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that fol:seq:sou:sec

1. every derivable φ is valid;
2. if a sentence is derivable from some others, it is also a consequence of them;
3. if a set of sentences is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Because all these proof-theoretic properties are defined via derivability in the sequent calculus of certain sequents, proving (1)–(3) above requires proving something about the semantic properties of derivable sequents. We will first define what it means for a sequent to be *valid*, and then show that every derivable sequent is valid. (1)–(3) then follow as corollaries from this result.

Definition 6.27. A structure $\mathfrak{M}$ *satisfies* a sequent $\Gamma \Rightarrow \Delta$ iff either $\mathfrak{M} \nvDash \varphi$ for some $\varphi \in \Gamma$ or $\mathfrak{M} \vDash \varphi$ for some $\varphi \in \Delta$.

A sequent is *valid* iff every structure $\mathfrak{M}$ satisfies it.

Theorem 6.28 (Soundness). *If* **LK** *derives $\Theta \Rightarrow \Xi$, then $\Theta \Rightarrow \Xi$ is valid.* *fol:seq:sou:thm:sequent-soundness*

Proof. Let π be a derivation of $\Theta \Rightarrow \Xi$. We proceed by induction on the number of inferences n in π.

If the number of inferences is 0, then π consists only of an initial sequent. Every initial sequent $\varphi \Rightarrow \varphi$ is obviously valid, since for every $\mathfrak{M}$, either $\mathfrak{M} \nvDash \varphi$ or $\mathfrak{M} \vDash \varphi$.

If the number of inferences is greater than 0, we distinguish cases according to the type of the lowermost inference. By induction hypothesis, we can assume that the premises of that inference are valid, since the number of inferences in the derivation of any premise is smaller than n.

First, we consider the possible inferences with only one premise.

1. The last inference is a weakening. Then $\Theta \Rightarrow \Xi$ is either $\varphi, \Gamma \Rightarrow \Delta$ (if the last inference is WL) or $\Gamma \Rightarrow \Delta, \varphi$ (if it's WR), and the derivation ends in one of

$$\dfrac{\begin{matrix}\vdots\\ \Gamma \Rightarrow \Delta\end{matrix}}{\varphi, \Gamma \Rightarrow \Delta}\;\mathrm{WL} \qquad \dfrac{\begin{matrix}\vdots\\ \Gamma \Rightarrow \Delta\end{matrix}}{\Gamma \Rightarrow \Delta, \varphi}\;\mathrm{WR}$$

By induction hypothesis, $\Gamma \Rightarrow \Delta$ is valid, i.e., for every structure $\mathfrak{M}$, either there is some $\chi \in \Gamma$ such that $\mathfrak{M} \nvDash \chi$ or there is some $\chi \in \Delta$ such that $\mathfrak{M} \vDash \chi$.

If $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Gamma$, then $\chi \in \Theta$ as well since $\Theta = \varphi, \Gamma$, and so $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Theta$. Similarly, if $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$, as $\chi \in \Xi$, $\mathfrak{M} \vDash \chi$ for some $\chi \in \Xi$. Consequently, $\Theta \Rightarrow \Xi$ is valid.

2. The last inference is $\neg$L: Then the premise of the last inference is $\Gamma \Rightarrow \Delta, \varphi$ and the conclusion is $\neg\varphi, \Gamma \Rightarrow \Delta$, i.e., the derivation ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \varphi\end{array}}{\neg\varphi, \Gamma \Rightarrow \Delta}\ \neg\mathrm{L}$$

and $\Theta = \neg\varphi, \Gamma$ while $\Xi = \Delta$.

The induction hypothesis tells us that $\Gamma \Rightarrow \Delta, \varphi$ is valid, i.e., for every $\mathfrak{M}$, either (a) for some $\chi \in \Gamma$, $\mathfrak{M} \nvDash \chi$, or (b) for some $\chi \in \Delta$, $\mathfrak{M} \vDash \chi$, or (c) $\mathfrak{M} \vDash \varphi$. We want to show that $\Theta \Rightarrow \Xi$ is also valid. Let $\mathfrak{M}$ be a structure. If (a) holds, then there is $\chi \in \Gamma$ so that $\mathfrak{M} \nvDash \chi$, but $\chi \in \Theta$ as well. If (b) holds, there is $\chi \in \Delta$ such that $\mathfrak{M} \vDash \chi$, but $\chi \in \Xi$ as well. Finally, if $\mathfrak{M} \vDash \varphi$, then $\mathfrak{M} \nvDash \neg\varphi$. Since $\neg\varphi \in \Theta$, there is $\chi \in \Theta$ such that $\mathfrak{M} \nvDash \chi$. Consequently, $\Theta \Rightarrow \Xi$ is valid.

3. The last inference is $\neg$R: Exercise.

4. The last inference is $\wedge$L: There are two variants: $\varphi \wedge \psi$ may be inferred on the left from φ or from ψ on the left side of the premise. In the first case, the π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \varphi, \Gamma \Rightarrow \Delta\end{array}}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}\ \wedge\mathrm{L}$$

and $\Theta = \varphi \wedge \psi, \Gamma$ while $\Xi = \Delta$. Consider a structure $\mathfrak{M}$. Since by induction hypothesis, $\varphi, \Gamma \Rightarrow \Delta$ is valid, (a) $\mathfrak{M} \nvDash \varphi$, (b) $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Gamma$, or (c) $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$. In case (a), $\mathfrak{M} \nvDash \varphi \wedge \psi$, so there is $\chi \in \Theta$ (namely, $\varphi \wedge \psi$) such that $\mathfrak{M} \nvDash \chi$. In case (b), there is $\chi \in \Gamma$ such that $\mathfrak{M} \nvDash \chi$, and $\chi \in \Theta$ as well. In case (c), there is $\chi \in \Delta$ such that $\mathfrak{M} \vDash \chi$, and $\chi \in \Xi$ as well since $\Xi = \Delta$. So in each case, $\mathfrak{M}$ satisfies $\varphi \wedge \psi, \Gamma \Rightarrow \Delta$. Since $\mathfrak{M}$ was arbitrary, $\Gamma \Rightarrow \Delta$ is valid. The case where $\varphi \wedge \psi$ is inferred from ψ is handled the same, changing φ to ψ.

5. The last inference is $\lor$R: There are two variants: $\varphi \lor \psi$ may be inferred on the right from φ or from ψ on the right side of the premise. In the first case, π ends in

$$\cfrac{\begin{matrix}\vdots\\ \Gamma \Rightarrow \Delta, \varphi\end{matrix}}{\Gamma \Rightarrow \Delta, \varphi \lor \psi}\ \lor\text{R}$$

Now $\Theta = \Gamma$ and $\Xi = \Delta, \varphi \lor \psi$. Consider a structure $\mathfrak{M}$. Since $\Gamma \Rightarrow \Delta, \varphi$ is valid, (a) $\mathfrak{M} \vDash \varphi$, (b) $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Gamma$, or (c) $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$. In case (a), $\mathfrak{M} \vDash \varphi \lor \psi$. In case (b), there is $\chi \in \Gamma$ such that $\mathfrak{M} \nvDash \chi$. In case (c), there is $\chi \in \Delta$ such that $\mathfrak{M} \vDash \chi$. So in each case, $\mathfrak{M}$ satisfies $\Gamma \Rightarrow \Delta, \varphi \lor \psi$, i.e., $\Theta \Rightarrow \Xi$. Since $\mathfrak{M}$ was arbitrary, $\Theta \Rightarrow \Xi$ is valid. The case where $\varphi \lor \psi$ is inferred from ψ is handled the same, changing φ to ψ.

6. The last inference is $\rightarrow$R: Then π ends in

$$\cfrac{\begin{matrix}\vdots\\ \varphi, \Gamma \Rightarrow \Delta, \psi\end{matrix}}{\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi}\ \rightarrow\text{R}$$

Again, the induction hypothesis says that the premise is valid; we want to show that the conclusion is valid as well. Let $\mathfrak{M}$ be arbitrary. Since $\varphi, \Gamma \Rightarrow \Delta, \psi$ is valid, at least one of the following cases obtains: (a) $\mathfrak{M} \nvDash \varphi$, (b) $\mathfrak{M} \vDash \psi$, (c) $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Gamma$, or (d) $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$. In cases (a) and (b), $\mathfrak{M} \vDash \varphi \rightarrow \psi$ and so there is a $\chi \in \Delta, \varphi \rightarrow \psi$ such that $\mathfrak{M} \vDash \chi$. In case (c), for some $\chi \in \Gamma$, $\mathfrak{M} \nvDash \chi$. In case (d), for some $\chi \in \Delta$, $\mathfrak{M} \vDash \chi$. In each case, $\mathfrak{M}$ satisfies $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$. Since $\mathfrak{M}$ was arbitrary, $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$ is valid.

7. The last inference is $\forall$L: Then there is a formula $\varphi(x)$ and a closed term t such that π ends in

$$\cfrac{\begin{matrix}\vdots\\ \varphi(t), \Gamma \Rightarrow \Delta\end{matrix}}{\forall x\, \varphi(x), \Gamma \Rightarrow \Delta}\ \forall\text{L}$$

We want to show that the conclusion $\forall x\, \varphi(x), \Gamma \Rightarrow \Delta$ is valid. Consider a structure $\mathfrak{M}$. Since the premise $\varphi(t), \Gamma \Rightarrow \Delta$ is valid, (a) $\mathfrak{M} \nvDash \varphi(t)$, (b) $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Gamma$, or (c) $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$. In case (a), by Proposition 3.30, if $\mathfrak{M} \vDash \forall x\, \varphi(x)$, then $\mathfrak{M} \vDash \varphi(t)$. Since $\mathfrak{M} \nvDash \varphi(t)$,

$\mathfrak{M} \nvDash \forall x\, \varphi(x)$. In case (b) and (c), $\mathfrak{M}$ also satisfies $\forall x\, \varphi(x), \Gamma \Rightarrow \Delta$. Since $\mathfrak{M}$ was arbitrary, $\forall x\, \varphi(x), \Gamma \Rightarrow \Delta$ is valid.

8. The last inference is $\exists$R: Exercise.

9. The last inference is $\forall$R: Then there is a formula $\varphi(x)$ and a constant symbol a such that π ends in

$$\dfrac{\begin{matrix}\vdots \\ \Gamma \Rightarrow \Delta, \varphi(a)\end{matrix}}{\Gamma \Rightarrow \Delta, \forall x\, \varphi(x)}\ \forall\text{R}$$

where the eigenvariable condition is satisfied, i.e., a does not occur in $\varphi(x)$, Γ, or Δ. By induction hypothesis, the premise of the last inference is valid. We have to show that the conclusion is valid as well, i.e., that for any structure $\mathfrak{M}$, (a) $\mathfrak{M} \vDash \forall x\, \varphi(x)$, (b) $\mathfrak{M} \nvDash \chi$ for some $\chi \in \Gamma$, or (c) $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$.

Suppose $\mathfrak{M}$ is an arbitrary structure. If (b) or (c) holds, we are done, so suppose neither holds: for all $\chi \in \Gamma$, $\mathfrak{M} \vDash \chi$, and for all $\chi \in \Delta$, $\mathfrak{M} \nvDash \chi$. We have to show that (a) holds, i.e., $\mathfrak{M} \vDash \forall x\, \varphi(x)$. By Proposition 3.18, if suffices to show that $\mathfrak{M}, s \vDash \varphi(x)$ for all variable assignments s. So let s be an arbitrary variable assignment. Consider the structure $\mathfrak{M}'$ which is just like $\mathfrak{M}$ except $a^{\mathfrak{M}'} = s(x)$. By Corollary 3.20, for any $\chi \in \Gamma$, $\mathfrak{M}' \vDash \chi$ since a does not occur in Γ, and for any $\chi \in \Delta$, $\mathfrak{M}' \nvDash \chi$. But the premise is valid, so $\mathfrak{M}' \vDash \varphi(a)$. By Proposition 3.17, $\mathfrak{M}', s \vDash \varphi(a)$, since $\varphi(a)$ is a sentence. Now $s \sim_x s$ with $s(x) = \mathrm{Val}_s^{\mathfrak{M}'}(a)$, since we've defined $\mathfrak{M}'$ in just this way. So Proposition 3.22 applies, and we get $\mathfrak{M}', s \vDash \varphi(x)$. Since a does not occur in $\varphi(x)$, by Proposition 3.19, $\mathfrak{M}, s \vDash \varphi(x)$. Since s was arbitrary, we've completed the proof that $\mathfrak{M}, s \vDash \varphi(x)$ for all variable assignments.

10. The last inference is $\exists$L: Exercise.

Now let's consider the possible inferences with two premises.

1. The last inference is a cut: then π ends in

$$\dfrac{\begin{matrix}\vdots \\ \Gamma \Rightarrow \Delta, \varphi\end{matrix} \qquad \begin{matrix}\vdots \\ \varphi, \Pi \Rightarrow \Lambda\end{matrix}}{\Gamma, \Pi \Rightarrow \Delta, \Lambda}\ \text{Cut}$$

Let $\mathfrak{M}$ be a structure. By induction hypothesis, the premises are valid, so $\mathfrak{M}$ satisfies both premises. We distinguish two cases: (a) $\mathfrak{M} \nvDash \varphi$ and (b) $\mathfrak{M} \vDash \varphi$. In case (a), in order for $\mathfrak{M}$ to satisfy the left premise, it must satisfy $\Gamma \Rightarrow \Delta$. But then it also satisfies the conclusion. In case (b), in order for $\mathfrak{M}$ to satisfy the right premise, it must satisfy $\Pi \setminus \Lambda$. Again, $\mathfrak{M}$ satisfies the conclusion.

2. The last inference is $\wedge$R. Then π ends in

$$\dfrac{\begin{matrix}\vdots \\ \Gamma \Rightarrow \Delta, \varphi\end{matrix} \qquad \begin{matrix}\vdots \\ \Gamma \Rightarrow \Delta, \psi\end{matrix}}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \wedge\mathrm{R}$$

   Consider a structure $\mathfrak{M}$. If $\mathfrak{M}$ satisfies $\Gamma \Rightarrow \Delta$, we are done. So suppose it doesn't. Since $\Gamma \Rightarrow \Delta, \varphi$ is valid by induction hypothesis, $\mathfrak{M} \vDash \varphi$. Similarly, since $\Gamma \Rightarrow \Delta, \psi$ is valid, $\mathfrak{M} \vDash \psi$. But then $\mathfrak{M} \vDash \varphi \wedge \psi$.

3. The last inference is $\vee$L: Exercise.

4. The last inference is $\rightarrow$L. Then π ends in

$$\dfrac{\begin{matrix}\vdots \\ \Gamma \Rightarrow \Delta, \varphi\end{matrix} \qquad \begin{matrix}\vdots \\ \psi, \Pi \Rightarrow \Lambda\end{matrix}}{\varphi \rightarrow \psi, \Gamma, \Pi \Rightarrow \Delta, \Lambda} \rightarrow\mathrm{L}$$

   Again, consider a structure $\mathfrak{M}$ and suppose $\mathfrak{M}$ doesn't satisfy $\Gamma, \Pi \Rightarrow \Delta, \Lambda$. We have to show that $\mathfrak{M} \nvDash \varphi \rightarrow \psi$. If $\mathfrak{M}$ doesn't satisfy $\Gamma, \Pi \Rightarrow \Delta, \Lambda$, it satisfies neither $\Gamma \Rightarrow \Delta$ nor $\Pi \Rightarrow \Lambda$. Since, $\Gamma \Rightarrow \Delta, \varphi$ is valid, we have $\mathfrak{M} \vDash \varphi$. Since $\psi, \Pi \Rightarrow \Lambda$ is valid, we have $\mathfrak{M} \nvDash \psi$. But then $\mathfrak{M} \nvDash \varphi \rightarrow \psi$, which is what we wanted to show. □

Problem 6.8. Complete the proof of Theorem 6.28.

Corollary 6.29. *If $\vdash \varphi$ then φ is valid.* fol:seq:sou:cor:weak-soundness

Corollary 6.30. *If $\Gamma \vdash \varphi$ then $\Gamma \vDash \varphi$.* fol:seq:sou:cor:entailment-soundness

Proof. If $\Gamma \vdash \varphi$ then for some finite subset $\Gamma_0 \subseteq \Gamma$, there is a derivation of $\Gamma_0 \Rightarrow \varphi$. By Theorem 6.28, every structure $\mathfrak{M}$ either makes some $\psi \in \Gamma_0$ false or makes φ true. Hence, if $\mathfrak{M} \vDash \Gamma$ then also $\mathfrak{M} \vDash \varphi$. □

Corollary 6.31. *If Γ is satisfiable, then it is consistent.* fol:seq:sou:cor:consistency-soundness

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then there is a finite $\Gamma_0 \subseteq \Gamma$ and a derivation of $\Gamma_0 \Rightarrow$. By Theorem 6.28, $\Gamma_0 \Rightarrow$ is valid. In other words, for every structure $\mathfrak{M}$, there is $\chi \in \Gamma_0$ so that $\mathfrak{M} \nvDash \chi$, and since $\Gamma_0 \subseteq \Gamma$, that χ is also in Γ. Thus, no $\mathfrak{M}$ satisfies Γ, and Γ is not satisfiable. □

6.13 Derivations with Identity predicate

fol:seq:ide: sec

Derivations with identity predicate require additional initial sequents and inference rules.

Definition 6.32 (Initial sequents for $=$). If t is a closed term, then $\Rightarrow t = t$ is an initial sequent.

The rules for $=$ are (t_1 and t_2 are closed terms):

$$\dfrac{t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_1)}{t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_2)} = \qquad \dfrac{t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_2)}{t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_1)} =$$

Example 6.33. If s and t are closed terms, then $s = t, \varphi(s) \vdash \varphi(t)$:

$$\dfrac{\dfrac{\varphi(s) \Rightarrow \varphi(s)}{s = t, \varphi(s) \Rightarrow \varphi(s)}\,\mathrm{WL}}{s = t, \varphi(s) \Rightarrow \varphi(t)} =$$

This may be familiar as the principle of substitutability of identicals, or Leibniz' Law.

LK proves that $=$ is symmetric and transitive:

$$\dfrac{\dfrac{\Rightarrow t_1 = t_1}{t_1 = t_2 \Rightarrow t_1 = t_1}\,\mathrm{WL}}{t_1 = t_2 \Rightarrow t_2 = t_1} = \qquad \dfrac{\dfrac{\dfrac{t_1 = t_2 \Rightarrow t_1 = t_2}{t_2 = t_3, t_1 = t_2 \Rightarrow t_1 = t_2}\,\mathrm{WL}}{t_2 = t_3, t_1 = t_2 \Rightarrow t_1 = t_3} =}{t_1 = t_2, t_2 = t_3 \Rightarrow t_1 = t_3}\,\mathrm{XL}$$

In the derivation on the left, the formula $x = t_1$ is our $\varphi(x)$. On the right, we take $\varphi(x)$ to be $t_1 = x$.

Problem 6.9. Give derivations of the following sequents:

1. $\Rightarrow \forall x\, \forall y\, ((x = y \land \varphi(x)) \to \varphi(y))$
2. $\exists x\, \varphi(x) \land \forall y\, \forall z\, ((\varphi(y) \land \varphi(z)) \to y = z) \Rightarrow \exists x\, (\varphi(x) \land \forall y\, (\varphi(y) \to y = x))$

6.14 Soundness with Identity predicate

fol:seq:sid: sec

Proposition 6.34. **LK** *with initial sequents and rules for identity is sound.*

Proof. Initial sequents of the form $\Rightarrow t = t$ are valid, since for every structure $\mathfrak{M}$, $\mathfrak{M} \vDash t = t$. (Note that we assume the term t to be closed, i.e., it contains no variables, so variable assignments are irrelevant).

Suppose the last inference in a derivation is $=$. Then the premise is $t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_1)$ and the conclusion is $t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_2)$. Consider a structure $\mathfrak{M}$. We need to show that the conclusion is valid, i.e., if $\mathfrak{M} \vDash t_1 = t_2$ and $\mathfrak{M} \vDash \Gamma$, then either $\mathfrak{M} \vDash \chi$ for some $\chi \in \Delta$ or $\mathfrak{M} \vDash \varphi(t_2)$.

By induction hypothesis, the premise is valid. This means that if $\mathfrak{M} \vDash t_1 = t_2$ and $\mathfrak{M} \vDash \Gamma$ either (a) for some $\chi \in \Delta$, $\mathfrak{M} \vDash \chi$ or (b) $\mathfrak{M} \vDash \varphi(t_1)$. In case (a) we are done. Consider case (b). Let s be a variable assignment with $s(x) = \mathrm{Val}^{\mathfrak{M}}(t_1)$. By Proposition 3.17, $\mathfrak{M}, s \vDash \varphi(t_1)$. Since $s \sim_x s$, by Proposition 3.22, $\mathfrak{M}, s \vDash \varphi(x)$. since $\mathfrak{M} \vDash t_1 = t_2$, we have $\mathrm{Val}^{\mathfrak{M}}(t_1) = \mathrm{Val}^{\mathfrak{M}}(t_2)$, and hence $s(x) = \mathrm{Val}^{\mathfrak{M}}(t_2)$. By applying Proposition 3.22 again, we also have $\mathfrak{M}, s \vDash \varphi(t_2)$. By Proposition 3.17, $\mathfrak{M} \vDash \varphi(t_2)$. $\square$

Chapter 7

Natural Deduction

This chapter presents a natural deduction system in the style of Gentzen/Prawitz.

To include or exclude material relevant to natural deduction as a proof system, use the "prfND" tag.

7.1 Rules and Derivations

fol:ntd:rul:sec Natural deduction systems are meant to closely parallel the informal reasoning used in mathematical proof (hence it is somewhat "natural"). Natural deduction proofs begin with assumptions. Inference rules are then applied. Assumptions are "discharged" by the ¬Intro, →Intro, ∨Elim and ∃Elim inference rules, and the label of the discharged assumption is placed beside the inference for clarity. explanation

Definition 7.1 (Assumption). An *assumption* is any sentence in the topmost position of any branch.

Derivations in natural deduction are certain trees of sentences, where the topmost sentences are assumptions, and if a sentence stands below one, two, or three other sequents, it must follow correctly by a rule of inference. The sentences at the top of the inference are called the *premises* and the sentence below the *conclusion* of the inference. The rules come in pairs, an introduction and an elimination rule for each logical operator. They introduce a logical operator in the conclusion or remove a logical operator from a premise of the rule. Some of the rules allow an assumption of a certain type to be *discharged*. To indicate which assumption is discharged by which inference, we also assign labels to both the assumption and the inference. This is indicated by writing the assumption as "$[\varphi]^n$."

It is customary to consider rules for all the logical operators $\land$, $\lor$, $\to$, $\neg$, and $\bot$, even if some of those are defined.

7.2 Propositional Rules

Rules for $\wedge$

fol:ntd:prl:sec

$$\frac{\varphi \qquad \psi}{\varphi \wedge \psi}\ \wedge\text{Intro}$$

$$\frac{\varphi \wedge \psi}{\varphi}\ \wedge\text{Elim} \qquad \frac{\varphi \wedge \psi}{\psi}\ \wedge\text{Elim}$$

Rules for $\vee$

$$\frac{\varphi}{\varphi \vee \psi}\ \vee\text{Intro} \qquad \frac{\psi}{\varphi \vee \psi}\ \vee\text{Intro}$$

$$\begin{array}{ccc} & [\varphi]^n & [\psi]^n \\ & \vdots & \vdots \end{array}$$
$$n\ \frac{\varphi \vee \psi \qquad \chi \qquad \chi}{\chi}\ \vee\text{Elim}$$

Rules for $\rightarrow$

$$\begin{array}{c} [\varphi]^n \\ \vdots \end{array}$$
$$n\ \frac{\psi}{\varphi \rightarrow \psi}\ \rightarrow\text{Intro}$$

$$\frac{\varphi \rightarrow \psi \qquad \varphi}{\psi}\ \rightarrow\text{Elim}$$

Rules for $\neg$

$$\begin{array}{c} [\varphi]^n \\ \vdots \end{array}$$
$$n\ \frac{\bot}{\neg\varphi}\ \neg\text{Intro}$$

$$\frac{\neg\varphi \qquad \varphi}{\bot}\ \neg\text{Elim}$$

Rules for $\perp$

$$\frac{\perp}{\varphi}\ \perp_I \qquad\qquad \begin{array}{c}[\neg\varphi]^n \\ \vdots \\ n\ \dfrac{\perp}{\varphi}\ \perp_C\end{array}$$

Note that $\neg$Intro and $\perp_C$ are very similar: The difference is that $\neg$Intro derives a negated sentence $\neg\varphi$ but $\perp_C$ a positive sentence φ.

Whenever a rule indicates that some assumption may be discharged, we take this to be a permission, but not a requirement. E.g., in the $\rightarrow$Intro rule, we may discharge any number of assumptions of the form φ in the derivation of the premise ψ, including zero.

7.3 Quantifier Rules

fol:ntd:qrl: sec

Rules for $\forall$

$$\frac{\varphi(a)}{\forall x\,\varphi(x)}\ \forall\text{Intro} \qquad\qquad \frac{\forall x\,\varphi(x)}{\varphi(t)}\ \forall\text{Elim}$$

In the rules for $\forall$, t is a closed term (a term that does not contain any variables), and a is a constant symbol which does not occur in the conclusion $\forall x\,\varphi(x)$, or in any assumption which is undischarged in the derivation ending with the premise $\varphi(a)$. We call a the *eigenvariable* of the $\forall$Intro inference.[1]

Rules for $\exists$

$$\frac{\varphi(t)}{\exists x\,\varphi(x)}\ \exists\text{Intro} \qquad\qquad n\ \dfrac{\exists x\,\varphi(x) \qquad \begin{array}{c}[\varphi(a)]^n \\ \vdots \\ \chi\end{array}}{\chi}\ \exists\text{Elim}$$

Again, t is a closed term, and a is a constant which does not occur in the premise $\exists x\,\varphi(x)$, in the conclusion χ, or any assumption which is undischarged

[1] We use the term "eigenvariable" even though a in the above rule is a constant. This has historical reasons.

in the derivations ending with the two premises (other than the assumptions $\varphi(a)$). We call a the *eigenvariable* of the $\exists$Elim inference.

The condition that an eigenvariable neither occur in the premises nor in any assumption that is undischarged in the derivations leading to the premises for the $\forall$Intro or $\exists$Elim inference is called the *eigenvariable condition*.

explanation Recall the convention that when φ is a formula with the variable x free, we indicate this by writing $\varphi(x)$. In the same context, $\varphi(t)$ then is short for $\varphi[t/x]$. So we could also write the $\exists$Intro rule as:

$$\cfrac{\varphi[t/x]}{\exists x\, \varphi}\ \exists\text{Intro}$$

Note that t may already occur in φ, e.g., φ might be $P(t, x)$. Thus, inferring $\exists x\, P(t, x)$ from $P(t, t)$ is a correct application of $\exists$Intro—you may "replace" one or more, and not necessarily all, occurrences of t in the premise by the bound variable x. However, the eigenvariable conditions in $\forall$Intro and $\exists$Elim require that the constant symbol a does not occur in φ. So, you cannot correctly infer $\forall x\, P(a, x)$ from $P(a, a)$ using $\forall$Intro.

explanation In $\exists$Intro and $\forall$Elim there are no restrictions, and the term t can be anything, so we do not have to worry about any conditions. On the other hand, in the $\exists$Elim and $\forall$Intro rules, the eigenvariable condition requires that the constant symbol a does not occur anywhere in the conclusion or in an undischarged assumption. The condition is necessary to ensure that the system is sound, i.e., only derives sentences from undischarged assumptions from which they follow. Without this condition, the following would be allowed:

$$\cfrac{\exists x\, \varphi(x) \qquad \cfrac{[\varphi(a)]^1}{\forall x\, \varphi(x)}\ {*}\forall\text{Intro}}{\forall x\, \varphi(x)}\ \exists\text{Elim}$$

However, $\exists x\, \varphi(x) \nvDash \forall x\, \varphi(x)$.

As the elimination rules for quantifiers only allow substituting closed terms for variables, it follows that any formula that can be derived from a set of sentences is itself a sentence.

7.4 Derivations

explanation We've said what an assumption is, and we've given the rules of inference. Derivations in natural deduction are inductively generated from these: each derivation either is an assumption on its own, or consists of one, two, or three derivations followed by a correct inference. fol:ntd:der:sec

Definition 7.2 (Derivation). A *derivation* of a sentence φ from assumptions Γ is a finite tree of sentences satisfying the following conditions:

1. The topmost sentences of the tree are either in Γ or are discharged by an inference in the tree.

2. The bottommost sentence of the tree is φ.

3. Every sentence in the tree except the sentence φ at the bottom is a premise of a correct application of an inference rule whose conclusion stands directly below that sentence in the tree.

We then say that φ is the *conclusion* of the derivation and Γ its undischarged assumptions.

If a derivation of φ from Γ exists, we say that φ is *derivable* from Γ, or in symbols: $\Gamma \vdash \varphi$. If there is a derivation of φ in which every assumption is discharged, we write $\vdash \varphi$.

Example 7.3. Every assumption on its own is a derivation. So, e.g., φ by itself is a derivation, and so is ψ by itself. We can obtain a new derivation from these by applying, say, the $\wedge$Intro rule,

$$\dfrac{\varphi \qquad \psi}{\varphi \wedge \psi}\ \wedge\text{Intro}$$

These rules are meant to be general: we can replace the φ and ψ in it with any sentences, e.g., by χ and θ. Then the conclusion would be $\chi \wedge \theta$, and so

$$\dfrac{\chi \qquad \theta}{\chi \wedge \theta}\ \wedge\text{Intro}$$

is a correct derivation. Of course, we can also switch the assumptions, so that θ plays the role of φ and χ that of ψ. Thus,

$$\dfrac{\theta \qquad \chi}{\theta \wedge \chi}\ \wedge\text{Intro}$$

is also a correct derivation.

We can now apply another rule, say, $\to$Intro, which allows us to conclude a conditional and allows us to discharge any assumption that is identical to the antecedent of that conditional. So both of the following would be correct derivations:

$$1\ \dfrac{\dfrac{[\chi]^1 \qquad \theta}{\chi \wedge \theta}\ \wedge\text{Intro}}{\chi \to (\chi \wedge \theta)}\ \to\text{Intro} \qquad 1\ \dfrac{\dfrac{\chi \qquad [\theta]^1}{\chi \wedge \theta}\ \wedge\text{Intro}}{\theta \to (\chi \wedge \theta)}\ \to\text{Intro}$$

They show, respectively, that $\theta \vdash \chi \to (\chi \wedge \theta)$ and $\chi \vdash \theta \to (\chi \wedge \theta)$.

Remember that discharging of assumptions is a permission, not a requirement: we don't have to discharge the assumptions. In particular, we can apply a rule even if the assumptions are not present in the derivation. For instance, the following is legal, even though there is no assumption φ to be discharged:

$$1\ \dfrac{\psi}{\varphi \to \psi}\ \to\text{Intro}$$

7.5 Examples of Derivations

fol:ntd:pro:sec

Example 7.4. Let's give a derivation of the sentence $(\varphi \wedge \psi) \to \varphi$.

We begin by writing the desired conclusion at the bottom of the derivation.

$$\frac{}{(\varphi \wedge \psi) \to \varphi}$$

Next, we need to figure out what kind of inference could result in a sentence of this form. The main operator of the conclusion is $\to$, so we'll try to arrive at the conclusion using the $\to$Intro rule. It is best to write down the assumptions involved and label the inference rules as you progress, so it is easy to see whether all assumptions have been discharged at the end of the proof.

$$\begin{array}{c} [\varphi \wedge \psi]^1 \\ \vdots \\ {}_1\,\dfrac{\varphi}{(\varphi \wedge \psi) \to \varphi}\,\to\text{Intro} \end{array}$$

We now need to fill in the steps from the assumption $\varphi \wedge \psi$ to φ. Since we only have one connective to deal with, $\wedge$, we must use the $\wedge$ elim rule. This gives us the following proof:

$${}_1\,\dfrac{\dfrac{[\varphi \wedge \psi]^1}{\varphi}\,\wedge\text{Elim}}{(\varphi \wedge \psi) \to \varphi}\,\to\text{Intro}$$

We now have a correct derivation of $(\varphi \wedge \psi) \to \varphi$.

Example 7.5. Now let's give a derivation of $(\neg\varphi \vee \psi) \to (\varphi \to \psi)$.

We begin by writing the desired conclusion at the bottom of the derivation.

$$\frac{}{(\neg\varphi \vee \psi) \to (\varphi \to \psi)}$$

To find a logical rule that could give us this conclusion, we look at the logical connectives in the conclusion: $\neg$, $\vee$, and $\to$. We only care at the moment about the first occurrence of $\to$ because it is the main operator of the sentence in the end-sequent, while $\neg$, $\vee$ and the second occurrence of $\to$ are inside the scope of another connective, so we will take care of those later. We therefore start with the $\to$Intro rule. A correct application must look like this:

$$\begin{array}{c} [\neg\varphi \vee \psi]^1 \\ \vdots \\ {}_1\,\dfrac{\varphi \to \psi}{(\neg\varphi \vee \psi) \to (\varphi \to \psi)}\,\to\text{Intro} \end{array}$$

This leaves us with two possibilities to continue. Either we can keep working from the bottom up and look for another application of the $\to$Intro rule, or we can work from the top down and apply a $\lor$Elim rule. Let us apply the latter. We will use the assumption $\neg\varphi \lor \psi$ as the leftmost premise of $\lor$Elim. For a valid application of $\lor$Elim, the other two premises must be identical to the conclusion $\varphi \to \psi$, but each may be derived in turn from another assumption, namely one of the two disjuncts of $\neg\varphi \lor \psi$. So our derivation will look like this:

$$
{}_1\dfrac{{}_2\dfrac{[\neg\varphi \lor \psi]^1 \qquad \begin{matrix}[\neg\varphi]^2 \\ \vdots \\ \varphi \to \psi\end{matrix} \qquad \begin{matrix}[\psi]^2 \\ \vdots \\ \varphi \to \psi\end{matrix}}{\varphi \to \psi}\ \lor\mathrm{Elim}}{(\neg\varphi \lor \psi) \to (\varphi \to \psi)}\ {\to}\mathrm{Intro}
$$

In each of the two branches on the right, we want to derive $\varphi \to \psi$, which is best done using $\to$Intro.

$$
{}_1\dfrac{{}_2\dfrac{[\neg\varphi \lor \psi]^1 \qquad {}_3\dfrac{\begin{matrix}[\neg\varphi]^2, [\varphi]^3 \\ \vdots \\ \psi\end{matrix}}{\varphi \to \psi}\ {\to}\mathrm{Intro} \qquad {}_4\dfrac{\begin{matrix}[\psi]^2, [\varphi]^4 \\ \vdots \\ \psi\end{matrix}}{\varphi \to \psi}\ {\to}\mathrm{Intro}}{\varphi \to \psi}\ \lor\mathrm{Elim}}{(\neg\varphi \lor \psi) \to (\varphi \to \psi)}\ {\to}\mathrm{Intro}
$$

For the two missing parts of the derivation, we need derivations of ψ from $\neg\varphi$ and φ in the middle, and from φ and ψ on the left. Let's take the former first. $\neg\varphi$ and φ are the two premises of $\neg$Elim:

$$
\begin{matrix}\dfrac{[\neg\varphi]^2 \qquad [\varphi]^3}{\bot}\ \neg\mathrm{Elim} \\ \vdots \\ \psi\end{matrix}
$$

By using $\bot_I$, we can obtain ψ as a conclusion and complete the branch.

$$
{}_1\dfrac{{}_2\dfrac{[\neg\varphi \lor \psi]^1 \qquad {}_3\dfrac{\dfrac{\dfrac{[\neg\varphi]^2 \qquad [\varphi]^3}{\bot}\ \bot\mathrm{Intro}}{\psi}\ \bot_I}{\varphi \to \psi}\ {\to}\mathrm{Intro} \qquad {}_4\dfrac{\begin{matrix}[\psi]^2, [\varphi]^4 \\ \vdots \\ \psi\end{matrix}}{\varphi \to \psi}\ {\to}\mathrm{Intro}}{\varphi \to \psi}\ \lor\mathrm{Elim}}{(\neg\varphi \lor \psi) \to (\varphi \to \psi)}\ {\to}\mathrm{Intro}
$$

Let's now look at the rightmost branch. Here it's important to realize that the definition of derivation *allows assumptions to be discharged* but *does not require* them to be. In other words, if we can derive ψ from one of the assumptions φ and ψ without using the other, that's ok. And to derive ψ from ψ is trivial: ψ by itself is such a derivation, and no inferences are needed. So we can simply delete the assumption φ.

$$
{}^{1}\,\dfrac{{}^{2}\,\dfrac{[\neg\varphi \lor \psi]^1 \qquad {}^{3}\,\dfrac{\dfrac{\dfrac{[\neg\varphi]^2 \qquad [\varphi]^3}{\bot}\,\neg\mathrm{Elim}}{\psi}\,\bot_I}{\varphi \to \psi}\,{\to}\mathrm{Intro} \qquad \dfrac{[\psi]^2}{\varphi \to \psi}\,{\to}\mathrm{Intro}}{\varphi \to \psi}\,\lor\mathrm{Elim}}{(\neg\varphi \lor \psi) \to (\varphi \to \psi)}\,{\to}\mathrm{Intro}
$$

Note that in the finished derivation, the rightmost $\to$Intro inference does not actually discharge any assumptions.

Example 7.6. So far we have not needed the $\bot_C$ rule. It is special in that it allows us to discharge an assumption that isn't a sub-formula of the conclusion of the rule. It is closely related to the $\bot_I$ rule. In fact, the $\bot_I$ rule is a special case of the $\bot_C$ rule—there is a logic called “intuitionistic logic” in which only $\bot_I$ is allowed. The $\bot_C$ rule is a last resort when nothing else works. For instance, suppose we want to derive $\varphi \lor \neg\varphi$. Our usual strategy would be to attempt to derive $\varphi \lor \neg\varphi$ using $\lor$Intro. But this would require us to derive either φ or $\neg\varphi$ from no assumptions, and this can't be done. $\bot_C$ to the rescue!

$$
\begin{array}{c}
[\neg(\varphi \lor \neg\varphi)]^1 \\
\vdots \\
{}^{1}\,\dfrac{\bot}{\varphi \lor \neg\varphi}\,\bot_C
\end{array}
$$

Now we're looking for a derivation of $\bot$ from $\neg(\varphi \lor \neg\varphi)$. Since $\bot$ is the conclusion of $\neg$Elim we might try that:

$$
{}^{1}\,\dfrac{\dfrac{\begin{array}{c}[\neg(\varphi \lor \neg\varphi)]^1 \\ \vdots \\ \neg\varphi\end{array} \qquad \begin{array}{c}[\neg(\varphi \lor \neg\varphi)]^1 \\ \vdots \\ \varphi\end{array}}{\bot}\,\neg\mathrm{Elim}}{\varphi \lor \neg\varphi}\,\bot_C
$$

Our strategy for finding a derivation of $\neg\varphi$ calls for an application of $\neg$Intro:

$$
\begin{prooftree}
\AxiomC{$[\neg(\varphi \lor \neg\varphi)]^1, [\varphi]^2$}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\bot$}
\LeftLabel{2}
\RightLabel{$\neg$Intro}
\UnaryInfC{$\neg\varphi$}
\AxiomC{$[\neg(\varphi \lor \neg\varphi)]^1$}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{φ}
\RightLabel{$\neg$Elim}
\BinaryInfC{$\bot$}
\LeftLabel{1}
\RightLabel{$\bot_C$}
\UnaryInfC{$\varphi \lor \neg\varphi$}
\end{prooftree}
$$

Here, we can get $\bot$ easily by applying $\neg$Elim to the assumption $\neg(\varphi \lor \neg\varphi)$ and $\varphi \lor \neg\varphi$ which follows from our new assumption φ by $\lor$Intro:

$$
\begin{prooftree}
\AxiomC{$[\neg(\varphi \lor \neg\varphi)]^1$}
\AxiomC{$[\varphi]^2$}
\RightLabel{$\lor$Intro}
\UnaryInfC{$\varphi \lor \neg\varphi$}
\RightLabel{$\neg$Elim}
\BinaryInfC{$\bot$}
\LeftLabel{2}
\RightLabel{$\neg$Intro}
\UnaryInfC{$\neg\varphi$}
\AxiomC{$[\neg(\varphi \lor \neg\varphi)]^1$}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{φ}
\RightLabel{$\neg$Elim}
\BinaryInfC{$\bot$}
\LeftLabel{1}
\RightLabel{$\bot_C$}
\UnaryInfC{$\varphi \lor \neg\varphi$}
\end{prooftree}
$$

On the right side we use the same strategy, except we get φ by $\bot_C$:

$$
\begin{prooftree}
\AxiomC{$[\neg(\varphi \lor \neg\varphi)]^1$}
\AxiomC{$[\varphi]^2$}
\RightLabel{$\lor$Intro}
\UnaryInfC{$\varphi \lor \neg\varphi$}
\RightLabel{$\neg$Elim}
\BinaryInfC{$\bot$}
\LeftLabel{2}
\RightLabel{$\neg$Intro}
\UnaryInfC{$\neg\varphi$}
\AxiomC{$[\neg(\varphi \lor \neg\varphi)]^1$}
\AxiomC{$[\neg\varphi]^3$}
\RightLabel{$\lor$Intro}
\UnaryInfC{$\varphi \lor \neg\varphi$}
\RightLabel{$\neg$Elim}
\BinaryInfC{$\bot$}
\LeftLabel{3}
\RightLabel{$\bot_C$}
\UnaryInfC{φ}
\RightLabel{$\neg$Elim}
\BinaryInfC{$\bot$}
\LeftLabel{1}
\RightLabel{$\bot_C$}
\UnaryInfC{$\varphi \lor \neg\varphi$}
\end{prooftree}
$$

Problem 7.1. Give derivations that show the following:

1. $\varphi \land (\psi \land \chi) \vdash (\varphi \land \psi) \land \chi$.
2. $\varphi \lor (\psi \lor \chi) \vdash (\varphi \lor \psi) \lor \chi$.
3. $\varphi \to (\psi \to \chi) \vdash \psi \to (\varphi \to \chi)$.
4. $\varphi \vdash \neg\neg\varphi$.

Problem 7.2. Give derivations that show the following:

1. $(\varphi \lor \psi) \to \chi \vdash \varphi \to \chi$.
2. $(\varphi \to \chi) \land (\psi \to \chi) \vdash (\varphi \lor \psi) \to \chi$.
3. $\vdash \neg(\varphi \land \neg\varphi)$.
4. $\psi \to \varphi \vdash \neg\varphi \to \neg\psi$.
5. $\vdash (\varphi \to \neg\varphi) \to \neg\varphi$.
6. $\vdash \neg(\varphi \to \psi) \to \neg\psi$.

7. $\varphi \to \chi \vdash \neg(\varphi \land \neg\chi)$.

8. $\varphi \land \neg\chi \vdash \neg(\varphi \to \chi)$.

9. $\varphi \lor \psi, \neg\psi \vdash \varphi$.

10. $\neg\varphi \lor \neg\psi \vdash \neg(\varphi \land \psi)$.

11. $\vdash (\neg\varphi \land \neg\psi) \to \neg(\varphi \lor \psi)$.

12. $\vdash \neg(\varphi \lor \psi) \to (\neg\varphi \land \neg\psi)$.

Problem 7.3. Give derivations that show the following:

1. $\neg(\varphi \to \psi) \vdash \varphi$.

2. $\neg(\varphi \land \psi) \vdash \neg\varphi \lor \neg\psi$.

3. $\varphi \to \psi \vdash \neg\varphi \lor \psi$.

4. $\vdash \neg\neg\varphi \to \varphi$.

5. $\varphi \to \psi, \neg\varphi \to \psi \vdash \psi$.

6. $(\varphi \land \psi) \to \chi \vdash (\varphi \to \chi) \lor (\psi \to \chi)$.

7. $(\varphi \to \psi) \to \varphi \vdash \varphi$.

8. $\vdash (\varphi \to \psi) \lor (\psi \to \chi)$.

(These all require the $\bot_C$ rule.)

7.6 Derivations with Quantifiers

Example 7.7. When dealing with quantifiers, we have to make sure not to violate the eigenvariable condition, and sometimes this requires us to play around with the order of carrying out certain inferences. In general, it helps to try and take care of rules subject to the eigenvariable condition first (they will be lower down in the finished proof). fol:ntd:prq:sec

Let's see how we'd give a derivation of the formula $\exists x\, \neg\varphi(x) \to \neg\forall x\, \varphi(x)$. Starting as usual, we write

$$\dfrac{}{\exists x\, \neg\varphi(x) \to \neg\forall x\, \varphi(x)}$$

We start by writing down what it would take to justify that last step using the $\to$Intro rule.

$$
{}_{1}\,\dfrac{\begin{array}{c}[\exists x\,\neg\varphi(x)]^1\\ \vdots\\ \neg\forall x\,\varphi(x)\end{array}}{\exists x\,\neg\varphi(x)\to\neg\forall x\,\varphi(x)}\;{\to}\mathrm{Intro}
$$

Since there is no obvious rule to apply to $\neg\forall x\,\varphi(x)$, we will proceed by setting up the derivation so we can use the $\exists$Elim rule. Here we must pay attention to the eigenvariable condition, and choose a constant that does not appear in $\exists x\,\varphi(x)$ or any assumptions that it depends on. (Since no constant symbols appear, however, any choice will do fine.)

$$
{}_{1}\,\dfrac{{}_{2}\,\dfrac{[\exists x\,\neg\varphi(x)]^1 \qquad \begin{array}{c}[\neg\varphi(a)]^2\\ \vdots\\ \neg\forall x\,\varphi(x)\end{array}}{\neg\forall x\,\varphi(x)}\;\exists\mathrm{Elim}}{\exists x\,\neg\varphi(x)\to\neg\forall x\,\varphi(x)}\;{\to}\mathrm{Intro}
$$

In order to derive $\neg\forall x\,\varphi(x)$, we will attempt to use the $\neg$Intro rule: this requires that we derive a contradiction, possibly using $\forall x\,\varphi(x)$ as an additional assumption. Of course, this contradiction may involve the assumption $\neg\varphi(a)$ which will be discharged by the $\exists$Elim inference. We can set it up as follows:

$$
{}_{1}\,\dfrac{{}_{2}\,\dfrac{[\exists x\,\neg\varphi(x)]^1 \qquad {}_{3}\,\dfrac{\begin{array}{c}[\neg\varphi(a)]^2, [\forall x\,\varphi(x)]^3\\ \vdots\\ \bot\end{array}}{\neg\forall x\,\varphi(x)}\;\neg\mathrm{Intro}}{\neg\forall x\,\varphi(x)}\;\exists\mathrm{Elim}}{\exists x\,\neg\varphi(x)\to\neg\forall x\,\varphi(x)}\;{\to}\mathrm{Intro}
$$

It looks like we are close to getting a contradiction. The easiest rule to apply is the $\forall$Elim, which has no eigenvariable conditions. Since we can use any term we want to replace the universally quantified x, it makes the most sense to continue using a so we can reach a contradiction.

$$
{}_{1}\,\dfrac{{}_{2}\,\dfrac{[\exists x\,\neg\varphi(x)]^1 \qquad {}_{3}\,\dfrac{\dfrac{[\neg\varphi(a)]^2 \qquad \dfrac{[\forall x\,\varphi(x)]^3}{\varphi(a)}\;\forall\mathrm{Elim}}{\bot}\;\neg\mathrm{Elim}}{\neg\forall x\,\varphi(x)}\;\neg\mathrm{Intro}}{\neg\forall x\,\varphi(x)}\;\exists\mathrm{Elim}}{\exists x\,\neg\varphi(x)\to\neg\forall x\,\varphi(x)}\;{\to}\mathrm{Intro}
$$

It is important, especially when dealing with quantifiers, to double check at this point that the eigenvariable condition has not been violated. Since the only rule we applied that is subject to the eigenvariable condition was $\exists$Elim, and the eigenvariable a does not occur in any assumptions it depends on, this is a correct derivation.

Example 7.8. Sometimes we may derive a formula from other formulas. In these cases, we may have undischarged assumptions. It is important to keep track of our assumptions as well as the end goal.

Let's see how we'd give a derivation of the formula $\exists x\, \chi(x, b)$ from the assumptions $\exists x\, (\varphi(x) \land \psi(x))$ and $\forall x\, (\psi(x) \to \chi(x, b))$. Starting as usual, we write the conclusion at the bottom.

$$\frac{}{\exists x\, \chi(x, b)}$$

We have two premises to work with. To use the first, i.e., try to find a derivation of $\exists x\, \chi(x, b)$ from $\exists x\, (\varphi(x) \land \psi(x))$ we would use the $\exists$Elim rule. Since it has an eigenvariable condition, we will apply that rule first. We get the following:

$$_1\ \dfrac{\exists x\, (\varphi(x) \land \psi(x)) \qquad \begin{array}{c}[\varphi(a) \land \psi(a)]^1 \\ \vdots \\ \exists x\, \chi(x, b)\end{array}}{\exists x\, \chi(x, b)}\ \exists\mathrm{Elim}$$

The two assumptions we are working with share ψ. It may be useful at this point to apply $\land$Elim to separate out $\psi(a)$.

$$_1\ \dfrac{\exists x\, (\varphi(x) \land \psi(x)) \qquad \begin{array}{c}\dfrac{[\varphi(a) \land \psi(a)]^1}{\psi(a)}\ \land\mathrm{Elim} \\ \vdots \\ \exists x\, \chi(x, b)\end{array}}{\exists x\, \chi(x, b)}\ \exists\mathrm{Elim}$$

The second assumption we have to work with is $\forall x\, (\psi(x) \to \chi(x, b))$. Since there is no eigenvariable condition we can instantiate x with the constant symbol a using $\forall$Elim to get $\psi(a) \to \chi(a, b)$. We now have both $\psi(a) \to \chi(a, b)$ and $\psi(a)$. Our next move should be a straightforward application of the $\to$Elim rule.

$$
{}^{1}\,\dfrac{\exists x\,(\varphi(x)\land\psi(x)) \qquad \begin{array}{c} \dfrac{\dfrac{\forall x\,(\psi(x)\to\chi(x,b))}{\psi(a)\to\chi(a,b)}\,\forall\text{Elim} \qquad \dfrac{[\varphi(a)\land\psi(a)]^1}{\psi(a)}\,\land\text{Elim}}{\chi(a,b)}\,{\to}\text{Elim} \\ \vdots \\ \exists x\,\chi(x,b) \end{array}}{\exists x\,\chi(x,b)}\,\exists\text{Elim}
$$

We are so close! One application of $\exists$Intro and we have reached our goal.

$$
{}^{1}\,\dfrac{\exists x\,(\varphi(x)\land\psi(x)) \qquad \dfrac{\dfrac{\dfrac{\forall x\,(\psi(x)\to\chi(x,b))}{\psi(a)\to\chi(a,b)}\,\forall\text{Elim} \qquad \dfrac{[\varphi(a)\land\psi(a)]^1}{\psi(a)}\,\land\text{Elim}}{\chi(a,b)}\,{\to}\text{Elim}}{\exists x\,\chi(x,b)}\,\exists\text{Intro}}{\exists x\,\chi(x,b)}\,\exists\text{Elim}
$$

Since we ensured at each step that the eigenvariable conditions were not violated, we can be confident that this is a correct derivation.

Example 7.9. Give a derivation of the formula $\lnot\forall x\,\varphi(x)$ from the assumptions $\forall x\,\varphi(x)\to\exists y\,\psi(y)$ and $\lnot\exists y\,\psi(y)$. Starting as usual, we write the target formula at the bottom.

$$
\dfrac{}{\lnot\forall x\,\varphi(x)}
$$

The last line of the derivation is a negation, so let's try using $\lnot$Intro. This will require that we figure out how to derive a contradiction.

$$
{}^{1}\,\dfrac{\begin{array}{c}[\forall x\,\varphi(x)]^1 \\ \vdots \\ \bot\end{array}}{\lnot\forall x\,\varphi(x)}\,\lnot\text{Intro}
$$

So far so good. We can use $\forall$Elim but it's not obvious if that will help us get to our goal. Instead, let's use one of our assumptions. $\forall x\,\varphi(x)\to\exists y\,\psi(y)$ together with $\forall x\,\varphi(x)$ will allow us to use the $\to$Elim rule.

$$
{}^{1}\,\dfrac{\begin{array}{c}\dfrac{\forall x\,\varphi(x)\to\exists y\,\psi(y) \qquad [\forall x\,\varphi(x)]^1}{\exists y\,\psi(y)}\,{\to}\text{Elim} \\ \vdots \\ \bot\end{array}}{\lnot\forall x\,\varphi(x)}\,\lnot\text{Intro}
$$

We now have one final assumption to work with, and it looks like this will help us reach a contradiction by using ¬Elim.

$$
{}^{1}\,\dfrac{\dfrac{\neg\exists y\,\psi(y) \qquad \dfrac{\forall x\,\varphi(x)\to\exists y\,\psi(y) \qquad [\forall x\,\varphi(x)]^1}{\exists y\,\psi(y)}\ {\to}\mathrm{Elim}}{\bot}\ \neg\mathrm{Elim}}{\neg\forall x\,\varphi(x)}\ \neg\mathrm{Intro}
$$

Problem 7.4. Give derivations that show the following:

1. $\vdash (\forall x\,\varphi(x) \land \forall y\,\psi(y)) \to \forall z\,(\varphi(z) \land \psi(z))$.
2. $\vdash (\exists x\,\varphi(x) \lor \exists y\,\psi(y)) \to \exists z\,(\varphi(z) \lor \psi(z))$.
3. $\forall x\,(\varphi(x) \to \psi) \vdash \exists y\,\varphi(y) \to \psi$.
4. $\forall x\,\neg\varphi(x) \vdash \neg\exists x\,\varphi(x)$.
5. $\vdash \neg\exists x\,\varphi(x) \to \forall x\,\neg\varphi(x)$.
6. $\vdash \neg\exists x\,\forall y\,((\varphi(x,y) \to \neg\varphi(y,y)) \land (\neg\varphi(y,y) \to \varphi(x,y)))$.

Problem 7.5. Give derivations that show the following:

1. $\vdash \neg\forall x\,\varphi(x) \to \exists x\,\neg\varphi(x)$.
2. $(\forall x\,\varphi(x) \to \psi) \vdash \exists y\,(\varphi(y) \to \psi)$.
3. $\vdash \exists x\,(\varphi(x) \to \forall y\,\varphi(y))$.

(These all require the $\bot_C$ rule.)

7.7 Proof-Theoretic Notions

fol:ntd:ptn:sec

> This section collects the definitions the provability relation and consistency for natural deduction.

explanation Just as we've defined a number of important semantic notions (validity, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the derivability or non-derivability of certain sentences from others. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*.

Definition 7.10 (Theorems). A sentence φ is a *theorem* if there is a derivation of φ in natural deduction in which all assumptions are discharged. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definition 7.11 (Derivability). A sentence φ is *derivable from* a set of sentences Γ, $\Gamma \vdash \varphi$, if there is a derivation with conclusion φ and in which every assumption is either discharged or is in Γ. If φ is not derivable from Γ we write $\Gamma \nvdash \varphi$.

Definition 7.12 (Consistency). A set of sentences Γ is *inconsistent* iff $\Gamma \vdash \bot$. If Γ is not inconsistent, i.e., if $\Gamma \nvdash \bot$, we say it is *consistent*.

fol:ntd:ptn: prop:reflexivity

Proposition 7.13 (Reflexivity). *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.*

Proof. The assumption φ by itself is a derivation of φ where every undischarged assumption (i.e., φ) is in Γ. □

fol:ntd:ptn: prop:monotonicity

Proposition 7.14 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.*

Proof. Any derivation of φ from Γ is also a derivation of φ from Δ. □

fol:ntd:ptn: prop:transitivity

Proposition 7.15 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.*

Proof. If $\Gamma \vdash \varphi$, there is a derivation δ_0 of φ with all undischarged assumptions in Γ. If $\{\varphi\} \cup \Delta \vdash \psi$, then there is a derivation δ_1 of ψ with all undischarged assumptions in $\{\varphi\} \cup \Delta$. Now consider:

$$\dfrac{{}^{1}\,\dfrac{\begin{array}{c}\Delta, [\varphi]^1 \\ \vdots\;\delta_1 \\ \psi\end{array}}{\varphi \to \psi}\,{\to}\mathrm{Intro} \qquad \begin{array}{c}\Gamma \\ \vdots\;\delta_0 \\ \varphi\end{array}}{\psi}\,{\to}\mathrm{Elim}$$

The undischarged assumptions are now all among $\Gamma \cup \Delta$, so this shows $\Gamma \cup \Delta \vdash \psi$. □

When $\Gamma = \{\varphi_1, \varphi_2, \dots, \varphi_k\}$ is a finite set we may use the simplified notation $\varphi_1, \varphi_2, \dots, \varphi_k \vdash \psi$ for $\Gamma \vdash \psi$, in particular $\varphi \vdash \psi$ means that $\{\varphi\} \vdash \psi$.

Note that if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \dots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i, then $\Gamma \vdash \psi$.

fol:ntd:ptn: prop:incons

Proposition 7.16. *The following are equivalent.*

1. *Γ is inconsistent.*
2. *$\Gamma \vdash \varphi$ for every sentence φ.*
3. *$\Gamma \vdash \varphi$ and $\Gamma \vdash \neg\varphi$ for some sentence φ.*

Proof. Exercise. □

Problem 7.6. Prove Proposition 7.16

Proposition 7.17 (Compactness). *fol:ntd:ptn: prop:proves-compact*

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Proof. 1. If $\Gamma \vdash \varphi$, then there is a derivation δ of φ from Γ. Let Γ_0 be the set of undischarged assumptions of δ. Since any derivation is finite, Γ_0 can only contain finitely many sentences. So, δ is a derivation of φ from a finite $\Gamma_0 \subseteq \Gamma$.

2. This is the contrapositive of (1) for the special case $\varphi \equiv \bot$. □

7.8 Derivability and Consistency

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem. *fol:ntd:prv: sec*

Proposition 7.18. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.* *fol:ntd:prv: prop:provability-contr*

Proof. Let the derivation of φ from Γ be δ_1 and the derivation of $\bot$ from $\Gamma \cup \{\varphi\}$ be δ_2. We can then derive:

$$\dfrac{{}^{1}\,\dfrac{\begin{matrix}\Gamma, [\varphi]^1\\ \vdots\ \delta_2\\ \bot\end{matrix}}{\neg\varphi}\,\neg\text{Intro} \qquad \begin{matrix}\Gamma\\ \vdots\ \delta_1\\ \varphi\end{matrix}}{\bot}\,\neg\text{Elim}$$

In the new derivation, the assumption φ is discharged, so it is a derivation from Γ. □

Proposition 7.19. *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.* *fol:ntd:prv: prop:prov-incons*

Proof. First suppose $\Gamma \vdash \varphi$, i.e., there is a derivation δ_0 of φ from undischarged assumptions Γ. We obtain a derivation of $\bot$ from $\Gamma \cup \{\neg\varphi\}$ as follows:

$$\dfrac{\neg\varphi \qquad \begin{matrix}\Gamma\\ \vdots\ \delta_0\\ \varphi\end{matrix}}{\bot}\,\neg\text{Elim}$$

Now assume $\Gamma \cup \{\neg\varphi\}$ is inconsistent, and let δ_1 be the corresponding derivation of $\bot$ from undischarged assumptions in $\Gamma \cup \{\neg\varphi\}$. We obtain a derivation of φ from Γ alone by using $\bot_C$:

$$\dfrac{\begin{matrix}\Gamma, [\neg\varphi]^1 \\ \vdots\ \delta_1 \\ \bot\end{matrix}}{\varphi}\ \bot_C\ {}^{1}$$

□

Problem 7.7. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

fol:ntd:prv: prop:explicit-inc

Proposition 7.20. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.*

Proof. Suppose $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$. Then there is a derivation δ of φ from Γ. Consider this simple application of the $\neg$Elim rule:

$$\dfrac{\neg\varphi \qquad \begin{matrix}\Gamma \\ \vdots\ \delta \\ \varphi\end{matrix}}{\bot}\ \neg\mathrm{Elim}$$

Since $\neg\varphi \in \Gamma$, all undischarged assumptions are in Γ, this shows that $\Gamma \vdash \bot$. □

fol:ntd:prv: prop:provability-exhaustive

Proposition 7.21. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

Proof. There are derivations δ_1 and δ_2 of $\bot$ from $\Gamma \cup \{\varphi\}$ and $\bot$ from $\Gamma \cup \{\neg\varphi\}$, respectively. We can then derive

$$\dfrac{{}^{2}\dfrac{\begin{matrix}\Gamma, [\neg\varphi]^2 \\ \vdots\ \delta_2 \\ \bot\end{matrix}}{\neg\neg\varphi}\ \neg\mathrm{Intro} \qquad {}^{1}\dfrac{\begin{matrix}\Gamma, [\varphi]^1 \\ \vdots\ \delta_1 \\ \bot\end{matrix}}{\neg\varphi}\ \neg\mathrm{Intro}}{\bot}\ \neg\mathrm{Elim}$$

Since the assumptions φ and $\neg\varphi$ are discharged, this is a derivation of $\bot$ from Γ alone. Hence Γ is inconsistent. □

7.9 Derivability and the Propositional Connectives

explanation We establish that the derivability relation $\vdash$ of natural deduction is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \land \psi \vdash \varphi$ and $\varphi, \varphi \to \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem. fol:ntd:ppr:sec

Proposition 7.22. fol:ntd:ppr:prop:provability-land

1. *Both* $\varphi \land \psi \vdash \varphi$ *and* $\varphi \land \psi \vdash \psi$ fol:ntd:ppr:prop:provability-land-left
2. $\varphi, \psi \vdash \varphi \land \psi$. fol:ntd:ppr:prop:provability-land-right

Proof. 1. We can derive both

$$\dfrac{\varphi \land \psi}{\varphi}\land\text{Elim} \qquad \dfrac{\varphi \land \psi}{\psi}\land\text{Elim}$$

2. We can derive:

$$\dfrac{\varphi \qquad \psi}{\varphi \land \psi}\land\text{Intro}$$

□

Proposition 7.23. fol:ntd:ppr:prop:provability-lor

1. $\varphi \lor \psi, \neg\varphi, \neg\psi$ *is inconsistent.*
2. *Both* $\varphi \vdash \varphi \lor \psi$ *and* $\psi \vdash \varphi \lor \psi$.

Proof. 1. Consider the following derivation:

$${}_1\,\dfrac{\varphi \lor \psi \qquad \dfrac{\neg\varphi \qquad [\varphi]^1}{\bot}\neg\text{Elim} \qquad \dfrac{\neg\psi \qquad [\psi]^1}{\bot}\neg\text{Elim}}{\bot}\lor\text{Elim}$$

This is a derivation of $\bot$ from undischarged assumptions $\varphi \lor \psi$, $\neg\varphi$, and $\neg\psi$.

2. We can derive both

$$\dfrac{\varphi}{\varphi \lor \psi}\lor\text{Intro} \qquad \dfrac{\psi}{\varphi \lor \psi}\lor\text{Intro}$$

□

Proposition 7.24. fol:ntd:ppr:prop:provability-lif

1. $\varphi, \varphi \to \psi \vdash \psi$. fol:ntd:ppr:prop:provability-lif-left
2. *Both* $\neg\varphi \vdash \varphi \to \psi$ *and* $\psi \vdash \varphi \to \psi$. fol:ntd:ppr:prop:provability-lif-right

Proof. 1. We can derive:

$$\frac{\varphi \to \psi \qquad \varphi}{\psi} \to\text{Elim}$$

2. This is shown by the following two derivations:

$$\dfrac{\dfrac{\dfrac{\neg\varphi \qquad [\varphi]^1}{\bot}\neg\text{Elim}}{\psi}\bot_I}{\varphi \to \psi}{}^{1}\to\text{Intro} \qquad\qquad \frac{\psi}{\varphi \to \psi}\to\text{Intro}$$

Note that →Intro may, but does not have to, discharge the assumption φ. □

7.10 Derivability and the Quantifiers

fol:ntd:qpr:sec The completeness theorem also requires that the natural deduction rules yield the facts about $\vdash$ established in this section. explanation

fol:ntd:qpr:thm:strong-generalization **Theorem 7.25.** *If c is a constant not occurring in Γ or $\varphi(x)$ and $\Gamma \vdash \varphi(c)$, then $\Gamma \vdash \forall x\, \varphi(x)$.*

Proof. Let δ be a derivation of $\varphi(c)$ from Γ. By adding a ∀Intro inference, we obtain a derivation of $\forall x\, \varphi(x)$. Since c does not occur in Γ or $\varphi(x)$, the eigenvariable condition is satisfied. □

fol:ntd:qpr:prop:provability-quantifiers **Proposition 7.26.**

1. *$\varphi(t) \vdash \exists x\, \varphi(x)$.*
2. *$\forall x\, \varphi(x) \vdash \varphi(t)$.*

Proof. 1. The following is a derivation of $\exists x\, \varphi(x)$ from $\varphi(t)$:

$$\frac{\varphi(t)}{\exists x\, \varphi(x)}\exists\text{Intro}$$

2. The following is a derivation of $\varphi(t)$ from $\forall x\, \varphi(x)$:

$$\frac{\forall x\, \varphi(x)}{\varphi(t)}\forall\text{Elim}$$

□

7.11 Soundness

explanation A derivation system, such as natural deduction, is *sound* if it cannot derive things that do not actually follow. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that fol:ntd:sou:sec

1. every derivable sentence is valid;
2. if a sentence is derivable from some others, it is also a consequence of them;
3. if a set of sentences is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Theorem 7.27 (Soundness). *If φ is derivable from the undischarged assumptions Γ, then $\Gamma \vDash \varphi$.* fol:ntd:sou:thm:soundness

Proof. Let δ be a derivation of φ. We proceed by induction on the number of inferences in δ.

For the induction basis we show the claim if the number of inferences is 0. In this case, δ consists only of a single sentence φ, i.e., an assumption. That assumption is undischarged, since assumptions can only be discharged by inferences, and there are no inferences. So, any structure $\mathfrak{M}$ that satisfies all of the undischarged assumptions of the proof also satisfies φ.

Now for the inductive step. Suppose that δ contains n inferences. The premise(s) of the lowermost inference are derived using sub-derivations, each of which contains fewer than n inferences. We assume the induction hypothesis: The premises of the lowermost inference follow from the undischarged assumptions of the sub-derivations ending in those premises. We have to show that the conclusion φ follows from the undischarged assumptions of the entire proof.

We distinguish cases according to the type of the lowermost inference. First, we consider the possible inferences with only one premise.

1. Suppose that the last inference is $\neg$Intro: The derivation has the form

$$\dfrac{\begin{array}{c}\Gamma, [\varphi]^n \\ \vdots\ \delta_1 \\ \bot\end{array}}{\neg\varphi}\ {}^{n}\ \neg\text{Intro}$$

By inductive hypothesis, $\bot$ follows from the undischarged assumptions $\Gamma \cup \{\varphi\}$ of δ_1. Consider a structure $\mathfrak{M}$. We need to show that, if $\mathfrak{M} \vDash \Gamma$, then $\mathfrak{M} \vDash \neg\varphi$. Suppose for reductio that $\mathfrak{M} \vDash \Gamma$, but $\mathfrak{M} \nvDash \neg\varphi$, i.e., $\mathfrak{M} \vDash \varphi$. This would mean that $\mathfrak{M} \vDash \Gamma \cup \{\varphi\}$. This is contrary to our inductive hypothesis. So, $\mathfrak{M} \vDash \neg\varphi$.

2. The last inference is $\wedge$Elim: There are two variants: φ or ψ may be inferred from the premise $\varphi \wedge \psi$. Consider the first case. The derivation δ looks like this:

$$\cfrac{\begin{matrix}\Gamma \\ \vdots\, \delta_1 \\ \varphi \wedge \psi\end{matrix}}{\varphi}\,\wedge\text{Elim}$$

By inductive hypothesis, $\varphi \wedge \psi$ follows from the undischarged assumptions Γ of δ_1. Consider a structure $\mathfrak{M}$. We need to show that, if $\mathfrak{M} \vDash \Gamma$, then $\mathfrak{M} \vDash \varphi$. Suppose $\mathfrak{M} \vDash \Gamma$. By our inductive hypothesis ($\Gamma \vDash \varphi \wedge \psi$), we know that $\mathfrak{M} \vDash \varphi \wedge \psi$. By definition, $\mathfrak{M} \vDash \varphi \wedge \psi$ iff $\mathfrak{M} \vDash \varphi$ and $\mathfrak{M} \vDash \psi$. (The case where ψ is inferred from $\varphi \wedge \psi$ is handled similarly.)

3. The last inference is $\vee$Intro: There are two variants: $\varphi \vee \psi$ may be inferred from the premise φ or the premise ψ. Consider the first case. The derivation has the form

$$\cfrac{\begin{matrix}\Gamma \\ \vdots\, \delta_1 \\ \varphi\end{matrix}}{\varphi \vee \psi}\,\vee\text{Intro}$$

By inductive hypothesis, φ follows from the undischarged assumptions Γ of δ_1. Consider a structure $\mathfrak{M}$. We need to show that, if $\mathfrak{M} \vDash \Gamma$, then $\mathfrak{M} \vDash \varphi \vee \psi$. Suppose $\mathfrak{M} \vDash \Gamma$; then $\mathfrak{M} \vDash \varphi$ since $\Gamma \vDash \varphi$ (the inductive hypothesis). So it must also be the case that $\mathfrak{M} \vDash \varphi \vee \psi$. (The case where $\varphi \vee \psi$ is inferred from ψ is handled similarly.)

4. The last inference is $\to$Intro: $\varphi \to \psi$ is inferred from a subproof with assumption φ and conclusion ψ, i.e.,

$$n\,\cfrac{\begin{matrix}\Gamma, [\varphi]^n \\ \vdots\, \delta_1 \\ \psi\end{matrix}}{\varphi \to \psi}\,\to\text{Intro}$$

By inductive hypothesis, ψ follows from the undischarged assumptions of δ_1, i.e., $\Gamma \cup \{\varphi\} \vDash \psi$. Consider a structure $\mathfrak{M}$. The undischarged assumptions of δ are just Γ, since φ is discharged at the last inference. So we need to show that $\Gamma \vDash \varphi \rightarrow \psi$. For reductio, suppose that for some structure $\mathfrak{M}$, $\mathfrak{M} \vDash \Gamma$ but $\mathfrak{M} \nvDash \varphi \rightarrow \psi$. So, $\mathfrak{M} \vDash \varphi$ and $\mathfrak{M} \nvDash \psi$. But by hypothesis, ψ is a consequence of $\Gamma \cup \{\varphi\}$, i.e., $\mathfrak{M} \vDash \psi$, which is a contradiction. So, $\Gamma \vDash \varphi \rightarrow \psi$.

5. The last inference is $\bot_I$: Here, δ ends in

$$\dfrac{\begin{matrix}\Gamma \\ \vdots\, \delta_1 \\ \bot\end{matrix}}{\varphi}\;\bot_I$$

By induction hypothesis, $\Gamma \vDash \bot$. We have to show that $\Gamma \vDash \varphi$. Suppose not; then for some $\mathfrak{M}$ we have $\mathfrak{M} \vDash \Gamma$ and $\mathfrak{M} \nvDash \varphi$. But we always have $\mathfrak{M} \nvDash \bot$, so this would mean that $\Gamma \nvDash \bot$, contrary to the induction hypothesis.

6. The last inference is $\bot_C$: Exercise.

7. The last inference is $\forall$Intro: Then δ has the form

$$\dfrac{\begin{matrix}\Gamma \\ \vdots\, \delta_1 \\ \varphi(a)\end{matrix}}{\forall x\, \varphi(x)}\;\forall\text{Intro}$$

The premise $\varphi(a)$ is a consequence of the undischarged assumptions Γ by induction hypothesis. Consider some structure, $\mathfrak{M}$, such that $\mathfrak{M} \vDash \Gamma$. We need to show that $\mathfrak{M} \vDash \forall x\, \varphi(x)$. Since $\forall x\, \varphi(x)$ is a sentence, this means we have to show that for every variable assignment s, $\mathfrak{M}, s \vDash \varphi(x)$ (Proposition 3.18). Since Γ consists entirely of sentences, $\mathfrak{M}, s \vDash \psi$ for all $\psi \in \Gamma$ by Definition 3.11. Let $\mathfrak{M}'$ be like $\mathfrak{M}$ except that $a^{\mathfrak{M}'} = s(x)$. Since a does not occur in Γ, $\mathfrak{M}' \vDash \Gamma$ by Corollary 3.20. Since $\Gamma \vDash \varphi(a)$, $\mathfrak{M}' \vDash \varphi(a)$. Since $\varphi(a)$ is a sentence, $\mathfrak{M}', s \vDash \varphi(a)$ by Proposition 3.17. $\mathfrak{M}', s \vDash \varphi(x)$ iff $\mathfrak{M}' \vDash \varphi(a)$ by Proposition 3.22 (recall that $\varphi(a)$ is just $\varphi(x)[a/x]$). So, $\mathfrak{M}', s \vDash \varphi(x)$. Since a does not occur in $\varphi(x)$, by Proposition 3.19, $\mathfrak{M}, s \vDash \varphi(x)$. But s was an arbitrary variable assignment, so $\mathfrak{M} \vDash \forall x\, \varphi(x)$.

8. The last inference is $\exists$Intro: Exercise.

9. The last inference is $\forall$Elim: Exercise.

Now let's consider the possible inferences with several premises: $\vee$Elim, $\wedge$Intro, $\rightarrow$Elim, and $\exists$Elim.

1. The last inference is $\wedge$Intro. $\varphi \wedge \psi$ is inferred from the premises φ and ψ and δ has the form

$$\dfrac{\begin{array}{c}\Gamma_1\\ \vdots\,\delta_1\\ \varphi\end{array} \qquad \begin{array}{c}\Gamma_2\\ \vdots\,\delta_2\\ \psi\end{array}}{\varphi \wedge \psi}\wedge\text{Intro}$$

   By induction hypothesis, φ follows from the undischarged assumptions Γ_1 of δ_1 and ψ follows from the undischarged assumptions Γ_2 of δ_2. The undischarged assumptions of δ are $\Gamma_1 \cup \Gamma_2$, so we have to show that $\Gamma_1 \cup \Gamma_2 \vDash \varphi \wedge \psi$. Consider a structure $\mathfrak{M}$ with $\mathfrak{M} \vDash \Gamma_1 \cup \Gamma_2$. Since $\mathfrak{M} \vDash \Gamma_1$, it must be the case that $\mathfrak{M} \vDash \varphi$ as $\Gamma_1 \vDash \varphi$, and since $\mathfrak{M} \vDash \Gamma_2$, $\mathfrak{M} \vDash \psi$ since $\Gamma_2 \vDash \psi$. Together, $\mathfrak{M} \vDash \varphi \wedge \psi$.

2. The last inference is $\vee$Elim: Exercise.

3. The last inference is $\rightarrow$Elim. ψ is inferred from the premises $\varphi \rightarrow \psi$ and φ. The derivation δ looks like this:

$$\dfrac{\begin{array}{c}\Gamma_1\\ \vdots\,\delta_1\\ \varphi \rightarrow \psi\end{array} \qquad \begin{array}{c}\Gamma_2\\ \vdots\,\delta_2\\ \varphi\end{array}}{\psi}\rightarrow\text{Elim}$$

   By induction hypothesis, $\varphi \rightarrow \psi$ follows from the undischarged assumptions Γ_1 of δ_1 and φ follows from the undischarged assumptions Γ_2 of δ_2. Consider a structure $\mathfrak{M}$. We need to show that, if $\mathfrak{M} \vDash \Gamma_1 \cup \Gamma_2$, then $\mathfrak{M} \vDash \psi$. Suppose $\mathfrak{M} \vDash \Gamma_1 \cup \Gamma_2$. Since $\Gamma_1 \vDash \varphi \rightarrow \psi$, $\mathfrak{M} \vDash \varphi \rightarrow \psi$. Since $\Gamma_2 \vDash \varphi$, we have $\mathfrak{M} \vDash \varphi$. This means that $\mathfrak{M} \vDash \psi$ (For if $\mathfrak{M} \nvDash \psi$, since $\mathfrak{M} \vDash \varphi$, we'd have $\mathfrak{M} \nvDash \varphi \rightarrow \psi$, contradicting $\mathfrak{M} \vDash \varphi \rightarrow \psi$).

4. The last inference is $\neg$Elim: Exercise.

5. The last inference is $\exists$Elim: Exercise. $\square$

Problem 7.8. Complete the proof of Theorem 7.27.

fol:ntd:sou: cor:weak-soundness **Corollary 7.28.** *If $\vdash \varphi$, then φ is valid.*

fol:ntd:sou: cor:consistency-soundness **Corollary 7.29.** *If Γ is satisfiable, then it is consistent.*

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then $\Gamma \vdash \bot$, i.e., there is a derivation of $\bot$ from undischarged assumptions in Γ. By Theorem 7.27, any structure $\mathfrak{M}$ that satisfies Γ must satisfy $\bot$. Since $\mathfrak{M} \nvDash \bot$ for every structure $\mathfrak{M}$, no $\mathfrak{M}$ can satisfy Γ, i.e., Γ is not satisfiable. □

7.12 Derivations with Identity predicate

Derivations with identity predicate require additional inference rules. fol:ntd:ide:sec

$$\frac{}{t = t}\;{=}\mathrm{Intro} \qquad\qquad \frac{t_1 = t_2 \qquad \varphi(t_1)}{\varphi(t_2)}\;{=}\mathrm{Elim} \qquad \frac{t_1 = t_2 \qquad \varphi(t_2)}{\varphi(t_1)}\;{=}\mathrm{Elim}$$

In the above rules, t, t_1, and t_2 are closed terms. The $=$Intro rule allows us to derive any identity statement of the form $t = t$ outright, from no assumptions.

Example 7.30. If s and t are closed terms, then $\varphi(s), s = t \vdash \varphi(t)$:

$$\frac{s = t \qquad \varphi(s)}{\varphi(t)}\;{=}\mathrm{Elim}$$

This may be familiar as the "principle of substitutability of identicals," or Leibniz' Law.

Problem 7.9. Prove that $=$ is both symmetric and transitive, i.e., give derivations of $\forall x\, \forall y\, (x = y \rightarrow y = x)$ and $\forall x\, \forall y\, \forall z((x = y \land y = z) \rightarrow x = z)$

Example 7.31. We derive the sentence

$$\forall x\, \forall y\, ((\varphi(x) \land \varphi(y)) \rightarrow x = y)$$

from the sentence

$$\exists x\, \forall y\, (\varphi(y) \rightarrow y = x)$$

We develop the derivation backwards:

$$\cfrac{\cfrac{{}^1\cfrac{\begin{array}{c}\exists x\, \forall y\, (\varphi(y) \rightarrow y = x) \quad [\varphi(a) \land \varphi(b)]^1 \\ \vdots \\ a = b\end{array}}{((\varphi(a) \land \varphi(b)) \rightarrow a = b)}\;{\rightarrow}\mathrm{Intro}}{\forall y\, ((\varphi(a) \land \varphi(y)) \rightarrow a = y)}\;\forall\mathrm{Intro}}{\forall x\, \forall y\, ((\varphi(x) \land \varphi(y)) \rightarrow x = y)}\;\forall\mathrm{Intro}$$

We'll now have to use the main assumption: since it is an existential formula, we use $\exists$Elim to derive the intermediary conclusion $a = b$.

$$
\begin{prooftree}
\AxiomC{$\exists x\, \forall y\, (\varphi(y) \to y = x)$}
\AxiomC{$[\forall y\, (\varphi(y) \to y = c)]^2$}
\noLine
\UnaryInfC{$[\varphi(a) \land \varphi(b)]^1$}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$a = b$}
\LeftLabel{2}
\RightLabel{$\exists$Elim}
\BinaryInfC{$a = b$}
\LeftLabel{1}
\RightLabel{$\to$Intro}
\UnaryInfC{$((\varphi(a) \land \varphi(b)) \to a = b)$}
\RightLabel{$\forall$Intro}
\UnaryInfC{$\forall y\, ((\varphi(a) \land \varphi(y)) \to a = y)$}
\RightLabel{$\forall$Intro}
\UnaryInfC{$\forall x\, \forall y\, ((\varphi(x) \land \varphi(y)) \to x = y)$}
\end{prooftree}
$$

The sub-derivation on the top right is completed by using its assumptions to show that $a = c$ and $b = c$. This requires two separate derivations. The derivation for $a = c$ is as follows:

$$
\begin{prooftree}
\AxiomC{$[\forall y\, (\varphi(y) \to y = c)]^2$}
\RightLabel{$\forall$Elim}
\UnaryInfC{$\varphi(a) \to a = c$}
\AxiomC{$[\varphi(a) \land \varphi(b)]^1$}
\RightLabel{$\land$Elim}
\UnaryInfC{$\varphi(a)$}
\RightLabel{$\to$Elim}
\BinaryInfC{$a = c$}
\end{prooftree}
$$

From $a = c$ and $b = c$ we derive $a = b$ by $=$Elim.

Problem 7.10. Give derivations of the following formulas:

1. $\forall x\, \forall y\, ((x = y \land \varphi(x)) \to \varphi(y))$
2. $\exists x\, \varphi(x) \land \forall y\, \forall z\, ((\varphi(y) \land \varphi(z)) \to y = z) \to \exists x\, (\varphi(x) \land \forall y\, (\varphi(y) \to y = x))$

7.13 Soundness with Identity predicate

fol:ntd:sid:sec

Proposition 7.32. *Natural deduction with rules for $=$ is sound.*

Proof. Any formula of the form $t = t$ is valid, since for every structure $\mathfrak{M}$, $\mathfrak{M} \vDash t = t$. (Note that we assume the term t to be closed, i.e., it contains no variables, so variable assignments are irrelevant).

Suppose the last inference in a derivation is $=$Elim, i.e., the derivation has the following form:

$$
\begin{prooftree}
\AxiomC{Γ_1}
\noLine
\UnaryInfC{$\vdots\ \delta_1$}
\noLine
\UnaryInfC{$t_1 = t_2$}
\AxiomC{Γ_2}
\noLine
\UnaryInfC{$\vdots\ \delta_2$}
\noLine
\UnaryInfC{$\varphi(t_1)$}
\RightLabel{$=$Elim}
\BinaryInfC{$\varphi(t_2)$}
\end{prooftree}
$$

The premises $t_1 = t_2$ and $\varphi(t_1)$ are derived from undischarged assumptions Γ_1 and Γ_2, respectively. We want to show that $\varphi(t_2)$ follows from $\Gamma_1 \cup \Gamma_2$. Consider a structure $\mathfrak{M}$ with $\mathfrak{M} \vDash \Gamma_1 \cup \Gamma_2$. By induction hypothesis, $\mathfrak{M} \vDash \varphi(t_1)$ and $\mathfrak{M} \vDash t_1 = t_2$. Therefore, $\mathrm{Val}^{\mathfrak{M}}(t_1) = \mathrm{Val}^{\mathfrak{M}}(t_2)$. Let s be any variable assignment, and $m = \mathrm{Val}^{\mathfrak{M}}(t_1) = \mathrm{Val}^{\mathfrak{M}}(t_2)$. By Proposition 3.22, $\mathfrak{M}, s \vDash \varphi(t_1)$ iff $\mathfrak{M}, s[m/x] \vDash \varphi(x)$ iff $\mathfrak{M}, s \vDash \varphi(t_2)$. Since $\mathfrak{M} \vDash \varphi(t_1)$, we have $\mathfrak{M} \vDash \varphi(t_2)$. $\square$

Chapter 8

Tableaux

This chapter presents a signed analytic tableaux system.

To include or exclude material relevant to natural deduction as a proof system, use the "prfTab" tag.

8.1 Rules and Tableaux

fol:tab:rul:sec A tableau is a systematic survey of the possible ways a sentence can be true or false in a structure. The building blocks of a tableau are signed formulas: sentences plus a truth value "sign," either $\mathbb{T}$ or $\mathbb{F}$. These signed formulas are arranged in a (downward growing) tree.

Definition 8.1. A *signed formula* is a pair consisting of a truth value and a sentence, i.e., either:

$$\mathbb{T}\,\varphi \text{ or } \mathbb{F}\,\varphi.$$

Intuitively, we might read $\mathbb{T}\,\varphi$ as "φ might be true" and $\mathbb{F}\,\varphi$ as "φ might be false" (in some structure).

Each signed formula in the tree is either an *assumption* (which are listed at the very top of the tree), or it is obtained from a signed formula above it by one of a number of rules of inference. There are two rules for each possible main operator of the preceding formula, one for the case where the sign is $\mathbb{T}$, and one for the case where the sign is $\mathbb{F}$. Some rules allow the tree to branch, and some only add signed formulas to the branch. A rule may be (and often must be) applied not to the immediately preceding signed formula, but to any signed formula in the branch from the root to the place the rule is applied.

A branch is *closed* when it contains both $\mathbb{T}\,\varphi$ and $\mathbb{F}\,\varphi$. A closed tableau is one where every branch is closed. Under the intuitive interpretation, any branch describes a joint possibility, but $\mathbb{T}\,\varphi$ and $\mathbb{F}\,\varphi$ are not jointly possible. In other words, if a branch is closed, the possibility it describes has been ruled out. In particular, that means that a closed tableau rules out all possibilities of simultaneously making every assumption of the form $\mathbb{T}\,\varphi$ true and every assumption of the form $\mathbb{F}\,\varphi$ false.

A closed tableau *for* φ is a closed tableau with root $\mathbb{F}\,\varphi$. If such a closed tableau exists, all possibilities for φ being false have been ruled out; i.e., φ must be true in every structure.

8.2 Propositional Rules

Rules for $\neg$

fol:tab:prl:sec

$$\frac{\mathbb{T}\,\neg\varphi}{\mathbb{F}\,\varphi}\;\neg\mathbb{T} \qquad\qquad \frac{\mathbb{F}\,\neg\varphi}{\mathbb{T}\,\varphi}\;\neg\mathbb{F}$$

Rules for $\land$

$$\frac{\mathbb{T}\,\varphi\land\psi}{\begin{array}{c}\mathbb{T}\,\varphi\\ \mathbb{T}\,\psi\end{array}}\;\land\mathbb{T} \qquad\qquad \frac{\mathbb{F}\,\varphi\land\psi}{\mathbb{F}\,\varphi \quad\mid\quad \mathbb{F}\,\psi}\;\land\mathbb{F}$$

Rules for $\lor$

$$\frac{\mathbb{T}\,\varphi\lor\psi}{\mathbb{T}\,\varphi \quad\mid\quad \mathbb{T}\,\psi}\;\lor\mathbb{T} \qquad\qquad \frac{\mathbb{F}\,\varphi\lor\psi}{\begin{array}{c}\mathbb{F}\,\varphi\\ \mathbb{F}\,\psi\end{array}}\;\lor\mathbb{F}$$

Rules for $\rightarrow$

$$\frac{\mathbb{T}\,\varphi\rightarrow\psi}{\mathbb{F}\,\varphi \quad\mid\quad \mathbb{T}\,\psi}\;\rightarrow\mathbb{T} \qquad\qquad \frac{\mathbb{F}\,\varphi\rightarrow\psi}{\begin{array}{c}\mathbb{T}\,\varphi\\ \mathbb{F}\,\psi\end{array}}\;\rightarrow\mathbb{F}$$

The Cut Rule

$$\frac{}{\mathbb{T}\,\varphi \quad\mid\quad \mathbb{F}\,\varphi}\;\text{Cut}$$

The Cut rule is not applied "to" a previous signed formula; rather, it allows every branch in a tableau to be split in two, one branch containing $\mathbb{T}\,\varphi$, the other $\mathbb{F}\,\varphi$. It is not necessary—any set of signed formulas with a closed tableau has one not using Cut—but it allows us to combine tableaux in a convenient way.

8.3 Quantifier Rules

fol:tab:qrl:sec

Rules for $\forall$

$$\frac{\mathbb{T}\,\forall x\,\varphi(x)}{\mathbb{T}\,\varphi(t)}\;\forall\mathbb{T} \qquad\qquad \frac{\mathbb{F}\,\forall x\,\varphi(x)}{\mathbb{F}\,\varphi(a)}\;\forall\mathbb{F}$$

In $\forall\mathbb{T}$, t is a closed term (i.e., one without variables). In $\forall\mathbb{F}$, a is a constant symbol which must not occur anywhere in the branch above $\forall\mathbb{F}$ rule. We call a the *eigenvariable* of the $\forall\mathbb{F}$ inference.[1]

Rules for $\exists$

$$\frac{\mathbb{T}\,\exists x\,\varphi(x)}{\mathbb{T}\,\varphi(a)}\;\exists\mathbb{T} \qquad\qquad \frac{\mathbb{F}\,\exists x\,\varphi(x)}{\mathbb{F}\,\varphi(t)}\;\exists\mathbb{F}$$

Again, t is a closed term, and a is a constant symbol which does not occur in the branch above the $\exists\mathbb{T}$ rule. We call a the *eigenvariable* of the $\exists\mathbb{T}$ inference.

The condition that an eigenvariable not occur in the branch above the $\forall\mathbb{F}$ or $\exists\mathbb{T}$ inference is called the *eigenvariable condition.*

Recall the convention that when φ is a formula with the variable x free, we indicate this by writing $\varphi(x)$. In the same context, $\varphi(t)$ then is short for $\varphi[t/x]$. So we could also write the $\exists\mathbb{F}$ rule as: explanation

$$\frac{\mathbb{F}\,\exists x\,\varphi}{\mathbb{F}\,\varphi[t/x]}\;\exists\mathbb{F}$$

Note that t may already occur in φ, e.g., φ might be $P(t,x)$. Thus, inferring $\mathbb{F}\,P(t,t)$ from $\mathbb{F}\,\exists x\,P(t,x)$ is a correct application of $\exists\mathbb{F}$. However, the eigenvariable conditions in $\forall\mathbb{F}$ and $\exists\mathbb{T}$ require that the constant symbol a does not occur in φ. So, you cannot correctly infer $\mathbb{F}\,P(a,a)$ from $\mathbb{F}\,\forall x\,P(a,x)$ using $\forall\mathbb{F}$.

In $\forall\mathbb{T}$ and $\exists\mathbb{F}$ there are no restrictions on the term t. On the other hand, in the $\exists\mathbb{T}$ and $\forall\mathbb{F}$ rules, the eigenvariable condition requires that the constant symbol a does not occur anywhere in the branches above the respective inference. explanation

[1] We use the term "eigenvariable" even though a in the above rule is a constant symbol. This has historical reasons.

It is necessary to ensure that the system is sound. Without this condition, the following would be a closed tableau for $\exists x\, \varphi(x) \to \forall x\, \varphi(x)$:

1.	$\mathbb{F}\, \exists x\, \varphi(x) \to \forall x\, \varphi(x)$	Assumption
2.	$\mathbb{T}\, \exists x\, \varphi(x)$	$\to\mathbb{F}\, 1$
3.	$\mathbb{F}\, \forall x\, \varphi(x)$	$\to\mathbb{F}\, 1$
4.	$\mathbb{T}\, \varphi(a)$	$\exists\mathbb{T}\, 2$
5.	$\mathbb{F}\, \varphi(a)$	$\forall\mathbb{F}\, 3$
	$\otimes$	

However, $\exists x\, \varphi(x) \to \forall x\, \varphi(x)$ is not valid.

8.4 Tableaux

explanation We've said what an assumption is, and we've given the rules of inference. Tableaux are inductively generated from these: each tableau either is a single branch consisting of one or more assumptions, or it results from a tableau by applying one of the rules of inference on a branch. fol:tab:der:sec

Definition 8.2 (Tableau). A tableau for assumptions $S_1\varphi_1, \ldots, S_n\varphi_n$ (where each S_i is either $\mathbb{T}$ or $\mathbb{F}$) is a finite tree of signed formulas satisfying the following conditions:

1. The n topmost signed formulas of the tree are $S_i\varphi_i$, one below the other.

2. Every signed formula in the tree that is not one of the assumptions results from a correct application of an inference rule to a signed formula in the branch above it.

A branch of a tableau is *closed* iff it contains both $\mathbb{T}\,\varphi$ and $\mathbb{F}\,\varphi$, and *open* otherwise. A tableau in which every branch is closed is a *closed tableau* (for its set of assumptions). If a tableau is not closed, i.e., if it contains at least one open branch, it is *open*.

Example 8.3. Every set of assumptions on its own is a tableau, but it will generally not be closed. (Obviously, it is closed only if the assumptions already contain a pair of signed formulas $\mathbb{T}\,\varphi$ and $\mathbb{F}\,\varphi$.)

From a tableau (open or closed) we can obtain a new, larger one by applying one of the rules of inference to a signed formula φ in it. The rule will append one or more signed formulas to the end of any branch containing the occurrence of φ to which we apply the rule.

For instance, consider the assumption $\mathbb{T}\,\varphi \wedge \neg\varphi$. Here is the (open) tableau consisting of just that assumption:

1.	$\mathbb{T}\,\varphi \wedge \neg\varphi$	Assumption

We obtain a new tableau from it by applying the $\wedge\mathbb{T}$ rule to the assumption. That rule allows us to add two new lines to the tableau, $\mathbb{T}\,\varphi$ and $\mathbb{T}\,\neg\varphi$:

1.	$\mathbb{T}\,\varphi \land \neg\varphi$	Assumption
2.	$\mathbb{T}\,\varphi$	$\land\mathbb{T}\,1$
3.	$\mathbb{T}\,\neg\varphi$	$\land\mathbb{T}\,1$

When we write down tableaux, we record the rules we've applied on the right (e.g., $\land\mathbb{T}1$ means that the signed formula on that line is the result of applying the $\land\mathbb{T}$ rule to the signed formula on line 1). This new tableau now contains additional signed formulas, but to only one ($\mathbb{T}\,\neg\varphi$) can we apply a rule (in this case, the $\neg\mathbb{T}$ rule). This results in the closed tableau

1.	$\mathbb{T}\,\varphi \land \neg\varphi$	Assumption
2.	$\mathbb{T}\,\varphi$	$\land\mathbb{T}\,1$
3.	$\mathbb{T}\,\neg\varphi$	$\land\mathbb{T}\,1$
4.	$\mathbb{F}\,\varphi$	$\neg\mathbb{T}\,3$
	$\otimes$	

8.5 Examples of Tableaux

fol:tab:pro:sec

Example 8.4. Let's find a closed tableau for the sentence $(\varphi \land \psi) \rightarrow \varphi$.

We begin by writing the corresponding assumption at the top of the tableau.

1.	$\mathbb{F}\,(\varphi \land \psi) \rightarrow \varphi$	Assumption

There is only one assumption, so only one signed formula to which we can apply a rule. (For every signed formula, there is always at most one rule that can be applied: it's the rule for the corresponding sign and main operator of the sentence.) In this case, this means, we must apply $\rightarrow\mathbb{F}$.

1.	$\mathbb{F}\,(\varphi \land \psi) \rightarrow \varphi$ ✓	Assumption
2.	$\mathbb{T}\,\varphi \land \psi$	$\rightarrow\mathbb{F}\,1$
3.	$\mathbb{F}\,\varphi$	$\rightarrow\mathbb{F}\,1$

To keep track of which signed formulas we have applied their corresponding rules to, we write a checkmark next to the sentence. However, *only* write a checkmark if the rule has been applied to all open branches. Once a signed formula has had the corresponding rule applied in every open branch, we will not have to return to it and apply the rule again. In this case, there is only one branch, so the rule only has to be applied once. (Note that checkmarks are only a convenience for constructing tableaux and are not officially part of the syntax of tableaux.)

There is one new signed formula to which we can apply a rule: the $\mathbb{T}\,\varphi \land \psi$ on line 2. Applying the $\land\mathbb{T}$ rule results in:

1.	$\mathbb{F}\,(\varphi \wedge \psi) \to \varphi$ ✓	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$ ✓	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,\varphi$	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\varphi$	$\wedge\mathbb{T}\,2$
5.	$\mathbb{T}\,\psi$	$\wedge\mathbb{T}\,2$
	$\otimes$	

Since the branch now contains both $\mathbb{T}\,\varphi$ (on line 4) and $\mathbb{F}\,\varphi$ (on line 3), the branch is closed. Since it is the only branch, the tableau is closed. We have found a closed tableau for $(\varphi \wedge \psi) \to \varphi$.

Example 8.5. Now let's find a closed tableau for $(\neg\varphi \vee \psi) \to (\varphi \to \psi)$.

We begin with the corresponding assumption:

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$	Assumption

The one signed formula in this tableau has main operator $\to$ and sign $\mathbb{F}$, so we apply the $\to\mathbb{F}$ rule to it to obtain:

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$ ✓	Assumption
2.	$\mathbb{T}\,\neg\varphi \vee \psi$	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,(\varphi \to \psi)$	$\to\mathbb{F}\,1$

We now have a choice as to whether to apply $\vee\mathbb{T}$ to line 2 or $\to\mathbb{F}$ to line 3. It actually doesn't matter which order we pick, as long as each signed formula has its corresponding rule applied in every branch. So let's pick the first one. The $\vee\mathbb{T}$ rule allows the tableau to branch, and the two conclusions of the rule will be the new signed formulas added to the two new branches. This results in:

1.	$\mathbb{F}\,(\neg\varphi \vee \psi) \to (\varphi \to \psi)$ ✓		Assumption
2.	$\mathbb{T}\,\neg\varphi \vee \psi$ ✓		$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,(\varphi \to \psi)$		$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\neg\varphi$	$\mathbb{T}\,\psi$	$\vee\mathbb{T}\,2$

We have not applied the $\to\mathbb{F}$ rule to line 3 yet: let's do that now. To save time, we apply it to both branches. Recall that we write a checkmark next to a signed formula only if we have applied the corresponding rule in every open branch. So it's a good idea to apply a rule at the end of every branch that contains the signed formula the rule applies to. That way we won't have to return to that signed formula lower down in the various branches.

$$
\begin{array}{lccl}
1. & \multicolumn{2}{c}{\mathbb{F}\,(\neg\varphi \lor \psi) \to (\varphi \to \psi) \checkmark} & \text{Assumption} \\
2. & \multicolumn{2}{c}{\mathbb{T}\,\neg\varphi \lor \psi \checkmark} & {\to}\mathbb{F}\,1 \\
3. & \multicolumn{2}{c}{\mathbb{F}\,(\varphi \to \psi) \checkmark} & {\to}\mathbb{F}\,1 \\
4. & \mathbb{T}\,\neg\varphi & \mathbb{T}\,\psi & {\lor}\mathbb{T}\,2 \\
5. & \mathbb{T}\,\varphi & \mathbb{T}\,\varphi & {\to}\mathbb{F}\,3 \\
6. & \mathbb{F}\,\psi & \mathbb{F}\,\psi & {\to}\mathbb{F}\,3 \\
 & & \otimes &
\end{array}
$$

The right branch is now closed. On the left branch, we can still apply the $\neg\mathbb{T}$ rule to line 4. This results in $\mathbb{F}\,\varphi$ and closes the left branch:

$$
\begin{array}{lccl}
1. & \multicolumn{2}{c}{\mathbb{F}\,(\neg\varphi \lor \psi) \to (\varphi \to \psi) \checkmark} & \text{Assumption} \\
2. & \multicolumn{2}{c}{\mathbb{T}\,\neg\varphi \lor \psi \checkmark} & {\to}\mathbb{F}\,1 \\
3. & \multicolumn{2}{c}{\mathbb{F}\,(\varphi \to \psi) \checkmark} & {\to}\mathbb{F}\,1 \\
4. & \mathbb{T}\,\neg\varphi & \mathbb{T}\,\psi & {\lor}\mathbb{T}\,2 \\
5. & \mathbb{T}\,\varphi & \mathbb{T}\,\varphi & {\to}\mathbb{F}\,3 \\
6. & \mathbb{F}\,\psi & \mathbb{F}\,\psi & {\to}\mathbb{F}\,3 \\
7. & \mathbb{F}\,\varphi & \otimes & \neg\mathbb{T}\,4 \\
 & \otimes & &
\end{array}
$$

Example 8.6. We can give tableaux for any number of signed formulas as assumptions. Often it is also necessary to apply more than one rule that allows branching; and in general a tableau can have any number of branches. For instance, consider a tableau for $\{\mathbb{T}\,\varphi \lor (\psi \land \chi), \mathbb{F}\,(\varphi \lor \psi) \land (\varphi \lor \chi)\}$. We start by applying the $\lor\mathbb{T}$ to the first assumption:

$$
\begin{array}{lccl}
1. & \multicolumn{2}{c}{\mathbb{T}\,\varphi \lor (\psi \land \chi) \checkmark} & \text{Assumption} \\
2. & \multicolumn{2}{c}{\mathbb{F}\,(\varphi \lor \psi) \land (\varphi \lor \chi)} & \text{Assumption} \\
3. & \mathbb{T}\,\varphi & \mathbb{T}\,\psi \land \chi & {\lor}\mathbb{T}\,1
\end{array}
$$

Now we can apply the $\land\mathbb{F}$ rule to line 2. We do this on both branches simultaneously, and can therefore check off line 2:

$$
\begin{array}{lccccl}
1. & \multicolumn{4}{c}{\mathbb{T}\,\varphi \lor (\psi \land \chi) \checkmark} & \text{Assumption} \\
2. & \multicolumn{4}{c}{\mathbb{F}\,(\varphi \lor \psi) \land (\varphi \lor \chi) \checkmark} & \text{Assumption} \\
3. & \multicolumn{2}{c}{\mathbb{T}\,\varphi} & \multicolumn{2}{c}{\mathbb{T}\,\psi \land \chi} & {\lor}\mathbb{T}\,1 \\
4. & \mathbb{F}\,\varphi \lor \psi & \mathbb{F}\,\varphi \lor \chi & \mathbb{F}\,\varphi \lor \psi & \mathbb{F}\,\varphi \lor \chi & {\land}\mathbb{F}\,2
\end{array}
$$

Now we can apply $\lor\mathbb{F}$ to all the branches containing $\varphi \lor \psi$:

1.		$\mathbb{T}\,\varphi \lor (\psi \land \chi)$ ✓			Assumption
2.		$\mathbb{F}\,(\varphi \lor \psi) \land (\varphi \lor \chi)$ ✓			Assumption
3.	$\mathbb{T}\,\varphi$		$\mathbb{T}\,\psi \land \chi$		$\lor\mathbb{T}\,1$
4.	$\mathbb{F}\,\varphi \lor \psi$ ✓	$\mathbb{F}\,\varphi \lor \chi$	$\mathbb{F}\,\varphi \lor \psi$ ✓	$\mathbb{F}\,\varphi \lor \chi$	$\land\mathbb{F}\,2$
5.	$\mathbb{F}\,\varphi$		$\mathbb{F}\,\varphi$		$\lor\mathbb{F}\,4$
6.	$\mathbb{F}\,\psi$		$\mathbb{F}\,\psi$		$\lor\mathbb{F}\,4$
	$\otimes$				

The leftmost branch is now closed. Let's now apply $\lor\mathbb{F}$ to $\varphi \lor \chi$:

1.		$\mathbb{T}\,\varphi \lor (\psi \land \chi)$ ✓			Assumption
2.		$\mathbb{F}\,(\varphi \lor \psi) \land (\varphi \lor \chi)$ ✓			Assumption
3.	$\mathbb{T}\,\varphi$		$\mathbb{T}\,\psi \land \chi$		$\lor\mathbb{T}\,1$
4.	$\mathbb{F}\,\varphi \lor \psi$ ✓	$\mathbb{F}\,\varphi \lor \chi$ ✓	$\mathbb{F}\,\varphi \lor \psi$ ✓	$\mathbb{F}\,\varphi \lor \chi$ ✓	$\land\mathbb{F}\,2$
5.	$\mathbb{F}\,\varphi$		$\mathbb{F}\,\varphi$		$\lor\mathbb{F}\,4$
6.	$\mathbb{F}\,\psi$		$\mathbb{F}\,\psi$		$\lor\mathbb{F}\,4$
7.	$\otimes$	$\mathbb{F}\,\varphi$		$\mathbb{F}\,\varphi$	$\lor\mathbb{F}\,4$
8.		$\mathbb{F}\,\chi$		$\mathbb{F}\,\chi$	$\lor\mathbb{F}\,4$
		$\otimes$			

Note that we moved the result of applying $\lor\mathbb{F}$ a second time below for clarity. In this instance it would not have been needed, since the justifications would have been the same.

Two branches remain open, and $\mathbb{T}\,\psi \land \chi$ on line 3 remains unchecked. We apply $\land\mathbb{T}$ to it to obtain a closed tableau:

1.		$\mathbb{T}\,\varphi \lor (\psi \land \chi)$ ✓			Assumption
2.		$\mathbb{F}\,(\varphi \lor \psi) \land (\varphi \lor \chi)$ ✓			Assumption
3.	$\mathbb{T}\,\varphi$		$\mathbb{T}\,\psi \land \chi$ ✓		$\lor\mathbb{T}\,1$
4.	$\mathbb{F}\,\varphi \lor \psi$ ✓	$\mathbb{F}\,\varphi \lor \chi$ ✓	$\mathbb{F}\,\varphi \lor \psi$ ✓	$\mathbb{F}\,\varphi \lor \chi$ ✓	$\land\mathbb{F}\,2$
5.	$\mathbb{F}\,\varphi$	$\mathbb{F}\,\varphi$	$\mathbb{F}\,\varphi$	$\mathbb{F}\,\varphi$	$\lor\mathbb{F}\,4$
6.	$\mathbb{F}\,\psi$	$\mathbb{F}\,\chi$	$\mathbb{F}\,\psi$	$\mathbb{F}\,\chi$	$\lor\mathbb{F}\,4$
7.	$\otimes$	$\otimes$	$\mathbb{T}\,\psi$	$\mathbb{T}\,\psi$	$\land\mathbb{T}\,3$
8.			$\mathbb{T}\,\chi$	$\mathbb{T}\,\chi$	$\land\mathbb{T}\,3$
			$\otimes$	$\otimes$	

For comparison, here's a closed tableau for the same set of assumptions in which the rules are applied in a different order:

1.	$\mathbb{T}\,\varphi \lor (\psi \land \chi)$ ✓				Assumption
2.	$\mathbb{F}\,(\varphi \lor \psi) \land (\varphi \lor \chi)$ ✓				Assumption
3.	$\mathbb{F}\,\varphi \lor \psi$ ✓		$\mathbb{F}\,\varphi \lor \chi$ ✓		$\land\mathbb{F}\,2$
4.	$\mathbb{F}\,\varphi$		$\mathbb{F}\,\varphi$		$\lor\mathbb{F}\,3$
5.	$\mathbb{F}\,\psi$		$\mathbb{F}\,\chi$		$\lor\mathbb{F}\,3$
6.	$\mathbb{T}\,\varphi$	$\mathbb{T}\,\psi \land \chi$ ✓	$\mathbb{T}\,\varphi$	$\mathbb{T}\,\psi \land \chi$ ✓	$\lor\mathbb{T}\,1$
7.	$\otimes$	$\mathbb{T}\,\psi$	$\otimes$	$\mathbb{T}\,\psi$	$\land\mathbb{T}\,6$
8.		$\mathbb{T}\,\chi$		$\mathbb{T}\,\chi$	$\land\mathbb{T}\,6$
		$\otimes$		$\otimes$	

Problem 8.1. Give closed tableaux of the following:

1. $\mathbb{T}\,\varphi \land (\psi \land \chi), \mathbb{F}\,(\varphi \land \psi) \land \chi$.
2. $\mathbb{T}\,\varphi \lor (\psi \lor \chi), \mathbb{F}\,(\varphi \lor \psi) \lor \chi$.
3. $\mathbb{T}\,\varphi \to (\psi \to \chi), \mathbb{F}\,\psi \to (\varphi \to \chi)$.
4. $\mathbb{T}\,\varphi, \mathbb{F}\,\neg\neg\varphi$.

Problem 8.2. Give closed tableaux of the following:

1. $\mathbb{T}\,(\varphi \lor \psi) \to \chi, \mathbb{F}\,\varphi \to \chi$.
2. $\mathbb{T}\,(\varphi \to \chi) \land (\psi \to \chi), \mathbb{F}\,(\varphi \lor \psi) \to \chi$.
3. $\mathbb{F}\,\neg(\varphi \land \neg\varphi)$.
4. $\mathbb{T}\,\psi \to \varphi, \mathbb{F}\,\neg\varphi \to \neg\psi$.
5. $\mathbb{F}\,(\varphi \to \neg\varphi) \to \neg\varphi$.
6. $\mathbb{F}\,\neg(\varphi \to \psi) \to \neg\psi$.
7. $\mathbb{T}\,\varphi \to \chi, \mathbb{F}\,\neg(\varphi \land \neg\chi)$.
8. $\mathbb{T}\,\varphi \land \neg\chi, \mathbb{F}\,\neg(\varphi \to \chi)$.
9. $\mathbb{T}\,\varphi \lor \psi, \neg\psi, \mathbb{F}\,\varphi$.
10. $\mathbb{T}\,\neg\varphi \lor \neg\psi, \mathbb{F}\,\neg(\varphi \land \psi)$.
11. $\mathbb{F}\,(\neg\varphi \land \neg\psi) \to \neg(\varphi \lor \psi)$.
12. $\mathbb{F}\,\neg(\varphi \lor \psi) \to (\neg\varphi \land \neg\psi)$.

Problem 8.3. Give closed tableaux of the following:

1. $\mathbb{T}\,\neg(\varphi \to \psi), \mathbb{F}\,\varphi$.

2. $\mathbb{T}\,\neg(\varphi \wedge \psi), \mathbb{F}\,\neg\varphi \vee \neg\psi$.

3. $\mathbb{T}\,\varphi \to \psi, \mathbb{F}\,\neg\varphi \vee \psi$.

4. $\mathbb{F}\,\neg\neg\varphi \to \varphi$.

5. $\mathbb{T}\,\varphi \to \psi, \mathbb{T}\,\neg\varphi \to \psi, \mathbb{F}\,\psi$.

6. $\mathbb{T}\,(\varphi \wedge \psi) \to \chi, \mathbb{F}\,(\varphi \to \chi) \vee (\psi \to \chi)$.

7. $\mathbb{T}\,(\varphi \to \psi) \to \varphi, \mathbb{F}\,\varphi$.

8. $\mathbb{F}\,(\varphi \to \psi) \vee (\psi \to \chi)$.

8.6 Tableaux with Quantifiers

fol:tab:prq:sec

Example 8.7. When dealing with quantifiers, we have to make sure not to violate the eigenvariable condition, and sometimes this requires us to play around with the order of carrying out certain inferences. In general, it helps to try and take care of rules subject to the eigenvariable condition first (they will be higher up in the finished tableau).

Let's see how we'd give a tableau for the sentence $\exists x\,\neg\varphi(x) \to \neg\forall x\,\varphi(x)$. Starting as usual, we start by recording the assumption,

1.	$\mathbb{F}\,\exists x\,\neg\varphi(x) \to \neg\forall x\,\varphi(x)$	Assumption

Since the main operator is $\to$, we apply the $\to\mathbb{F}$:

1.	$\mathbb{F}\,\exists x\,\neg\varphi(x) \to \neg\forall x\,\varphi(x)$ ✓	Assumption
2.	$\mathbb{T}\,\exists x\,\neg\varphi(x)$	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,\neg\forall x\,\varphi(x)$	$\to\mathbb{F}\,1$

The next line to deal with is 2. We use $\exists\mathbb{T}$. This requires a new constant symbol; since no constant symbols yet occur, we can pick any one, say, a.

1.	$\mathbb{F}\,\exists x\,\neg\varphi(x) \to \neg\forall x\,\varphi(x)$ ✓	Assumption
2.	$\mathbb{T}\,\exists x\,\neg\varphi(x)$ ✓	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,\neg\forall x\,\varphi(x)$	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\neg\varphi(a)$	$\exists\mathbb{T}\,2$

Now we apply $\neg\mathbb{F}$ to line 3:

1.	$\mathbb{F}\,\exists x\,\neg\varphi(x) \to \neg\forall x\,\varphi(x)$ ✓	Assumption
2.	$\mathbb{T}\,\exists x\,\neg\varphi(x)$ ✓	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,\neg\forall x\,\varphi(x)$ ✓	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\neg\varphi(a)$	$\exists\mathbb{T}\,2$
5.	$\mathbb{T}\,\forall x\,\varphi(x)$	$\neg\mathbb{F}\,3$

We obtain a closed tableau by applying $\neg\mathbb{T}$ to line 4, followed by $\forall\mathbb{T}$ to line 5.

1.	$\mathbb{F}\,\exists x\,\neg\varphi(x) \to \neg\forall x\,\varphi(x)$ ✓	Assumption
2.	$\mathbb{T}\,\exists x\,\neg\varphi(x)$ ✓	$\to\mathbb{F}\,1$
3.	$\mathbb{F}\,\neg\forall x\,\varphi(x)$ ✓	$\to\mathbb{F}\,1$
4.	$\mathbb{T}\,\neg\varphi(a)$	$\exists\mathbb{T}\,2$
5.	$\mathbb{T}\,\forall x\,\varphi(x)$	$\neg\mathbb{F}\,3$
6.	$\mathbb{F}\,\varphi(a)$	$\neg\mathbb{T}\,4$
7.	$\mathbb{T}\,\varphi(a)$	$\forall\mathbb{T}\,5$
	$\otimes$	

Example 8.8. Let's see how we'd give a tableau for the set

$$\mathbb{F}\,\exists x\,\chi(x,b), \mathbb{T}\,\exists x\,(\varphi(x) \land \psi(x)), \mathbb{T}\,\forall x\,(\psi(x) \to \chi(x,b)).$$

Starting as usual, we start with the assumptions:

1.	$\mathbb{F}\,\exists x\,\chi(x,b)$	Assumption
2.	$\mathbb{T}\,\exists x\,(\varphi(x) \land \psi(x))$	Assumption
3.	$\mathbb{T}\,\forall x\,(\psi(x) \to \chi(x,b))$	Assumption

We should always apply a rule with the eigenvariable condition first; in this case that would be $\exists\mathbb{T}$ to line 2. Since the assumptions contain the constant symbol b, we have to use a different one; let's pick a again.

1.	$\mathbb{F}\,\exists x\,\chi(x,b)$	Assumption
2.	$\mathbb{T}\,\exists x\,(\varphi(x) \land \psi(x))$ ✓	Assumption
3.	$\mathbb{T}\,\forall x\,(\psi(x) \to \chi(x,b))$	Assumption
4.	$\mathbb{T}\,\varphi(a) \land \psi(a)$	$\exists\mathbb{T}\,2$

If we now apply $\exists\mathbb{F}$ to line 1 or $\forall\mathbb{T}$ to line 3, we have to decide which term t to substitute for x. Since there is no eigenvariable condition for these rules, we can pick any term we like. In some cases we may even have to apply the rule several times with different ts. But as a general rule, it pays to pick one of the terms already occurring in the tableau—in this case, a and b—and in this case we can guess that a will be more likely to result in a closed branch.

1.	$\mathbb{F}\,\exists x\,\chi(x,b)$	Assumption
2.	$\mathbb{T}\,\exists x\,(\varphi(x) \land \psi(x))$ ✓	Assumption
3.	$\mathbb{T}\,\forall x\,(\psi(x) \to \chi(x,b))$	Assumption
4.	$\mathbb{T}\,\varphi(a) \land \psi(a)$	$\exists\mathbb{T}\,2$
5.	$\mathbb{F}\,\chi(a,b)$	$\exists\mathbb{F}\,1$
6.	$\mathbb{T}\,\psi(a) \to \chi(a,b)$	$\forall\mathbb{T}\,3$

We don't check the signed formulas in lines 1 and 3, since we may have to use them again. Now apply $\land\mathbb{T}$ to line 4:

$$
\begin{array}{lcl}
1. & \mathbb{F}\,\exists x\,\chi(x,b) & \text{Assumption}\\
2. & \mathbb{T}\,\exists x\,(\varphi(x)\land\psi(x))\ \checkmark & \text{Assumption}\\
3. & \mathbb{T}\,\forall x\,(\psi(x)\to\chi(x,b)) & \text{Assumption}\\
4. & \mathbb{T}\,\varphi(a)\land\psi(a)\ \checkmark & \exists\mathbb{T}\,2\\
5. & \mathbb{F}\,\chi(a,b) & \exists\mathbb{F}\,1\\
6. & \mathbb{T}\,\psi(a)\to\chi(a,b) & \forall\mathbb{T}\,3\\
7. & \mathbb{T}\,\varphi(a) & \land\mathbb{T}\,4\\
8. & \mathbb{T}\,\psi(a) & \land\mathbb{T}\,4
\end{array}
$$

If we now apply $\to\mathbb{T}$ to line 6, the tableau closes:

$$
\begin{array}{lcl}
1. & \mathbb{F}\,\exists x\,\chi(x,b) & \text{Assumption}\\
2. & \mathbb{T}\,\exists x\,(\varphi(x)\land\psi(x))\ \checkmark & \text{Assumption}\\
3. & \mathbb{T}\,\forall x\,(\psi(x)\to\chi(x,b)) & \text{Assumption}\\
4. & \mathbb{T}\,\varphi(a)\land\psi(a)\ \checkmark & \exists\mathbb{T}\,2\\
5. & \mathbb{F}\,\chi(a,b) & \exists\mathbb{F}\,1\\
6. & \mathbb{T}\,\psi(a)\to\chi(a,b)\ \checkmark & \forall\mathbb{T}\,3\\
7. & \mathbb{T}\,\varphi(a) & \land\mathbb{T}\,4\\
8. & \mathbb{T}\,\psi(a) & \land\mathbb{T}\,4\\
9. & \begin{array}{cc}\mathbb{F}\,\psi(a) & \mathbb{T}\,\chi(a,b)\\ \otimes & \otimes\end{array} & \to\mathbb{T}\,6
\end{array}
$$

Example 8.9. We construct a tableau for the set

$$\mathbb{T}\,\forall x\,\varphi(x), \mathbb{T}\,\forall x\,\varphi(x)\to\exists y\,\psi(y), \mathbb{T}\,\lnot\exists y\,\psi(y).$$

Starting as usual, we write down the assumptions:

$$
\begin{array}{lcl}
1. & \mathbb{T}\,\forall x\,\varphi(x) & \text{Assumption}\\
2. & \mathbb{T}\,\forall x\,\varphi(x)\to\exists y\,\psi(y) & \text{Assumption}\\
3. & \mathbb{T}\,\lnot\exists y\,\psi(y) & \text{Assumption}
\end{array}
$$

We begin by applying the $\lnot\mathbb{T}$ rule to line 3. A corollary to the rule "always apply rules with eigenvariable conditions first" is "defer applying quantifier rules without eigenvariable conditions until needed." Also, defer rules that result in a split.

$$
\begin{array}{lcl}
1. & \mathbb{T}\,\forall x\,\varphi(x) & \text{Assumption}\\
2. & \mathbb{T}\,\forall x\,\varphi(x)\to\exists y\,\psi(y) & \text{Assumption}\\
3. & \mathbb{T}\,\lnot\exists y\,\psi(y)\ \checkmark & \text{Assumption}\\
4. & \mathbb{F}\,\exists y\,\psi(y) & \lnot\mathbb{T}\,3
\end{array}
$$

The new line 4 requires $\exists\mathbb{F}$, a quantifier rule without the eigenvariable condition. So we defer this in favor of using $\to\mathbb{T}$ on line 2.

1.	$\mathbb{T} \forall x\, \varphi(x)$		Assumption
2.	$\mathbb{T} \forall x\, \varphi(x) \to \exists y\, \psi(y)$ ✓		Assumption
3.	$\mathbb{T} \neg \exists y\, \psi(y)$ ✓		Assumption
4.	$\mathbb{F} \exists y\, \psi(y)$		$\neg\mathbb{T}\, 3$
5.	$\mathbb{F} \forall x\, \varphi(x)$	$\mathbb{T} \exists y\, \psi(y)$	$\to\mathbb{T}\, 2$

Both new signed formulas require rules with eigenvariable conditions, so these should be next:

1.	$\mathbb{T} \forall x\, \varphi(x)$		Assumption
2.	$\mathbb{T} \forall x\, \varphi(x) \to \exists y\, \psi(y)$ ✓		Assumption
3.	$\mathbb{T} \neg \exists y\, \psi(y)$ ✓		Assumption
4.	$\mathbb{F} \exists y\, \psi(y)$		$\neg\mathbb{T}\, 3$
5.	$\mathbb{F} \forall x\, \varphi(x)$ ✓	$\mathbb{T} \exists y\, \psi(y)$ ✓	$\to\mathbb{T}\, 2$
6.	$\mathbb{F}\, \varphi(b)$	$\mathbb{T}\, \psi(c)$	$\forall\mathbb{F}\, 5$; $\exists\mathbb{T}\, 5$

To close the branches, we have to use the signed formulas on lines 1 and 3. The corresponding rules ($\forall\mathbb{T}$ and $\exists\mathbb{F}$) don't have eigenvariable conditions, so we are free to pick whichever terms are suitable. In this case, that's b and c, respectively.

1.	$\mathbb{T} \forall x\, \varphi(x)$		Assumption
2.	$\mathbb{T} \forall x\, \varphi(x) \to \exists y\, \psi(y)$ ✓		Assumption
3.	$\mathbb{T} \neg \exists y\, \psi(y)$ ✓		Assumption
4.	$\mathbb{F} \exists y\, \psi(y)$		$\neg\mathbb{T}\, 3$
5.	$\mathbb{F} \forall x\, \varphi(x)$ ✓	$\mathbb{T} \exists y\, \psi(y)$ ✓	$\to\mathbb{T}\, 2$
6.	$\mathbb{F}\, \varphi(b)$	$\mathbb{T}\, \psi(c)$	$\forall\mathbb{F}\, 5$; $\exists\mathbb{T}\, 5$
7.	$\mathbb{T}\, \varphi(b)$	$\mathbb{F}\, \psi(c)$	$\forall\mathbb{T}\, 1$; $\exists\mathbb{F}\, 4$
	$\otimes$	$\otimes$	

Problem 8.4. Give closed tableaux of the following:

1. $\mathbb{F}\, (\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \forall z\, (\varphi(z) \land \psi(z))$.
2. $\mathbb{F}\, (\exists x\, \varphi(x) \lor \exists y\, \psi(y)) \to \exists z\, (\varphi(z) \lor \psi(z))$.
3. $\mathbb{T} \forall x\, (\varphi(x) \to \psi), \mathbb{F} \exists y\, \varphi(y) \to \psi$.
4. $\mathbb{T} \forall x\, \neg\varphi(x), \mathbb{F} \neg\exists x\, \varphi(x)$.
5. $\mathbb{F} \neg\exists x\, \varphi(x) \to \forall x\, \neg\varphi(x)$.
6. $\mathbb{F} \neg\exists x\, \forall y\, ((\varphi(x, y) \to \neg\varphi(y, y)) \land (\neg\varphi(y, y) \to \varphi(x, y)))$.

Problem 8.5. Give closed tableaux of the following:

1. $\mathbb{F}\,\lnot\forall x\,\varphi(x) \rightarrow \exists x\,\lnot\varphi(x)$.
2. $\mathbb{T}\,(\forall x\,\varphi(x) \rightarrow \psi), \mathbb{F}\,\exists y\,(\varphi(y) \rightarrow \psi)$.
3. $\mathbb{F}\,\exists x\,(\varphi(x) \rightarrow \forall y\,\varphi(y))$.

8.7 Proof-Theoretic Notions

fol:tab:ptn:sec

This section collects the definitions of the provability relation and consistency for tableaux.

explanation

Just as we've defined a number of important semantic notions (validity, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the existence of certain closed tableaux. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*.

Definition 8.10 (Theorems). A sentence φ is a *theorem* if there is a closed tableau for $\mathbb{F}\,\varphi$. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definition 8.11 (Derivability). A sentence φ is *derivable from* a set of sentences Γ, $\Gamma \vdash \varphi$ iff there is a finite set $\{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ and a closed tableau for the set

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}.$$

If φ is not derivable from Γ we write $\Gamma \nvdash \varphi$.

Definition 8.12 (Consistency). A set of sentences Γ is *inconsistent* iff there is a finite set $\{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ and a closed tableau for the set

$$\{\mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}.$$

If Γ is not inconsistent, we say it is *consistent*.

Proposition 8.13 (Reflexivity). *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.* fol:tab:ptn:prop:reflexivity

Proof. If $\varphi \in \Gamma$, $\{\varphi\}$ is a finite subset of Γ and the tableau

1.	$\mathbb{F}\,\varphi$	Assumption
2.	$\mathbb{T}\,\varphi$	Assumption
	$\otimes$	

is closed. □

fol:tab:ptn:
prop:monotonicity

Proposition 8.14 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.*

Proof. Any finite subset of Γ is also a finite subset of Δ. □

fol:tab:ptn:
prop:transitivity

Proposition 8.15 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.*

Proof. If $\{\varphi\} \cup \Delta \vdash \psi$, then there is a finite subset $\Delta_0 = \{\chi_1, \dots, \chi_n\} \subseteq \Delta$ such that

$$\{\mathbb{F}\,\psi, \mathbb{T}\,\varphi, \mathbb{T}\,\chi_1, \dots, \mathbb{T}\,\chi_n\}$$

has a closed tableau. If $\Gamma \vdash \varphi$ then there are $\theta_1, \dots, \theta_m$ such that

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\theta_1, \dots, \mathbb{T}\,\theta_m\}$$

has a closed tableau.

Now consider the tableau with assumptions

$$\mathbb{F}\,\psi, \mathbb{T}\,\chi_1, \dots, \mathbb{T}\,\chi_n, \mathbb{T}\,\theta_1, \dots, \mathbb{T}\,\theta_m.$$

Apply the Cut rule on φ. This generates two branches, one has $\mathbb{T}\,\varphi$ in it, the other $\mathbb{F}\,\varphi$. Thus, on the one branch, all of

$$\{\mathbb{F}\,\psi, \mathbb{T}\,\varphi, \mathbb{T}\,\chi_1, \dots, \mathbb{T}\,\chi_n\}$$

are available. Since there is a closed tableau for these assumptions, we can attach it to that branch; every branch through $\mathbb{T}\,\varphi$ closes. On the other branch, all of

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\theta_1, \dots, \mathbb{T}\,\theta_m\}$$

are available, so we can also complete the other side to obtain a closed tableau. This shows $\Gamma \cup \Delta \vdash \psi$. □

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \dots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i, then $\Gamma \vdash \psi$.

fol:tab:ptn:
prop:incons

Proposition 8.16. *Γ is inconsistent iff $\Gamma \vdash \varphi$ for every sentence φ.*

Proof. Exercise. □

Problem 8.6. Prove Proposition 8.16

fol:tab:ptn:
prop:proves-compact

Proposition 8.17 (Compactness).

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Proof. 1. If $\Gamma \vdash \varphi$, then there is a finite subset $\Gamma_0 = \{\psi_1, \dots, \psi_n\}$ and a closed tableau for

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}$$

This tableau also shows $\Gamma_0 \vdash \varphi$.

2. If Γ is inconsistent, then for some finite subset $\Gamma_0 = \{\psi_1, \dots, \psi_n\}$ there is a closed tableau for

$$\{\mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}$$

This closed tableau shows that Γ_0 is inconsistent. □

8.8 Derivability and Consistency

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem. fol:tab:prv:sec

Proposition 8.18. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.* fol:tab:prv:prop:provability-contr

Proof. There are finite $\Gamma_0 = \{\psi_1, \dots, \psi_n\}$ and $\Gamma_1 = \{\chi_1, \dots, \chi_n\} \subseteq \Gamma$ such that

$$\{\mathbb{F}\,\varphi,\mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}$$
$$\{\mathbb{T}\,\varphi,\mathbb{T}\,\chi_1, \dots, \mathbb{T}\,\chi_m\}$$

have closed tableaux. Using the Cut rule on φ we can combine these into a single closed tableau that shows $\Gamma_0 \cup \Gamma_1$ is inconsistent. Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$, hence Γ is inconsistent. □

Proposition 8.19. *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.* fol:tab:prv:prop:prov-incons

Proof. First suppose $\Gamma \vdash \varphi$, i.e., there is a closed tableau for

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}$$

Using the $\neg\mathbb{T}$ rule, this can be turned into a closed tableau for

$$\{\mathbb{T}\,\neg\varphi, \mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}.$$

On the other hand, if there is a closed tableau for the latter, we can turn it into a closed tableau of the former by removing every formula that results from $\neg\mathbb{T}$ applied to the first assumption $\mathbb{T}\,\neg\varphi$ as well as that assumption, and adding the assumption $\mathbb{F}\,\varphi$. For if a branch was closed before because it contained the conclusion of $\neg\mathbb{T}$ applied to $\mathbb{T}\,\neg\varphi$, i.e., $\mathbb{F}\,\varphi$, the corresponding branch in the new tableau is also closed. If a branch in the old tableau was closed because it contained the assumption $\mathbb{T}\,\neg\varphi$ as well as $\mathbb{F}\,\neg\varphi$ we can turn it into a closed branch by applying $\neg\mathbb{F}$ to $\mathbb{F}\,\neg\varphi$ to obtain $\mathbb{T}\,\varphi$. This closes the branch since we added $\mathbb{F}\,\varphi$ as an assumption. □

Problem 8.7. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

fol:tab:prv: prop:explicit-inc

Proposition 8.20. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.*

Proof. Suppose $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$. Then there are $\psi_1, \ldots, \psi_n \in \Gamma$ such that

$$\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\}$$

has a closed tableau. Replace the assumption $\mathbb{F}\,\varphi$ by $\mathbb{T}\,\neg\varphi$, and insert the conclusion of $\neg\mathbb{T}$ applied to $\mathbb{F}\,\varphi$ after the assumptions. Any sentence in the tableau justified by appeal to line 1 in the old tableau is now justified by appeal to line $n+1$. So if the old tableau was closed, the new one is. It shows that Γ is inconsistent, since all assumptions are in Γ. □

fol:tab:prv: prop:provability-exhaustive

Proposition 8.21. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

Proof. If there are $\psi_1, \ldots, \psi_n \in \Gamma$ and $\chi_1, \ldots, \chi_m \in \Gamma$ such that

$$\{\mathbb{T}\,\varphi, \mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n\} \text{ and}$$
$$\{\mathbb{T}\,\neg\varphi, \mathbb{T}\,\chi_1, \ldots, \mathbb{T}\,\chi_m\}$$

both have closed tableaux, we can construct a single, combined tableau that shows that Γ is inconsistent by using as assumptions $\mathbb{T}\,\psi_1, \ldots, \mathbb{T}\,\psi_n$ together with $\mathbb{T}\,\chi_1, \ldots, \mathbb{T}\,\chi_m$, followed by an application of the Cut rule. This yields two branches, one starting with $\mathbb{T}\,\varphi$, the other with $\mathbb{F}\,\varphi$.

On the left left side, add the part of the first tableau below its assumptions. Here, every rule application is still correct, since each of the assumptions of the first tableau, including $\mathbb{T}\,\varphi$, is available. Thus, every branch below $\mathbb{T}\,\varphi$ closes.

On the right side, add the part of the second tableau below its assumption, with the results of any applications of $\neg\mathbb{T}$ to $\mathbb{T}\,\neg\varphi$ removed. The conclusion of $\neg\mathbb{T}$ to $\mathbb{T}\,\neg\varphi$ is $\mathbb{F}\,\varphi$, which is nevertheless available, as it is the conclusion of the Cut rule on the right side of the combined tableau.

If a branch in the second tableau was closed because it contained the assumption $\mathbb{T}\,\neg\varphi$ (which no longer appears as an assumption in the combined tableau) as well as $\mathbb{F}\,\neg\varphi$, we can applying $\neg\mathbb{F}$ to $\mathbb{F}\,\neg\varphi$ to obtain $\mathbb{T}\,\varphi$. Now the corresponding branch in the combined tableau also closes, because it contains the right-hand conclusion of the Cut rule, $\mathbb{F}\,\varphi$. If a branch in the second tableau closed for any other reason, the corresponding branch in the combined tableau also closes, since any signed formulas other than $\mathbb{T}\,\neg\varphi$ occurring on the branch in the old, second tableau also occur on the corresponding branch in the combined tableau. □

8.9 Derivability and the Propositional Connectives

fol:tab:ppr: sec

We establish that the derivability relation $\vdash$ of tableaux is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \rightarrow \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem. explanation

Proposition 8.22. fol:tab:ppr: prop:provability-land fol:tab:ppr: prop:provability-land-left fol:tab:ppr: prop:provability-land-right

1. *Both* $\varphi \wedge \psi \vdash \varphi$ *and* $\varphi \wedge \psi \vdash \psi$.
2. $\varphi, \psi \vdash \varphi \wedge \psi$.

Proof. 1. Both $\{\mathbb{F}\,\varphi, \mathbb{T}\,\varphi \wedge \psi\}$ and $\{\mathbb{F}\,\psi, \mathbb{T}\,\varphi \wedge \psi\}$ have closed tableaux

1.	$\mathbb{F}\,\varphi$	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$	Assumption
3.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}\,2$
4.	$\mathbb{T}\psi$	$\wedge\mathbb{T}\,2$
	$\otimes$	

1.	$\mathbb{F}\,\psi$	Assumption
2.	$\mathbb{T}\,\varphi \wedge \psi$	Assumption
3.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}\,2$
4.	$\mathbb{T}\psi$	$\wedge\mathbb{T}\,2$
	$\otimes$	

2. Here is a closed tableau for $\{\mathbb{T}\,\varphi, \mathbb{T}\,\psi, \mathbb{F}\,\varphi \wedge \psi\}$:

1.	$\mathbb{F}\,\varphi \wedge \psi$		Assumption
2.	$\mathbb{T}\,\varphi$		Assumption
3.	$\mathbb{T}\,\psi$		Assumption
4.	$\mathbb{F}\,\varphi$	$\mathbb{F}\,\psi$	$\wedge\mathbb{F}\,1$
	$\otimes$	$\otimes$	

Proposition 8.23. fol:tab:ppr: prop:provability-lor

1. $\{\varphi \vee \psi, \neg\varphi, \neg\psi\}$ *is inconsistent.*
2. *Both* $\varphi \vdash \varphi \vee \psi$ *and* $\psi \vdash \varphi \vee \psi$.

Proof. 1. We give a closed tableau of $\{\mathbb{T}\,\varphi \vee \psi, \mathbb{T}\,\neg\varphi, \mathbb{T}\,\neg\psi\}$:

1.	$\mathbb{T}\,\varphi \vee \psi$		Assumption
2.	$\mathbb{T}\,\neg\varphi$		Assumption
3.	$\mathbb{T}\,\neg\psi$		Assumption
4.	$\mathbb{F}\,\varphi$		$\neg\mathbb{T}\,2$
5.	$\mathbb{F}\,\psi$		$\neg\mathbb{T}\,3$
6.	$\mathbb{T}\,\varphi$	$\mathbb{T}\,\psi$	$\vee\mathbb{T}\,1$
	$\otimes$	$\otimes$	

2. Both $\{\mathbb{F}\,\varphi \lor \psi, \mathbb{T}\,\varphi\}$ and $\{\mathbb{F}\,\varphi \lor \psi, \mathbb{T}\,\psi\}$ have closed tableaux:

1.	$\mathbb{F}\,\varphi \lor \psi$	Assumption
2.	$\mathbb{T}\,\varphi$	Assumption
3.	$\mathbb{F}\varphi$	$\lor\mathbb{F}\,1$
4.	$\mathbb{F}\psi$	$\lor\mathbb{F}\,1$
	$\otimes$	

1.	$\mathbb{F}\,\varphi \lor \psi$	Assumption
2.	$\mathbb{T}\,\psi$	Assumption
3.	$\mathbb{F}\varphi$	$\lor\mathbb{F}\,1$
4.	$\mathbb{F}\psi$	$\lor\mathbb{F}\,1$
	$\otimes$	

fol:tab:ppr:prop:provability-lif

Proposition 8.24.

fol:tab:ppr:prop:provability-lif-left

1. *$\varphi, \varphi \to \psi \vdash \psi$.*

fol:tab:ppr:prop:provability-lif-right

2. *Both $\neg\varphi \vdash \varphi \to \psi$ and $\psi \vdash \varphi \to \psi$.*

Proof. 1. $\{\mathbb{F}\,\psi, \mathbb{T}\,\varphi \to \psi, \mathbb{T}\,\varphi\}$ has a closed tableau:

1.	$\mathbb{F}\,\psi$		Assumption
2.	$\mathbb{T}\,\varphi \to \psi$		Assumption
3.	$\mathbb{T}\,\varphi$		Assumption
4.	$\mathbb{F}\,\varphi$	$\mathbb{T}\,\psi$	$\to\mathbb{T}\,2$
	$\otimes$	$\otimes$	

2. Both $\{\mathbb{F}\,\varphi \to \psi, \mathbb{T}\,\neg\varphi\}$ and $\{\mathbb{F}\,\varphi \to \psi, \mathbb{T}\,\psi\}$ have closed tableaux:

1.	$\mathbb{F}\,\varphi \to \psi$	Assumption
2.	$\mathbb{T}\,\neg\varphi$	Assumption
3.	$\mathbb{T}\,\varphi$	$\to\mathbb{F}\,1$
4.	$\mathbb{F}\,\psi$	$\to\mathbb{F}\,1$
5.	$\mathbb{F}\,\varphi$	$\neg\mathbb{T}\,2$
	$\otimes$	

1.	$\mathbb{F}\,\varphi \to \psi$	Assumption
2.	$\mathbb{T}\,\psi$	Assumption
3.	$\mathbb{T}\,\varphi$	$\to\mathbb{F}\,1$
4.	$\mathbb{F}\,\psi$	$\to\mathbb{F}\,1$
	$\otimes$	

8.10 Derivability and the Quantifiers

explanation The completeness theorem also requires that the tableaux rules yield the facts about $\vdash$ established in this section. fol:tab:qpr: sec

Theorem 8.25. *If c is a constant not occurring in Γ or $\varphi(x)$ and $\Gamma \vdash \varphi(c)$, then $\Gamma \vdash \forall x\, \varphi(x)$.* fol:tab:qpr: thm:strong-generalization

Proof. Suppose $\Gamma \vdash \varphi(c)$, i.e., there are $\psi_1, \dots, \psi_n \in \Gamma$ and a closed tableau for

$$\{\mathbb{F}\,\varphi(c), \mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}.$$

We have to show that there is also a closed tableau for

$$\{\mathbb{F}\,\forall x\, \varphi(x), \mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}.$$

Take the closed tableau and replace the first assumption with $\mathbb{F}\,\forall x\, \varphi(x)$, and insert $\mathbb{F}\,\varphi(c)$ after the assumptions.

$$\begin{array}{c} \mathbb{F}\,\varphi(c) \\ \mathbb{T}\,\psi_1 \\ \vdots \\ \mathbb{T}\,\psi_n \end{array} \qquad \begin{array}{c} \mathbb{F}\,\forall x\, \varphi(x) \\ \mathbb{T}\,\psi_1 \\ \vdots \\ \mathbb{T}\,\psi_n \\ \mathbb{F}\,\varphi(c) \end{array}$$

The tableau is still closed, since all sentences available as assumptions before are still available at the top of the tableau. The inserted line is the result of a correct application of $\forall\mathbb{F}$, since the constant symbol c does not occur in ψ_1, $\dots$, ψ_n or $\forall x\, \varphi(x)$, i.e., it does not occur above the inserted line in the new tableau. □

Proposition 8.26. fol:tab:qpr: prop:provability-quantifiers

1. $\varphi(t) \vdash \exists x\, \varphi(x)$.
2. $\forall x\, \varphi(x) \vdash \varphi(t)$.

Proof. 1. A closed tableau for $\mathbb{F}\,\exists x\, \varphi(x), \mathbb{T}\,\varphi(t)$ is:

1.	$\mathbb{F}\,\exists x\, \varphi(x)$	Assumption
2.	$\mathbb{T}\,\varphi(t)$	Assumption
3.	$\mathbb{F}\,\varphi(t)$	$\exists\mathbb{F}\,1$
	$\otimes$	

2. A closed tableau for $\mathbb{F}\,\varphi(t), \mathbb{T}\,\forall x\, \varphi(x)$, is:

1.	$\mathbb{F}\,\varphi(t)$	Assumption
2.	$\mathbb{T}\,\forall x\,\varphi(x)$	Assumption
3.	$\mathbb{T}\,\varphi(t)$	$\forall\mathbb{T}\,2$
	$\otimes$	

8.11 Soundness

fol:tab:sou: sec A derivation system, such as tableaux, is *sound* if it cannot derive things that do not actually hold. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that explanation

1. every derivable φ is valid;
2. if a sentence is derivable from some others, it is also a consequence of them;
3. if a set of sentences is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Because all these proof-theoretic properties are defined via closed tableaux of some kind or other, proving (1)–(3) above requires proving something about the semantic properties of closed tableaux. We will first define what it means for a signed formula to be satisfied in a structure, and then show that if a tableau is closed, no structure satisfies all its assumptions. (1)–(3) then follow as corollaries from this result.

Definition 8.27. A structure $\mathfrak{M}$ *satisfies* a signed formula $\mathbb{T}\,\varphi$ iff $\mathfrak{M} \vDash \varphi$, and it satisfies $\mathbb{F}\,\varphi$ iff $\mathfrak{M} \nvDash \varphi$. $\mathfrak{M}$ satisfies a set of signed formulas Γ iff it satisfies every $S\,\varphi \in \Gamma$. Γ is *satisfiable* if there is a structure that satisfies it, and *unsatisfiable* otherwise.

fol:tab:sou: thm:tableau-soundness **Theorem 8.28 (Soundness).** *If Γ has a closed tableau, Γ is unsatisfiable.*

Proof. Let's call a branch of a tableau satisfiable iff the set of signed formulas on it is satisfiable, and let's call a tableau satisfiable if it contains at least one satisfiable branch.

We show the following: Extending a satisfiable tableau by one of the rules of inference always results in a satisfiable tableau. This will prove the theorem: any closed tableau results by applying rules of inference to the tableau consisting only of assumptions from Γ. So if Γ were satisfiable, any tableau for it would be satisfiable. A closed tableau, however, is clearly not satisfiable:

every branch contains both $\mathbb{T}\,\varphi$ and $\mathbb{F}\,\varphi$, and no structure can both satisfy and not satisfy φ.

Suppose we have a satisfiable tableau, i.e., a tableau with at least one satisfiable branch. Applying a rule of inference either adds signed formulas to a branch, or splits a branch in two. If the tableau has a satisfiable branch which is not extended by the rule application in question, it remains a satisfiable branch in the extended tableau, so the extended tableau is satisfiable. So we only have to consider the case where a rule is applied to a satisfiable branch.

Let Γ be the set of signed formulas on that branch, and let $S\,\varphi \in \Gamma$ be the signed formula to which the rule is applied. If the rule does not result in a split branch, we have to show that the extended branch, i.e., Γ together with the conclusions of the rule, is still satisfiable. If the rule results in a split branch, we have to show that at least one of the two resulting branches is satisfiable.

First, we consider the possible inferences that do not result in a split branch.

1. The branch is expanded by applying $\neg\mathbb{T}$ to $\mathbb{T}\,\neg\psi \in \Gamma$. Then the extended branch contains the signed formulas $\Gamma \cup \{\mathbb{F}\,\psi\}$. Suppose $\mathfrak{M} \vDash \Gamma$. In particular, $\mathfrak{M} \vDash \neg\psi$. Thus, $\mathfrak{M} \nvDash \psi$, i.e., $\mathfrak{M}$ satisfies $\mathbb{F}\,\psi$.

2. The branch is expanded by applying $\neg\mathbb{F}$ to $\mathbb{F}\,\neg\psi \in \Gamma$: Exercise.

3. The branch is expanded by applying $\wedge\mathbb{T}$ to $\mathbb{T}\,\psi \wedge \chi \in \Gamma$, which results in two new signed formulas on the branch: $\mathbb{T}\,\psi$ and $\mathbb{T}\,\chi$. Suppose $\mathfrak{M} \vDash \Gamma$, in particular $\mathfrak{M} \vDash \psi \wedge \chi$. Then $\mathfrak{M} \vDash \psi$ and $\mathfrak{M} \vDash \chi$. This means that $\mathfrak{M}$ satisfies both $\mathbb{T}\,\psi$ and $\mathbb{T}\,\chi$.

4. The branch is expanded by applying $\vee\mathbb{F}$ to $\mathbb{F}\,\psi \vee \chi \in \Gamma$: Exercise.

5. The branch is expanded by applying $\rightarrow\mathbb{F}$ to $\mathbb{F}\,\psi \rightarrow \chi \in \Gamma$: This results in two new signed formulas on the branch: $\mathbb{T}\,\psi$ and $\mathbb{F}\,\chi$. Suppose $\mathfrak{M} \vDash \Gamma$, in particular $\mathfrak{M} \nvDash \psi \rightarrow \chi$. Then $\mathfrak{M} \vDash \psi$ and $\mathfrak{M} \nvDash \chi$. This means that $\mathfrak{M}$ satisfies both $\mathbb{T}\,\psi$ and $\mathbb{F}\,\chi$.

6. The branch is expanded by applying $\forall\mathbb{T}$ to $\mathbb{T}\,\forall x\,\psi(x) \in \Gamma$: This results in a new signed formula $\mathbb{T}\,\varphi(t)$ on the branch. Suppose $\mathfrak{M} \vDash \Gamma$, in particular, $\mathfrak{M} \vDash \forall x\,\varphi(x)$. By Proposition 3.30, $\mathfrak{M} \vDash \varphi(t)$. Consequently, $\mathfrak{M}$ satisfies $\mathbb{T}\,\varphi(t)$.

7. The branch is expanded by applying $\forall\mathbb{F}$ to $\mathbb{F}\,\forall x\,\psi(x) \in \Gamma$: This results in a new signed formula $\mathbb{F}\,\varphi(a)$ where a is a constant symbol not occurring in Γ. Since Γ is satisfiable, there is a $\mathfrak{M}$ such that $\mathfrak{M} \vDash \Gamma$, in particular $\mathfrak{M} \nvDash \forall x\,\psi(x)$. We have to show that $\Gamma \cup \{\mathbb{F}\,\varphi(a)\}$ is satisfiable. To do this, we define a suitable $\mathfrak{M}'$ as follows.

   By Proposition 3.18, $\mathfrak{M} \nvDash \forall x\,\psi(x)$ iff for some s, $\mathfrak{M}, s \nvDash \psi(x)$. Now let $\mathfrak{M}'$ be just like $\mathfrak{M}$, except $a^{\mathfrak{M}'} = s(x)$. By Corollary 3.20, for any $\mathbb{T}\,\chi \in \Gamma$, $\mathfrak{M}' \vDash \chi$, and for any $\mathbb{F}\,\chi \in \Gamma$, $\mathfrak{M}' \nvDash \chi$, since a does not occur in Γ.

By Proposition 3.19, $\mathfrak{M}', s \nvDash \varphi(x)$. By Proposition 3.22, $\mathfrak{M}', s \nvDash \varphi(a)$. Since $\varphi(a)$ is a sentence, by Proposition 3.17, $\mathfrak{M}' \nvDash \varphi(a)$, i.e., $\mathfrak{M}'$ satisfies $\mathbb{F}\,\varphi(a)$.

8. The branch is expanded by applying $\exists\mathbb{T}$ to $\mathbb{T}\,\exists x\,\psi(x) \in \Gamma$: Exercise.

9. The branch is expanded by applying $\exists\mathbb{F}$ to $\mathbb{F}\,\exists x\,\psi(x) \in \Gamma$: Exercise.

Now let's consider the possible inferences that result in a split branch.

1. The branch is expanded by applying $\land\mathbb{F}$ to $\mathbb{F}\,\psi \land \chi \in \Gamma$, which results in two branches, a left one continuing through $\mathbb{F}\,\psi$ and a right one through $\mathbb{F}\,\chi$. Suppose $\mathfrak{M} \vDash \Gamma$, in particular $\mathfrak{M} \nvDash \psi \land \chi$. Then $\mathfrak{M} \nvDash \psi$ or $\mathfrak{M} \nvDash \chi$. In the former case, $\mathfrak{M}$ satisfies $\mathbb{F}\,\psi$, i.e., $\mathfrak{M}$ satisfies the formulas on the left branch. In the latter, $\mathfrak{M}$ satisfies $\mathbb{F}\,\chi$, i.e., $\mathfrak{M}$ satisfies the formulas on the right branch.

2. The branch is expanded by applying $\lor\mathbb{T}$ to $\mathbb{T}\,\psi \lor \chi \in \Gamma$: Exercise.

3. The branch is expanded by applying $\to\mathbb{T}$ to $\mathbb{T}\,\psi \to \chi \in \Gamma$: Exercise.

4. The branch is expanded by Cut: This results in two branches, one containing $\mathbb{T}\,\psi$, the other containing $\mathbb{F}\,\psi$. Since $\mathfrak{M} \vDash \Gamma$ and either $\mathfrak{M} \vDash \psi$ or $\mathfrak{M} \nvDash \psi$, $\mathfrak{M}$ satisfies either the left or the right branch. □

Problem 8.8. Complete the proof of Theorem 8.28.

fol:tab:sou: cor:weak-soundness **Corollary 8.29.** *If $\vdash \varphi$ then φ is valid.*

fol:tab:sou: cor:entailment-soundness **Corollary 8.30.** *If $\Gamma \vdash \varphi$ then $\Gamma \vDash \varphi$.*

Proof. If $\Gamma \vdash \varphi$ then for some $\psi_1, \dots, \psi_n \in \Gamma$, $\{\mathbb{F}\,\varphi, \mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}$ has a closed tableau. By Theorem 8.28, every structure $\mathfrak{M}$ either makes some ψ_i false or makes φ true. Hence, if $\mathfrak{M} \vDash \Gamma$ then also $\mathfrak{M} \vDash \varphi$. □

fol:tab:sou: cor:consistency-soundness **Corollary 8.31.** *If Γ is satisfiable, then it is consistent.*

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then there are $\psi_1, \dots, \psi_n \in \Gamma$ and a closed tableau for $\{\mathbb{T}\,\psi_1, \dots, \mathbb{T}\,\psi_n\}$. By Theorem 8.28, there is no $\mathfrak{M}$ such that $\mathfrak{M} \vDash \psi_i$ for all $i = 1, \dots, n$. But then Γ is not satisfiable. □

8.12 Tableaux with Identity predicate

fol:tab:ide: sec Tableaux with identity predicate require additional inference rules. The rules for $=$ are (t, t_1, and t_2 are closed terms):

$$\frac{}{\mathbb{T}\, t = t} = \qquad \frac{\begin{array}{c}\mathbb{T}\, t_1 = t_2 \\ \mathbb{T}\, \varphi(t_1)\end{array}}{\mathbb{T}\, \varphi(t_2)} =\mathbb{T} \qquad \frac{\begin{array}{c}\mathbb{T}\, t_1 = t_2 \\ \mathbb{F}\, \varphi(t_1)\end{array}}{\mathbb{F}\, \varphi(t_2)} =\mathbb{F}$$

Note that in contrast to all the other rules, $=\mathbb{T}$ and $=\mathbb{F}$ require that *two* signed formulas already appear on the branch, namely both $\mathbb{T}\, t_1 = t_2$ and $S\, \varphi(t_1)$.

Example 8.32. If s and t are closed terms, then $s = t, \varphi(s) \vdash \varphi(t)$:

1.	$\mathbb{F}\, \varphi(t)$	Assumption
2.	$\mathbb{T}\, s = t$	Assumption
3.	$\mathbb{T}\, \varphi(s)$	Assumption
4.	$\mathbb{T}\, \varphi(t)$	$=\mathbb{T}\, 2, 3$
	$\otimes$	

This may be familiar as the principle of substitutability of identicals, or Leibniz' Law.

Tableaux prove that $=$ is symmetric, i.e., that $s_1 = s_2 \vdash s_2 = s_1$:

1.	$\mathbb{F}\, s_2 = s_1$	Assumption
2.	$\mathbb{T}\, s_1 = s_2$	Assumption
3.	$\mathbb{T}\, s_1 = s_1$	$=$
4.	$\mathbb{T}\, s_2 = s_1$	$=\mathbb{T}\, 2, 3$
	$\otimes$	

Here, line 2 is the first prerequisite formula $\mathbb{T}\, s_1 = s_2$ of $=\mathbb{T}$. Line 3 is the second one, of the form $\mathbb{T}\, \varphi(s_2)$—think of $\varphi(x)$ as $x = s_1$, then $\varphi(s_1)$ is $s_1 = s_1$ and $\varphi(s_2)$ is $s_2 = s_1$.

They also prove that $=$ is transitive, i.e., that $s_1 = s_2, s_2 = s_3 \vdash s_1 = s_3$:

1.	$\mathbb{F}\, s_1 = s_3$	Assumption
2.	$\mathbb{T}\, s_1 = s_2$	Assumption
3.	$\mathbb{T}\, s_2 = s_3$	Assumption
4.	$\mathbb{T}\, s_1 = s_3$	$=\mathbb{T}\, 3, 2$
	$\otimes$	

In this tableau, the first prerequisite formula of $=\mathbb{T}$ is line 3, $\mathbb{T}\, s_2 = s_3$ (s_2 plays the role of t_1, and s_3 the role of t_2). The second prerequisite, of the form $\mathbb{T}\, \varphi(s_2)$ is line 2. Here, think of $\varphi(x)$ as $s_1 = x$; that makes $\varphi(s_2)$ into $t_1 = t_2$ (i.e., line 2) and $\varphi(s_3)$ into the formula $s_1 = s_3$ in the conclusion.

Problem 8.9. Give closed tableaux for the following:

1. $\mathbb{F}\, \forall x\, \forall y\, ((x = y \land \varphi(x)) \to \varphi(y))$

2. $\mathbb{F}\,\exists x\,(\varphi(x) \land \forall y\,(\varphi(y) \to y = x))$,
   $\mathbb{T}\,\exists x\,\varphi(x) \land \forall y\,\forall z\,((\varphi(y) \land \varphi(z)) \to y = z)$

8.13 Soundness with Identity predicate

fol:tab:sid:sec

Proposition 8.33. *Tableaux with rules for identity are sound: no closed tableau is satisfiable.*

Proof. We just have to show as before that if a tableau has a satisfiable branch, the branch resulting from applying one of the rules for $=$ to it is also satisfiable. Let Γ be the set of signed formulas on the branch, and let $\mathfrak{M}$ be a structure satisfying Γ.

Suppose the branch is expanded using $=$, i.e., by adding the signed formula $\mathbb{T}\,t = t$. Trivially, $\mathfrak{M} \vDash t = t$, so $\mathfrak{M}$ also satisfies $\Gamma \cup \{\mathbb{T}\,t = t\}$.

If the branch is expanded using $=\mathbb{T}$, we add a signed formula $S\,\varphi(t_2)$, but Γ contains both $\mathbb{T}\,t_1 = t_2$ and $\mathbb{T}\,\varphi(t_1)$. Thus we have $\mathfrak{M} \vDash t_1 = t_2$ and $\mathfrak{M} \vDash \varphi(t_1)$. Let s be a variable assignment with $s(x) = \mathrm{Val}^{\mathfrak{M}}(t_1)$. By Proposition 3.17, $\mathfrak{M}, s \vDash \varphi(t_1)$. Since $s \sim_x s$, by Proposition 3.22, $\mathfrak{M}, s \vDash \varphi(x)$. since $\mathfrak{M} \vDash t_1 = t_2$, we have $\mathrm{Val}^{\mathfrak{M}}(t_1) = \mathrm{Val}^{\mathfrak{M}}(t_2)$, and hence $s(x) = \mathrm{Val}^{\mathfrak{M}}(t_2)$. By applying Proposition 3.22 again, we also have $\mathfrak{M}, s \vDash \varphi(t_2)$. By Proposition 3.17, $\mathfrak{M} \vDash \varphi(t_2)$. The case of $=\mathbb{F}$ is treated similarly. □

Chapter 9

Axiomatic Derivations

No effort has been made yet to ensure that the material in this chapter respects various tags indicating which connectives and quantifiers are primitive or defined: all are assumed to be primitive, except $\leftrightarrow$ which is assumed to be defined. If the FOL tag is true, we produce a version with quantifiers, otherwise without.

9.1 Rules and Derivations

explanation Axiomatic derivations are perhaps the simplest derivation system for logic. A derivation is just a sequence of formulas. To count as a derivation, every formula in the sequence must either be an instance of an axiom, or must follow from one or more formulas that precede it in the sequence by a rule of inference. A derivation derives its last formula. fol:axd:rul:sec

Definition 9.1 (Derivability). If Γ is a set of formulas of $\mathcal{L}$ then a *derivation* from Γ is a finite sequence $\varphi_1, \dots, \varphi_n$ of formulas where for each $i \leq n$ one of the following holds:

1. $\varphi_i \in \Gamma$; or
2. φ_i is an axiom; or
3. φ_i follows from some φ_j (and φ_k) with $j < i$ (and $k < i$) by a rule of inference.

What counts as a correct derivation depends on which inference rules we allow (and of course what we take to be axioms). And an inference rule is an if-then statement that tells us that, under certain conditions, a step A_i in a derivation is a correct inference step.

Definition 9.2 (Rule of inference). A *rule of inference* gives a sufficient condition for what counts as a correct inference step in a derivation from Γ.

For instance, since any one-element sequence φ with $\varphi \in \Gamma$ trivially counts as a derivation, the following might be a very simple rule of inference:

> If $\varphi \in \Gamma$, then φ is always a correct inference step in any derivation from Γ.

Similarly, if φ is one of the axioms, then φ by itself is a derivation, and so this is also a rule of inference:

> If φ is an axiom, then φ is a correct inference step.

It gets more interesting if the rule of inference appeals to formulas that appear before the step considered. The following rule is called *modus ponens:*

> If $\psi \to \varphi$ and ψ occur higher up in the derivation, then φ is a correct inference step.

If this is the only rule of inference, then our definition of derivation above amounts to this: $\varphi_1, \dots, \varphi_n$ is a derivation iff for each $i \leq n$ one of the following holds:

1. $\varphi_i \in \Gamma$; or
2. φ_i is an axiom; or
3. for some $j < i$, φ_j is $\psi \to \varphi_i$, and for some $k < i$, φ_k is ψ.

The last clause says that φ_i follows from φ_j (ψ) and φ_k ($\psi \to \varphi_i$) by modus ponens. If we can go from 1 to n, and each time we find a formula φ_i that is either in Γ, an axiom, or which a rule of inference tells us that it is a correct inference step, then the entire sequence counts as a correct derivation.

Definition 9.3 (Derivability). A formula φ is *derivable* from Γ, written $\Gamma \vdash \varphi$, if there is a derivation from Γ ending in φ.

Definition 9.4 (Theorems). A formula φ is a *theorem* if there is a derivation of φ from the empty set. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

9.2 Axiom and Rules for the Propositional Connectives

fol:axd:prp:sec

Definition 9.5 (Axioms). The set of Ax_0 of *axioms* for the propositional connectives comprises all formulas of the following forms:

$$(\varphi \wedge \psi) \to \varphi \tag{9.1}$$
fol:axd:prp:ax:land1

$$(\varphi \wedge \psi) \to \psi \tag{9.2}$$
fol:axd:prp:ax:land2

$$\varphi \to (\psi \to (\varphi \wedge \psi)) \tag{9.3}$$
fol:axd:prp:ax:land3

$$\varphi \to (\varphi \vee \psi) \tag{9.4}$$
fol:axd:prp:ax:lor1

$$\varphi \to (\psi \vee \varphi) \tag{9.5}$$
fol:axd:prp:ax:lor2

$$(\varphi \to \chi) \to ((\psi \to \chi) \to ((\varphi \vee \psi) \to \chi)) \tag{9.6}$$
fol:axd:prp:ax:lor3

$$\varphi \to (\psi \to \varphi) \tag{9.7}$$
fol:axd:prp:ax:lif1

$$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)) \tag{9.8}$$
fol:axd:prp:ax:lif2

$$(\varphi \to \psi) \to ((\varphi \to \neg\psi) \to \neg\varphi) \tag{9.9}$$
fol:axd:prp:ax:lnot1

$$\neg\varphi \to (\varphi \to \psi) \tag{9.10}$$
fol:axd:prp:ax:lnot2

$$\top \tag{9.11}$$
fol:axd:prp:ax:ltrue

$$\bot \to \varphi \tag{9.12}$$
fol:axd:prp:ax:lfalse1

$$(\varphi \to \bot) \to \neg\varphi \tag{9.13}$$
fol:axd:prp:ax:lfalse2

$$\neg\neg\varphi \to \varphi \tag{9.14}$$
fol:axd:prp:ax:dne

Definition 9.6 (Modus ponens). If ψ and $\psi \to \varphi$ already occur in a derivation, then φ is a correct inference step.

We'll abbreviate the rule modus ponens as "MP."

9.3 Axioms and Rules for Quantifiers

fol:axd:qua:sec

Definition 9.7 (Axioms for quantifiers). The *axioms* governing quantifiers are all instances of the following:

$$\forall x\, \psi \to \psi(t), \tag{9.15}$$
fol:axd:qua:ax:q1

$$\psi(t) \to \exists x\, \psi. \tag{9.16}$$
fol:axd:qua:ax:q2

for any closed term t.

Definition 9.8 (Rules for quantifiers).

If $\psi \to \varphi(a)$ already occurs in the derivation and a does not occur in Γ or ψ, then $\psi \to \forall x\, \varphi(x)$ is a correct inference step.

If $\varphi(a) \to \psi$ already occurs in the derivation and a does not occur in Γ or ψ, then $\exists x\, \varphi(x) \to \psi$ is a correct inference step.

We'll abbreviate either of these by "QR."

9.4 Examples of Derivations

fol:axd:pro:sec

Example 9.9. Suppose we want to prove $(\neg\theta \lor \alpha) \to (\theta \to \alpha)$. Clearly, this is not an instance of any of our axioms, so we have to use the MP rule to derive it. Our only rule is MP, which given φ and $\varphi \to \psi$ allows us to justify ψ. One strategy would be to use eq. (9.6) with φ being $\neg\theta$, ψ being α, and χ being $\theta \to \alpha$, i.e., the instance

$$(\neg\theta \to (\theta \to \alpha)) \to ((\alpha \to (\theta \to \alpha)) \to ((\neg\theta \lor \alpha) \to (\theta \to \alpha))).$$

Why? Two applications of MP yield the last part, which is what we want. And we easily see that $\neg\theta \to (\theta \to \alpha)$ is an instance of eq. (9.10), and $\alpha \to (\theta \to \alpha)$ is an instance of eq. (9.7). So our derivation is:

1.	$\neg\theta \to (\theta \to \alpha)$	eq. (9.10)
2.	$(\neg\theta \to (\theta \to \alpha)) \to$ $((\alpha \to (\theta \to \alpha)) \to ((\neg\theta \lor \alpha) \to (\theta \to \alpha)))$	eq. (9.6)
3.	$((\alpha \to (\theta \to \alpha)) \to ((\neg\theta \lor \alpha) \to (\theta \to \alpha))$	1, 2, MP
4.	$\alpha \to (\theta \to \alpha)$	eq. (9.7)
5.	$(\neg\theta \lor \alpha) \to (\theta \to \alpha)$	3, 4, MP

fol:axd:pro:ex:identity

Example 9.10. Let's try to find a derivation of $\theta \to \theta$. It is not an instance of an axiom, so we have to use MP to derive it. eq. (9.7) is an axiom of the form $\varphi \to \psi$ to which we could apply MP. To be useful, of course, the ψ which MP would justify as a correct step in this case would have to be $\theta \to \theta$, since this is what we want to derive. That means φ would also have to be θ, i.e., we might look at this instance of eq. (9.7):

$$\theta \to (\theta \to \theta)$$

In order to apply MP, we would also need to justify the corresponding second premise, namely φ. But in our case, that would be θ, and we won't be able to derive θ by itself. So we need a different strategy.

The other axiom involving just $\to$ is eq. (9.8), i.e.,

$$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$$

We could get to the last nested conditional by applying MP twice. Again, that would mean that we want an instance of eq. (9.8) where $\varphi \to \chi$ is $\theta \to \theta$, the formula we are aiming for. Then of course, φ and χ are both θ. How should we pick ψ so that both $\varphi \to (\psi \to \chi)$ and $\varphi \to \psi$, i.e., in our case $\theta \to (\psi \to \theta)$ and $\theta \to \psi$, are also derivable? Well, the first of these is already an instance of eq. (9.7), whatever we decide ψ to be. And $\theta \to \psi$ would be another instance of eq. (9.7) if ψ were $(\theta \to \theta)$. So, our derivation is:

1. $\theta \to ((\theta \to \theta) \to \theta)$ eq. (9.7)
2. $(\theta \to ((\theta \to \theta) \to \theta)) \to$
   $((\theta \to (\theta \to \theta)) \to (\theta \to \theta))$ eq. (9.8)
3. $(\theta \to (\theta \to \theta)) \to (\theta \to \theta)$ 1, 2, MP
4. $\theta \to (\theta \to \theta)$ eq. (9.7)
5. $\theta \to \theta$ 3, 4, MP

Example 9.11. Sometimes we want to show that there is a derivation of some formula from some other formulas Γ. For instance, let's show that we can derive $\varphi \to \chi$ from $\Gamma = \{\varphi \to \psi, \psi \to \chi\}$. fol:axd:pro: ex:chain

1. $\varphi \to \psi$ HYP
2. $\psi \to \chi$ HYP
3. $(\psi \to \chi) \to (\varphi \to (\psi \to \chi))$ eq. (9.7)
4. $\varphi \to (\psi \to \chi)$ 2, 3, MP
5. $(\varphi \to (\psi \to \chi)) \to$
   $((\varphi \to \psi) \to (\varphi \to \chi))$ eq. (9.8)
6. $((\varphi \to \psi) \to (\varphi \to \chi))$ 4, 5, MP
7. $\varphi \to \chi$ 1, 6, MP

The lines labelled "HYP" (for "hypothesis") indicate that the formula on that line is an element of Γ.

Proposition 9.12. *If $\Gamma \vdash \varphi \to \psi$ and $\Gamma \vdash \psi \to \chi$, then $\Gamma \vdash \varphi \to \chi$* fol:axd:pro: prop:chain

Proof. Suppose $\Gamma \vdash \varphi \to \psi$ and $\Gamma \vdash \psi \to \chi$. Then there is a derivation of $\varphi \to \psi$ from Γ; and a derivation of $\psi \to \chi$ from Γ as well. Combine these into a single derivation by concatenating them. Now add lines 3–7 of the derivation in the preceding example. This is a derivation of $\varphi \to \chi$—which is the last line of the new derivation—from Γ. Note that the justifications of lines 4 and 7 remain valid if the reference to line number 2 is replaced by reference to the last line of the derivation of $\varphi \to \psi$, and reference to line number 1 by reference to the last line of the derivation of $B \to \chi$. □

Problem 9.1. Show that the following hold by exhibiting derivations from the axioms:

1. $(\varphi \land \psi) \to (\psi \land \varphi)$
2. $((\varphi \land \psi) \to \chi) \to (\varphi \to (\psi \to \chi))$
3. $\neg(\varphi \lor \psi) \to \neg\varphi$

9.5 Derivations with Quantifiers

fol:axd:prq: sec

Example 9.13. Let us give a derivation of $(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \forall x\, (\varphi(x) \land \psi(x))$.

First, note that

$$(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \forall x\, \varphi(x)$$

is an instance of eq. (9.1), and

$$\forall x\, \varphi(x) \to \varphi(a)$$

of eq. (9.15). So, by Proposition 9.12, we know that

$$(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \varphi(a)$$

is derivable. Likewise, since

$$\begin{aligned} (\forall x\, \varphi(x) \land \forall y\, \psi(y)) &\to \forall y\, \psi(y) \qquad \text{and} \\ \forall y\, \psi(y) &\to \psi(a) \end{aligned}$$

are instances of eq. (9.2) and eq. (9.15), respectively,

$$(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \psi(a)$$

is derivable by Proposition 9.12. Using an appropriate instance of eq. (9.3) and two applications of MP, we see that

$$(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to (\varphi(a) \land \psi(a))$$

is derivable. We can now apply QR to obtain

$$(\forall x\, \varphi(x) \land \forall y\, \psi(y)) \to \forall x\, (\varphi(x) \land \psi(x)).$$

9.6 Proof-Theoretic Notions

fol:axd:ptn:sec Just as we've defined a number of important semantic notions (validity, entailment, satisfiabilty), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the derivability or non-derivability of certain formulas. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*. explanation

Definition 9.14 (Derivability). A formula φ is *derivable* from Γ, written $\Gamma \vdash \varphi$, if there is a derivation from Γ ending in φ.

Definition 9.15 (Theorems). A formula φ is a *theorem* if there is a derivation of φ from the empty set. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definition 9.16 (Consistency). A set Γ of formulas is *consistent* if and only if $\Gamma \nvdash \bot$; it is *inconsistent* otherwise.

Proposition 9.17 (Reflexivity). *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.* *fol:axd:ptn: prop:reflexivity*

Proof. The formula φ by itself is a derivation of φ from Γ. □

Proposition 9.18 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.* *fol:axd:ptn: prop:monotonicity*

Proof. Any derivation of φ from Γ is also a derivation of φ from Δ. □

Proposition 9.19 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.* *fol:axd:ptn: prop:transitivity*

Proof. Suppose $\{\varphi\} \cup \Delta \vdash \psi$. Then there is a derivation $\psi_1, \ldots, \psi_l = \psi$ from $\{\varphi\} \cup \Delta$. Some of the steps in that derivation will be correct because of a rule which refers to a prior line $\psi_i = \varphi$. By hypothesis, there is a derivation of φ from Γ, i.e., a derivation $\varphi_1, \ldots, \varphi_k = \varphi$ where every φ_i is an axiom, an element of Γ, or correct by a rule of inference. Now consider the sequence

$$\varphi_1, \ldots, \varphi_k = \varphi, \psi_1, \ldots, \psi_l = \psi.$$

This is a correct derivation of ψ from $\Gamma \cup \Delta$ since every $B_i = \varphi$ is now justified by the same rule which justifies $\varphi_k = \varphi$. □

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \ldots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i, then $\Gamma \vdash \psi$.

Proposition 9.20. *Γ is inconsistent iff $\Gamma \vdash \varphi$ for every φ.* *fol:axd:ptn: prop:incons*

Proof. Exercise. □

Problem 9.2. Prove Proposition 9.20.

Proposition 9.21 (Compactness). *fol:axd:ptn: prop:proves-compact*

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Proof.

1. If $\Gamma \vdash \varphi$, then there is a finite sequence of formulas $\varphi_1, \ldots, \varphi_n$ so that $\varphi \equiv \varphi_n$ and each φ_i is either a logical axiom, an element of Γ or follows from previous formulas by modus ponens. Take Γ_0 to be those φ_i which are in Γ. Then the derivation is likewise a derivation from Γ_0, and so $\Gamma_0 \vdash \varphi$.
2. This is the contrapositive of (1) for the special case $\varphi \equiv \bot$. □

9.7 The Deduction Theorem

fol:axd:ded: sec

As we've seen, giving derivations in an axiomatic system is cumbersome, and derivations may be hard to find. Rather than actually write out long lists of formulas, it is generally easier to argue that such derivations exist, by making use of a few simple results. We've already established three such results: Proposition 9.17 says we can always assert that $\Gamma \vdash \varphi$ when we know that $\varphi \in \Gamma$. Proposition 9.18 says that if $\Gamma \vdash \varphi$ then also $\Gamma \cup \{\psi\} \vdash \varphi$. And Proposition 9.19 implies that if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. Here's another simple result, a "meta"-version of modus ponens:

fol:axd:ded: prop:mp

Proposition 9.22. *If $\Gamma \vdash \varphi$ and $\Gamma \vdash \varphi \to \psi$, then $\Gamma \vdash \psi$.*

Proof. We have that $\{\varphi, \varphi \to \psi\} \vdash \psi$:

1.	φ	Hyp.
2.	$\varphi \to \psi$	Hyp.
3.	ψ	1, 2, MP

By Proposition 9.19, $\Gamma \vdash \psi$. □

The most important result we'll use in this context is the deduction theorem:

fol:axd:ded: thm:deduction-thm

Theorem 9.23 (Deduction Theorem). *$\Gamma \cup \{\varphi\} \vdash \psi$ if and only if $\Gamma \vdash \varphi \to \psi$.*

Proof. The "if" direction is immediate. If $\Gamma \vdash \varphi \to \psi$ then also $\Gamma \cup \{\varphi\} \vdash \varphi \to \psi$ by Proposition 9.18. Also, $\Gamma \cup \{\varphi\} \vdash \varphi$ by Proposition 9.17. So, by Proposition 9.22, $\Gamma \cup \{\varphi\} \vdash \psi$.

For the "only if" direction, we proceed by induction on the length of the derivation of ψ from $\Gamma \cup \{\varphi\}$.

For the induction basis, we prove the claim for every derivation of length 1. A derivation of ψ from $\Gamma \cup \{\varphi\}$ of length 1 consists of ψ by itself; and if it is correct ψ is either $\in \Gamma \cup \{\varphi\}$ or is an axiom. If $\psi \in \Gamma$ or is an axiom, then $\Gamma \vdash \psi$. We also have that $\Gamma \vdash \psi \to (\varphi \to \psi)$ by eq. (9.7), and Proposition 9.22 gives $\Gamma \vdash \varphi \to \psi$. If $\psi \in \{\varphi\}$ then $\Gamma \vdash \varphi \to \psi$ because then last sentence $\varphi \to \psi$ is the same as $\varphi \to \varphi$, and we have derived that in Example 9.10.

For the inductive step, suppose a derivation of ψ from $\Gamma \cup \{\varphi\}$ ends with a step ψ which is justified by modus ponens. (If it is not justified by modus ponens, $\psi \in \Gamma$, $\psi \equiv \varphi$, or ψ is an axiom, and the same reasoning as in the induction basis applies.) Then some previous steps in the derivation are $\chi \to \psi$ and χ, for some formula χ, i.e., $\Gamma \cup \{\varphi\} \vdash \chi \to \psi$ and $\Gamma \cup \{\varphi\} \vdash \chi$, and the respective derivations are shorter, so the inductive hypothesis applies to them. We thus have both:

$$\Gamma \vdash \varphi \to (\chi \to \psi);$$
$$\Gamma \vdash \varphi \to \chi.$$

But also

$$\Gamma \vdash (\varphi \to (\chi \to \psi)) \to ((\varphi \to \chi) \to (\varphi \to \psi)),$$

by eq. (9.8), and two applications of Proposition 9.22 give $\Gamma \vdash \varphi \to \psi$, as required. □

Notice how eq. (9.7) and eq. (9.8) were chosen precisely so that the Deduction Theorem would hold.

The following are some useful facts about derivability, which we leave as exercises.

Proposition 9.24. fol:axd:ded:prop:derivfacts

1. $\vdash (\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$; fol:axd:ded:derivfacts:a
2. *If* $\Gamma \cup \{\neg\varphi\} \vdash \neg\psi$ *then* $\Gamma \cup \{\psi\} \vdash \varphi$ *(Contraposition);* fol:axd:ded:derivfacts:b
3. $\{\varphi, \neg\varphi\} \vdash \psi$ *(Ex Falso Quodlibet, Explosion);* fol:axd:ded:derivfacts:c
4. $\{\neg\neg\varphi\} \vdash \varphi$ *(Double Negation Elimination);* fol:axd:ded:derivfacts:d
5. *If* $\Gamma \vdash \neg\neg\varphi$ *then* $\Gamma \vdash \varphi$*;* fol:axd:ded:derivfacts:e

Problem 9.3. Prove Proposition 9.24

9.8 The Deduction Theorem with Quantifiers

fol:axd:ddq:sec

Theorem 9.25 (Deduction Theorem). *If* $\Gamma \cup \{\varphi\} \vdash \psi$*, then* $\Gamma \vdash \varphi \to \psi$*.* fol:axd:ddq:thm:deduction-thm-q

Proof. We again proceed by induction on the length of the derivation of ψ from $\Gamma \cup \{\varphi\}$.

The proof of the induction basis is identical to that in the proof of Theorem 9.23.

For the inductive step, suppose again that the derivation of ψ from $\Gamma \cup \{\varphi\}$ ends with a step ψ which is justified by an inference rule. If the inference rule is modus ponens, we proceed as in the proof of Theorem 9.23. If the inference rule is QR, we know that $\psi \equiv \chi \to \forall x\, \theta(x)$ and a formula of the form $\chi \to \theta(a)$ appears earlier in the derivation, where a does not occur in χ, φ, or Γ. We thus have that

$$\Gamma \cup \{\varphi\} \vdash \chi \to \theta(a),$$

and the induction hypothesis applies, i.e., we have that

$$\Gamma \vdash \varphi \to (\chi \to \theta(a)).$$

By

$$\vdash (\varphi \to (\chi \to \theta(a))) \to ((\varphi \land \chi) \to \theta(a))$$

and modus ponens we get

$$\Gamma \vdash (\varphi \land \chi) \to \theta(a).$$

Since the eigenvariable condition still applies, we can add a step to this derivation justified by QR, and get

$$\Gamma \vdash (\varphi \land \chi) \to \forall x\, \theta(x).$$

We also have

$$\vdash ((\varphi \land \chi) \to \forall x\, \theta(x)) \to (\varphi \to (\chi \to \forall x\, \theta(x)),$$

so by modus ponens,

$$\Gamma \vdash \varphi \to (\chi \to \forall x\, \theta(x)),$$

i.e., $\Gamma \vdash \psi$.

We leave the case where ψ is justified by the rule QR, but is of the form $\exists x\, \theta(x) \to \chi$, as an exercise. □

Problem 9.4. Complete the proof of Theorem 9.25.

9.9 Derivability and Consistency

fol:axd:prv: sec

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem.

fol:axd:prv: prop:provability-contr

Proposition 9.26. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.*

Proof. If $\Gamma \cup \{\varphi\}$ is inconsistent, then $\Gamma \cup \{\varphi\} \vdash \bot$. By Proposition 9.17, $\Gamma \vdash \psi$ for every $\psi \in \Gamma$. Since also $\Gamma \vdash \varphi$ by hypothesis, $\Gamma \vdash \psi$ for every $\psi \in \Gamma \cup \{\varphi\}$. By Proposition 9.19, $\Gamma \vdash \bot$, i.e., Γ is inconsistent. □

fol:axd:prv: prop:prov-incons

Proposition 9.27. *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.*

Proof. First suppose $\Gamma \vdash \varphi$. Then $\Gamma \cup \{\neg\varphi\} \vdash \varphi$ by Proposition 9.18. $\Gamma \cup \{\neg\varphi\} \vdash \neg\varphi$ by Proposition 9.17. We also have $\vdash \neg\varphi \to (\varphi \to \bot)$ by eq. (9.10). So by two applications of Proposition 9.22, we have $\Gamma \cup \{\neg\varphi\} \vdash \bot$.

Now assume $\Gamma \cup \{\neg\varphi\}$ is inconsistent, i.e., $\Gamma \cup \{\neg\varphi\} \vdash \bot$. By the deduction theorem, $\Gamma \vdash \neg\varphi \to \bot$. $\Gamma \vdash (\neg\varphi \to \bot) \to \neg\neg\varphi$ by eq. (9.13), so $\Gamma \vdash \neg\neg\varphi$ by Proposition 9.22. Since $\Gamma \vdash \neg\neg\varphi \to \varphi$ (eq. (9.14)), we have $\Gamma \vdash \varphi$ by Proposition 9.22 again. □

Problem 9.5. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

Proposition 9.28. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.* fol:axd:prv: prop:explicit-inc

Proof. $\Gamma \vdash \neg\varphi \to (\varphi \to \bot)$ by eq. (9.10). $\Gamma \vdash \bot$ by two applications of Proposition 9.22. □

Proposition 9.29. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.* fol:axd:prv: prop:provability-exhaustive

Proof. Exercise. □

Problem 9.6. Prove Proposition 9.29

9.10 Derivability and the Propositional Connectives

explanation We establish that the derivability relation $\vdash$ of axiomatic deduction is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \to \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem. fol:axd:ppr: sec

Proposition 9.30. fol:axd:ppr: prop:provability-land

1. *Both $\varphi \wedge \psi \vdash \varphi$ and $\varphi \wedge \psi \vdash \psi$* fol:axd:ppr: prop:provability-land-left
2. *$\varphi, \psi \vdash \varphi \wedge \psi$.* fol:axd:ppr: prop:provability-land-right

Proof. 1. From eq. (9.1) and eq. (9.1) by modus ponens.

2. From eq. (9.3) by two applications of modus ponens. □

Proposition 9.31. fol:axd:ppr: prop:provability-lor

1. *$\varphi \vee \psi, \neg\varphi, \neg\psi$ is inconsistent.*
2. *Both $\varphi \vdash \varphi \vee \psi$ and $\psi \vdash \varphi \vee \psi$.*

Proof. 1. From eq. (9.9) we get $\vdash \neg\varphi \to (\varphi \to \bot)$ and $\vdash \neg\psi \to (\psi \to \bot)$. So by the deduction theorem, we have $\{\neg\varphi\} \vdash \varphi \to \bot$ and $\{\neg\psi\} \vdash \psi \to \bot$. From eq. (9.6) we get $\{\neg\varphi, \neg\psi\} \vdash (\varphi \vee \psi) \to \bot$. By the deduction theorem, $\{\varphi \vee \psi, \neg\varphi, \neg\psi\} \vdash \bot$.

2. From eq. (9.4) and eq. (9.5) by modus ponsens. □

Proposition 9.32. fol:axd:ppr: prop:provability-lif

1. *$\varphi, \varphi \to \psi \vdash \psi$.* fol:axd:ppr: prop:provability-lif-left
2. *Both $\neg\varphi \vdash \varphi \to \psi$ and $\psi \vdash \varphi \to \psi$.* fol:axd:ppr: prop:provability-lif-right

Proof. 1. We can derive:

1.	φ	Hyp
2.	$\varphi \to \psi$	Hyp
3.	ψ	1, 2, MP

2. By eq. (9.10) and eq. (9.7) and the deduction theorem, respectively. □

9.11 Derivability and the Quantifiers

fol:axd:qpr:sec The completeness theorem also requires that axiomatic deductions yield the facts about $\vdash$ established in this section. explanation

fol:axd:qpr:thm:strong-generalization **Theorem 9.33.** *If c is a constant symbol not occurring in Γ or $\varphi(x)$ and $\Gamma \vdash \varphi(c)$, then $\Gamma \vdash \forall x\, \varphi(x)$.*

Proof. By the deduction theorem, $\Gamma \vdash \top \to \varphi(c)$. Since c does not occur in Γ or $\top$, we get $\Gamma \vdash \top \to \varphi(c)$. By the deduction theorem again, $\Gamma \vdash \forall x\, \varphi(x)$. □

fol:axd:qpr:prop:provability-quantifiers **Proposition 9.34.**

1. $\varphi(t) \vdash \exists x\, \varphi(x)$.
2. $\forall x\, \varphi(x) \vdash \varphi(t)$.

Proof. 1. By eq. (9.16) and the deduction theorem.

2. By eq. (9.15) and the deduction theorem. □

9.12 Soundness

fol:axd:sou:sec A derivation system, such as axiomatic deduction, is *sound* if it cannot derive things that do not actually hold. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that explanation

1. every derivable φ is valid;
2. if φ is derivable from some others Γ, it is also a consequence of them;
3. if a set of formulas Γ is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Proposition 9.35. *If φ is an axiom, then $\mathfrak{M}, s \vDash \varphi$ for each structure $\mathfrak{M}$ and assignment s.*

Proof. We have to verify that all the axioms are valid. For instance, here is the case for eq. (9.15): suppose t is free for x in φ, and assume $\mathfrak{M}, s \vDash \forall x\, \varphi$. Then by definition of satisfaction, for each $s' \sim_x s$, also $\mathfrak{M}, s' \vDash \varphi$, and in particular this holds when $s'(x) = \mathrm{Val}_s^{\mathfrak{M}}(t)$. By Proposition 3.22, $\mathfrak{M}, s \vDash \varphi[t/x]$. This shows that $\mathfrak{M}, s \vDash (\forall x\, \varphi \rightarrow \varphi[t/x])$. □

Theorem 9.36 (Soundness). *If $\Gamma \vdash \varphi$ then $\Gamma \vDash \varphi$.* fol:axd:sou:thm:soundness

Proof. By induction on the length of the derivation of φ from Γ. If there are no steps justified by inferences, then all formulas in the derivation are either instances of axioms or are in Γ. By the previous proposition, all the axioms are valid, and hence if φ is an axiom then $\Gamma \vDash \varphi$. If $\varphi \in \Gamma$, then trivially $\Gamma \vDash \varphi$.

If the last step of the derivation of φ is justified by modus ponens, then there are formulas ψ and $\psi \rightarrow \varphi$ in the derivation, and the induction hypothesis applies to the part of the derivation ending in those formulas (since they contain at least one fewer steps justified by an inference). So, by induction hypothesis, $\Gamma \vDash \psi$ and $\Gamma \vDash \psi \rightarrow \varphi$. Then $\Gamma \vDash \varphi$ by Theorem 3.29.

Now suppose the last step is justified by QR. Then that step has the form $\chi \rightarrow \forall x\, B(x)$ and there is a preceding step $\chi \rightarrow \psi(c)$ with c not in Γ, χ, or $\forall x\, B(x)$. By induction hypothesis, $\Gamma \vDash \chi \rightarrow \forall x\, B(x)$. By Theorem 3.29, $\Gamma \cup \{\chi\} \vDash \psi(c)$.

Consider some structure $\mathfrak{M}$ such that $\mathfrak{M} \vDash \Gamma \cup \{\chi\}$. We need to show that $\mathfrak{M} \vDash \forall x\, \psi(x)$. Since $\forall x\, \psi(x)$ is a sentence, this means we have to show that for every variable assignment s, $\mathfrak{M}, s \vDash \psi(x)$ (Proposition 3.18). Since $\Gamma \cup \{\chi\}$ consists entirely of sentences, $\mathfrak{M}, s \vDash \theta$ for all $\theta \in \Gamma$ by Definition 3.11. Let $\mathfrak{M}'$ be like $\mathfrak{M}$ except that $c^{\mathfrak{M}'} = s(x)$. Since c does not occur in Γ or χ, $\mathfrak{M}' \vDash \Gamma \cup \{\chi\}$ by Corollary 3.20. Since $\Gamma \cup \{\chi\} \vDash \psi(c)$, $\mathfrak{M}' \vDash B(c)$. Since $\psi(c)$ is a sentence, $\mathfrak{M}, s \vDash \psi(c)$ by Proposition 3.17. $\mathfrak{M}', s \vDash \psi(x)$ iff $\mathfrak{M}' \vDash \psi(c)$ by Proposition 3.22 (recall that $\psi(c)$ is just $\psi(x)[c/x]$). So, $\mathfrak{M}', s \vDash \psi(x)$. Since c does not occur in $\psi(x)$, by Proposition 3.19, $\mathfrak{M}, s \vDash \psi(x)$. But s was an arbitrary variable assignment, so $\mathfrak{M} \vDash \forall x\, \psi(x)$. Thus $\Gamma \cup \{\chi\} \vDash \forall x\, \psi(x)$. By Theorem 3.29, $\Gamma \vDash \chi \rightarrow \forall x\, \psi(x)$.

The case where φ is justified by QR but is of the form $\exists x\, \psi(x) \rightarrow \chi$ is left as an exercise. □

Problem 9.7. Complete the proof of Theorem 9.36.

Corollary 9.37. *If $\vdash \varphi$, then φ is valid.* fol:axd:sou:cor:weak-soundness

Corollary 9.38. *If Γ is satisfiable, then it is consistent.* fol:axd:sou:cor:consistency-soundness

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then $\Gamma \vdash \bot$, i.e., there is a derivation of $\bot$ from Γ. By Theorem 9.36, any structure $\mathfrak{M}$ that satisfies Γ must satisfy $\bot$. Since $\mathfrak{M} \nvDash \bot$ for every structure $\mathfrak{M}$, no $\mathfrak{M}$ can satisfy Γ, i.e., Γ is not satisfiable. □

9.13 Derivations with Identity predicate

fol:axd:ide: sec In order to accommodate $=$ in derivations, we simply add new axiom schemas. The definition of derivation and $\vdash$ remains the same, we just also allow the new axioms.

Definition 9.39 (Axioms for identity predicate).

fol:axd:ide: ax:id1 fol:axd:ide: ax:id2

$$t = t, \tag{9.17}$$

$$t_1 = t_2 \rightarrow (\psi(t_1) \rightarrow \psi(t_2)), \tag{9.18}$$

for any closed terms t, t_1, t_2.

fol:axd:ide: prop:sound **Proposition 9.40.** *The axioms eq. (9.17) and eq. (9.18) are valid.*

Proof. Exercise. □

Problem 9.8. Prove Proposition 9.40.

fol:axd:ide: prop:iden1 **Proposition 9.41.** *$\Gamma \vdash t = t$, for any term t and set Γ.*

fol:axd:ide: prop:iden2 **Proposition 9.42.** *If $\Gamma \vdash \varphi(t_1)$ and $\Gamma \vdash t_1 = t_2$, then $\Gamma \vdash \varphi(t_2)$.*

Proof. The formula

$$(t_1 = t_2 \rightarrow (\varphi(t_1) \rightarrow \varphi(t_2)))$$

is an instance of eq. (9.18). The conclusion follows by two applications of MP. □

Chapter 10

The Completeness Theorem

10.1 Introduction

The completeness theorem is one of the most fundamental results about logic. fol:com:int:sec It comes in two formulations, the equivalence of which we'll prove. In its first formulation it says something fundamental about the relationship between semantic consequence and our derivation system: if a sentence φ follows from some sentences Γ, then there is also a derivation that establishes $\Gamma \vdash \varphi$. Thus, the derivation system is as strong as it can possibly be without proving things that don't actually follow.

In its second formulation, it can be stated as a model existence result: every consistent set of sentences is satisfiable. Consistency is a proof-theoretic notion: it says that our derivation system is unable to produce certain derivations. But who's to say that just because there are no derivations of a certain sort from Γ, it's guaranteed that there is a structure $\mathfrak{M}$? Before the completeness theorem was first proved—in fact before we had the derivation systems we now do—the great German mathematician David Hilbert held the view that consistency of mathematical theories guarantees the existence of the objects they are about. He put it as follows in a letter to Gottlob Frege:

> If the arbitrarily given axioms do not contradict one another with all their consequences, then they are true and the things defined by the axioms exist. This is for me the criterion of truth and existence.

Frege vehemently disagreed. The second formulation of the completeness theorem shows that Hilbert was right in at least the sense that if the axioms are consistent, then *some* structure exists that makes them all true.

These aren't the only reasons the completeness theorem—or rather, its proof—is important. It has a number of important consequences, some of which we'll discuss separately. For instance, since any derivation that shows $\Gamma \vdash \varphi$ is finite and so can only use finitely many of the sentences in Γ, it follows by the completeness theorem that if φ is a consequence of Γ, it is already a consequence of a finite subset of Γ. This is called *compactness*. Equivalently, if every finite subset of Γ is consistent, then Γ itself must be consistent.

Although the compactness theorem follows from the completeness theorem via the detour through derivations, it is also possible to use the *the proof of* the completeness theorem to establish it directly. For what the proof does is take a set of sentences with a certain property—consistency—and constructs a structure out of this set that has certain properties (in this case, that it satisfies the set). Almost the very same construction can be used to directly establish compactness, by starting from "finitely satisfiable" sets of sentences instead of consistent ones. The construction also yields other consequences, e.g., that any satisfiable set of sentences has a finite or denumerable model. (This result is called the Löwenheim-Skolem theorem.) In general, the construction of structures from sets of sentences is used often in logic, and sometimes even in philosophy.

10.2 Outline of the Proof

fol:com:out:sec The proof of the completeness theorem is a bit complex, and upon first reading it, it is easy to get lost. So let us outline the proof. The first step is a shift of perspective, that allows us to see a route to a proof. When completeness is thought of as "whenever $\Gamma \vDash \varphi$ then $\Gamma \vdash \varphi$," it may be hard to even come up with an idea: for to show that $\Gamma \vdash \varphi$ we have to find a derivation, and it does not look like the hypothesis that $\Gamma \vDash \varphi$ helps us for this in any way. For some proof systems it is possible to directly construct a derivation, but we will take a slightly different approach. The shift in perspective required is this: completeness can also be formulated as: "if Γ is consistent, it is satisfiable." Perhaps we can use the information in Γ together with the hypothesis that it is consistent to construct a structure that satisfies every sentence in Γ. After all, we know what kind of structure we are looking for: one that is as Γ describes it!

If Γ contains only atomic sentences, it is easy to construct a model for it. Suppose the atomic sentences are all of the form $P(a_1, \dots, a_n)$ where the a_i are constant symbols. All we have to do is come up with a domain $|\mathfrak{M}|$ and an assignment for P so that $\mathfrak{M} \vDash P(a_1, \dots, a_n)$. But that's not very hard: put $|\mathfrak{M}| = \mathbb{N}$, $c_i^{\mathfrak{M}} = i$, and for every $P(a_1, \dots, a_n) \in \Gamma$, put the tuple $\langle k_1, \dots, k_n \rangle$ into $P^{\mathfrak{M}}$, where k_i is the index of the constant symbol a_i (i.e., $a_i \equiv c_{k_i}$).

Now suppose Γ contains some formula $\lnot\psi$, with ψ atomic. We might worry that the construction of $\mathfrak{M}$ interferes with the possibility of making $\lnot\psi$ true. But here's where the consistency of Γ comes in: if $\lnot\psi \in \Gamma$, then $\psi \notin \Gamma$, or else Γ would be inconsistent. And if $\psi \notin \Gamma$, then according to our construction of $\mathfrak{M}$, $\mathfrak{M} \nvDash \psi$, so $\mathfrak{M} \vDash \lnot\psi$. So far so good.

What if Γ contains complex, non-atomic formulas? Say it contains $\varphi \land \psi$. To make that true, we should proceed as if both φ and ψ were in Γ. And if $\varphi \lor \psi \in \Gamma$, then we will have to make at least one of them true, i.e., proceed as if one of them was in Γ.

This suggests the following idea: we add additional formulas to Γ so as to (a) keep the resulting set consistent and (b) make sure that for every possible atomic sentence φ, either φ is in the resulting set, or $\neg\varphi$ is, and (c) such that, whenever $\varphi \land \psi$ is in the set, so are both φ and ψ, if $\varphi \lor \psi$ is in the set, at least one of φ or ψ is also, etc. We keep doing this (potentially forever). Call the set of all formulas so added Γ^*. Then our construction above would provide us with a structure $\mathfrak{M}$ for which we could prove, by induction, that it satisfies all sentences in Γ^*, and hence also all sentence in Γ since $\Gamma \subseteq \Gamma^*$. It turns out that guaranteeing (a) and (b) is enough. A set of sentences for which (b) holds is called *complete*. So our task will be to extend the consistent set Γ to a consistent and complete set Γ^*.

There is one wrinkle in this plan: if $\exists x\, \varphi(x) \in \Gamma$ we would hope to be able to pick some constant symbol c and add $\varphi(c)$ in this process. But how do we know we can always do that? Perhaps we only have a few constant symbols in our language, and for each one of them we have $\neg\varphi(c) \in \Gamma$. We can't also add $\varphi(c)$, since this would make the set inconsistent, and we wouldn't know whether $\mathfrak{M}$ has to make $\varphi(c)$ or $\neg\varphi(c)$ true. Moreover, it might happen that Γ contains only sentences in a language that has no constant symbols at all (e.g., the language of set theory).

The solution to this problem is to simply add infinitely many constants at the beginning, plus sentences that connect them with the quantifiers in the right way. (Of course, we have to verify that this cannot introduce an inconsistency.)

Our original construction works well if we only have constant symbols in the atomic sentences. But the language might also contain function symbols. In that case, it might be tricky to find the right functions on $\mathbb{N}$ to assign to these function symbols to make everything work. So here's another trick: instead of using i to interpret c_i, just take the set of constant symbols itself as the domain. Then $\mathfrak{M}$ can assign every constant symbol to itself: $c_i^{\mathfrak{M}} = c_i$. But why not go all the way: let $|\mathfrak{M}|$ be all *terms* of the language! If we do this, there is an obvious assignment of functions (that take terms as arguments and have terms as values) to function symbols: we assign to the function symbol f_i^n the function which, given n terms t_1, ..., t_n as input, produces the term $f_i^n(t_1, \dots, t_n)$ as value.

The last piece of the puzzle is what to do with $=$. The predicate symbol $=$ has a fixed interpretation: $\mathfrak{M} \vDash t = t'$ iff $\mathrm{Val}^{\mathfrak{M}}(t) = \mathrm{Val}^{\mathfrak{M}}(t')$. Now if we set things up so that the value of a term t is t itself, then this structure will make *no* sentence of the form $t = t'$ true unless t and t' are one and the same term. And of course this is a problem, since basically every interesting theory in a language with function symbols will have as theorems sentences $t = t'$ where t and t' are not the same term (e.g., in theories of arithmetic: $(\mathsf{o} + \mathsf{o}) = \mathsf{o}$). To solve this problem, we change the domain of $\mathfrak{M}$: instead of using terms as the objects in $|\mathfrak{M}|$, we use sets of terms, and each set is so that it contains all those terms which the sentences in Γ require to be equal. So, e.g., if Γ is a theory of arithmetic, one of these sets will contain: o, $(\mathsf{o} + \mathsf{o})$, $(\mathsf{o} \times \mathsf{o})$, etc. This will be the set we assign to o, and it will turn out that this set is also the value of all the terms in it, e.g., also of $(\mathsf{o} + \mathsf{o})$. Therefore, the sentence $(\mathsf{o} + \mathsf{o}) = \mathsf{o}$ will be true in this revised structure.

So here's what we'll do. First we investigate the properties of complete consistent sets, in particular we prove that a complete consistent set contains $\varphi \wedge \psi$ iff it contains both φ and ψ, $\varphi \vee \psi$ iff it contains at least one of them, etc. (Proposition 10.2). Then we define and investigate "saturated" sets of sentences. A saturated set is one which contains conditionals that link each quantified sentence to instances of it (Definition 10.5). We show that any consistent set Γ can always be extended to a saturated set Γ' (Lemma 10.6). If a set is consistent, saturated, and complete it also has the property that it contains $\exists x\, \varphi(x)$ iff it contains $\varphi(t)$ for some closed term t and $\forall x\, \varphi(x)$ iff it contains $\varphi(t)$ for all closed terms t (Proposition 10.7). We'll then take the saturated consistent set Γ' and show that it can be extended to a saturated, consistent, and complete set Γ^* (Lemma 10.8). This set Γ^* is what we'll use to define our term model $\mathfrak{M}(\Gamma^*)$. The term model has the set of closed terms as its domain, and the interpretation of its predicate symbols is given by the atomic sentences in Γ^* (Definition 10.9). We'll use the properties of saturated, complete consistent sets to show that indeed $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\varphi \in \Gamma^*$ (Lemma 10.12), and thus in particular, $\mathfrak{M}(\Gamma^*) \vDash \Gamma$. Finally, we'll consider how to define a term model if Γ contains $=$ as well (Definition 10.16) and show that it satisfies Γ^* (Lemma 10.19).

10.3 Complete Consistent Sets of Sentences

fol:com:ccs: sec

fol:com:ccs: def:complete-set

Definition 10.1 (Complete set). A set Γ of sentences is *complete* iff for any sentence φ, either $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$.

Complete sets of sentences leave no questions unanswered. For any sentence φ, Γ "says" if φ is true or false. The importance of complete sets extends beyond the proof of the completeness theorem. A theory which is complete and axiomatizable, for instance, is always decidable. explanation

Complete consistent sets are important in the completeness proof since we can guarantee that every consistent set of sentences Γ is contained in a complete consistent set Γ^*. A complete consistent set contains, for each sentence φ, either φ or its negation $\neg\varphi$, but not both. This is true in particular for atomic sentences, so from a complete consistent set in a language suitably expanded by constant symbols, we can construct a structure where the interpretation of predicate symbols is defined according to which atomic sentences are in Γ^*. This structure can then be shown to make all sentences in Γ^* (and hence also all those in Γ) true. The proof of this latter fact requires that $\neg\varphi \in \Gamma^*$ iff $\varphi \notin \Gamma^*$, $(\varphi \vee \psi) \in \Gamma^*$ iff $\varphi \in \Gamma^*$ or $\psi \in \Gamma^*$, etc. explanation

In what follows, we will often tacitly use the properties of reflexivity, monotonicity, and transitivity of $\vdash$ (see sections 6.8, 7.7, 8.7 and 9.6).

fol:com:ccs: prop:ccs

Proposition 10.2. *Suppose Γ is complete and consistent. Then:*

fol:com:ccs: prop:ccs-prov-in

1. *If $\Gamma \vdash \varphi$, then $\varphi \in \Gamma$.*

2. $\varphi \wedge \psi \in \Gamma$ *iff both* $\varphi \in \Gamma$ *and* $\psi \in \Gamma$. *fol:com:ccs: prop:ccs-and*

3. $\varphi \vee \psi \in \Gamma$ *iff either* $\varphi \in \Gamma$ *or* $\psi \in \Gamma$. *fol:com:ccs: prop:ccs-or*

4. $\varphi \rightarrow \psi \in \Gamma$ *iff either* $\varphi \notin \Gamma$ *or* $\psi \in \Gamma$. *fol:com:ccs: prop:ccs-if*

Proof. Let us suppose for all of the following that Γ is complete and consistent.

1. If $\Gamma \vdash \varphi$, then $\varphi \in \Gamma$.

   Suppose that $\Gamma \vdash \varphi$. Suppose to the contrary that $\varphi \notin \Gamma$. Since Γ is complete, $\neg\varphi \in \Gamma$. By Propositions 6.20, 7.20, 8.20 and 9.28, Γ is inconsistent. This contradicts the assumption that Γ is consistent. Hence, it cannot be the case that $\varphi \notin \Gamma$, so $\varphi \in \Gamma$.

2. $\varphi \wedge \psi \in \Gamma$ iff both $\varphi \in \Gamma$ and $\psi \in \Gamma$:

   For the forward direction, suppose $\varphi \wedge \psi \in \Gamma$. Then by Propositions 6.22, 7.22, 8.22 and 9.30, item (1), $\Gamma \vdash \varphi$ and $\Gamma \vdash \psi$. By (1), $\varphi \in \Gamma$ and $\psi \in \Gamma$, as required.

   For the reverse direction, let $\varphi \in \Gamma$ and $\psi \in \Gamma$. By Propositions 6.22, 7.22, 8.22 and 9.30, item (2), $\Gamma \vdash \varphi \wedge \psi$. By (1), $\varphi \wedge \psi \in \Gamma$.

3. First we show that if $\varphi \vee \psi \in \Gamma$, then either $\varphi \in \Gamma$ or $\psi \in \Gamma$. Suppose $\varphi \vee \psi \in \Gamma$ but $\varphi \notin \Gamma$ and $\psi \notin \Gamma$. Since Γ is complete, $\neg\varphi \in \Gamma$ and $\neg\psi \in \Gamma$. By Propositions 6.23, 7.23, 8.23 and 9.31, item (1), Γ is inconsistent, a contradiction. Hence, either $\varphi \in \Gamma$ or $\psi \in \Gamma$.

   For the reverse direction, suppose that $\varphi \in \Gamma$ or $\psi \in \Gamma$. By Propositions 6.23, 7.23, 8.23 and 9.31, item (2), $\Gamma \vdash \varphi \vee \psi$. By (1), $\varphi \vee \psi \in \Gamma$, as required.

4. For the forward direction, suppose $\varphi \rightarrow \psi \in \Gamma$, and suppose to the contrary that $\varphi \in \Gamma$ and $\psi \notin \Gamma$. On these assumptions, $\varphi \rightarrow \psi \in \Gamma$ and $\varphi \in \Gamma$. By Propositions 6.24, 7.24, 8.24 and 9.32, item (1), $\Gamma \vdash \psi$. But then by (1), $\psi \in \Gamma$, contradicting the assumption that $\psi \notin \Gamma$.

   For the reverse direction, first consider the case where $\varphi \notin \Gamma$. Since Γ is complete, $\neg\varphi \in \Gamma$. By Propositions 6.24, 7.24, 8.24 and 9.32, item (2), $\Gamma \vdash \varphi \rightarrow \psi$. Again by (1), we get that $\varphi \rightarrow \psi \in \Gamma$, as required.

   Now consider the case where $\psi \in \Gamma$. By Propositions 6.24, 7.24, 8.24 and 9.32, item (2) again, $\Gamma \vdash \varphi \rightarrow \psi$. By (1), $\varphi \rightarrow \psi \in \Gamma$. □

Problem 10.1. Complete the proof of Proposition 10.2.

10.4 Henkin Expansion

fol:com:hen:sec

Part of the challenge in proving the completeness theorem is that the model we construct from a complete consistent set Γ must make all the quantified formulas in Γ true. In order to guarantee this, we use a trick due to Leon Henkin. In essence, the trick consists in expanding the language by infinitely many constant symbols and adding, for each formula with one free variable $\varphi(x)$ a formula of the form $\exists x\, \varphi(x) \rightarrow \varphi(c)$, where c is one of the new constant symbols. When we construct the structure satisfying Γ, this will guarantee that each true existential sentence has a witness among the new constants. explanation

fol:com:hen:prop:lang-exp

Proposition 10.3. *If Γ is consistent in $\mathcal{L}$ and $\mathcal{L}'$ is obtained from $\mathcal{L}$ by adding a denumerable set of new constant symbols d_0, d_1, ..., then Γ is consistent in $\mathcal{L}'$.*

Definition 10.4 (Saturated set). A set Γ of formulas of a language $\mathcal{L}$ is *saturated* iff for each formula $\varphi(x) \in \mathrm{Frm}(\mathcal{L})$ with one free variable x there is a constant symbol $c \in \mathcal{L}$ such that $\exists x\, \varphi(x) \rightarrow \varphi(c) \in \Gamma$.

The following definition will be used in the proof of the next theorem.

fol:com:hen:defn:henkin-exp

Definition 10.5. Let $\mathcal{L}'$ be as in Proposition 10.3. Fix an enumeration $\varphi_0(x_0)$, $\varphi_1(x_1)$, ... of all formulas $\varphi_i(x_i)$ of $\mathcal{L}'$ in which one variable (x_i) occurs free. We define the sentences θ_n by induction on n.

Let c_0 be the first constant symbol among the d_i we added to $\mathcal{L}$ which does not occur in $\varphi_0(x_0)$. Assuming that θ_0, ..., θ_{n-1} have already been defined, let c_n be the first among the new constant symbols d_i that occurs neither in θ_0, ..., θ_{n-1} nor in $\varphi_n(x_n)$.

Now let θ_n be the formula $\exists x_n\, \varphi_n(x_n) \rightarrow \varphi_n(c_n)$.

fol:com:hen:lem:henkin

Lemma 10.6. *Every consistent set Γ can be extended to a saturated consistent set Γ'.*

Proof. Given a consistent set of sentences Γ in a language $\mathcal{L}$, expand the language by adding a denumerable set of new constant symbols to form $\mathcal{L}'$. By Proposition 10.3, Γ is still consistent in the richer language. Further, let θ_i be as in Definition 10.5. Let

$$\Gamma_0 = \Gamma$$
$$\Gamma_{n+1} = \Gamma_n \cup \{\theta_n\}$$

i.e., $\Gamma_{n+1} = \Gamma \cup \{\theta_0, \dots, \theta_n\}$, and let $\Gamma' = \bigcup_n \Gamma_n$. Γ' is clearly saturated.

If Γ' were inconsistent, then for some n, Γ_n would be inconsistent (Exercise: explain why). So to show that Γ' is consistent it suffices to show, by induction on n, that each set Γ_n is consistent.

The induction basis is simply the claim that $\Gamma_0 = \Gamma$ is consistent, which is the hypothesis of the theorem. For the induction step, suppose that Γ_n is consistent but $\Gamma_{n+1} = \Gamma_n \cup \{\theta_n\}$ is inconsistent. Recall that θ_n is $\exists x_n\, \varphi_n(x_n) \to \varphi_n(c_n)$, where $\varphi_n(x_n)$ is a formula of $\mathcal{L}'$ with only the variable x_n free. By the way we've chosen the c_n (see Definition 10.5), c_n does not occur in $\varphi_n(x_n)$ nor in Γ_n.

If $\Gamma_n \cup \{\theta_n\}$ is inconsistent, then $\Gamma_n \vdash \neg\theta_n$, and hence both of the following hold:

$$\Gamma_n \vdash \exists x_n\, \varphi_n(x_n) \qquad \Gamma_n \vdash \neg\varphi_n(c_n)$$

Since c_n does not occur in Γ_n or in $\varphi_n(x_n)$, Theorems 6.25, 7.25, 8.25 and 9.33 applies. From $\Gamma_n \vdash \neg\varphi_n(c_n)$, we obtain $\Gamma_n \vdash \forall x_n\, \neg\varphi_n(x_n)$. Thus we have that both $\Gamma_n \vdash \exists x_n\, \varphi_n(x_n)$ and $\Gamma_n \vdash \forall x_n\, \neg\varphi_n(x_n)$, so Γ_n itself is inconsistent. (Note that $\forall x_n\, \neg\varphi_n(x_n) \vdash \neg\exists x_n\, \varphi_n(x_n)$.) Contradiction: Γ_n was supposed to be consistent. Hence $\Gamma_n \cup \{\theta_n\}$ is consistent. □

explanation We'll now show that *complete*, consistent sets which are saturated have the property that it contains a universally quantified sentence iff it contains all its instances and it contains an existentially quantified sentence iff it contains at least one instance. We'll use this to show that the structure we'll generate from a complete, consistent, saturated set makes all its quantified sentences true.

Proposition 10.7. *Suppose Γ is complete, consistent, and saturated.* *fol:com:hen: prop:saturated-instances*

1. *$\exists x\, \varphi(x) \in \Gamma$ iff $\varphi(t) \in \Gamma$ for at least one closed term t.*
2. *$\forall x\, \varphi(x) \in \Gamma$ iff $\varphi(t) \in \Gamma$ for all closed terms t.*

Proof. 1. First suppose that $\exists x\, \varphi(x) \in \Gamma$. Because Γ is saturated, $(\exists x\, \varphi(x) \to \varphi(c)) \in \Gamma$ for some constant symbol c. By Propositions 6.24, 7.24, 8.24 and 9.32, item (1), and Proposition 10.2(1), $\varphi(c) \in \Gamma$.

   For the other direction, saturation is not necessary: Suppose $\varphi(t) \in \Gamma$. Then $\Gamma \vdash \exists x\, \varphi(x)$ by Propositions 6.26, 7.26, 8.26 and 9.34, item (1). By Proposition 10.2(1), $\exists x\, \varphi(x) \in \Gamma$.

2. Suppose that $\varphi(t) \in \Gamma$ for all closed terms t. By way of contradiction, assume $\forall x\, \varphi(x) \notin \Gamma$. Since Γ is complete, $\neg\forall x\, \varphi(x) \in \Gamma$. By saturation, $(\exists x\, \neg\varphi(x) \to \neg\varphi(c)) \in \Gamma$ for some constant symbol c. By assumption, since c is a closed term, $\varphi(c) \in \Gamma$. But this would make Γ inconsistent. (Exercise: give the derivation that shows

$$\neg\forall x\, \varphi(x), \exists x\, \neg\varphi(x) \to \neg\varphi(c), \varphi(c)$$

   is inconsistent.)

   For the reverse direction, we do not need saturation: Suppose $\forall x\, \varphi(x) \in \Gamma$. Then $\Gamma \vdash \varphi(t)$ by Propositions 6.26, 7.26, 8.26 and 9.34, item (2). We get $\varphi(t) \in \Gamma$ by Proposition 10.2. □

10.5 Lindenbaum's Lemma

fol:com:lin: sec

We now prove a lemma that shows that any consistent set of sentences is contained in some set of sentences which is not just consistent, but also complete. The proof works by adding one sentence at a time, guaranteeing at each step that the set remains consistent. We do this so that for every φ, either φ or $\neg\varphi$ gets added at some stage. The union of all stages in that construction then contains either φ or its negation $\neg\varphi$ and is thus complete. It is also consistent, since we made sure at each stage not to introduce an inconsistency. explanation

fol:com:lin: lem:lindenbaum

Lemma 10.8 (Lindenbaum's Lemma). *Every consistent set Γ in a language $\mathcal{L}$ can be extended to a complete and consistent set Γ^*.*

Proof. Let Γ be consistent. Let φ_0, φ_1, ... be an enumeration of all the sentences of $\mathcal{L}$. Define $\Gamma_0 = \Gamma$, and

$$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{\varphi_n\} & \text{if } \Gamma_n \cup \{\varphi_n\} \text{ is consistent;} \\ \Gamma_n \cup \{\neg\varphi_n\} & \text{otherwise.} \end{cases}$$

Let $\Gamma^* = \bigcup_{n \geq 0} \Gamma_n$.

Each Γ_n is consistent: Γ_0 is consistent by definition. If $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_n\}$, this is because the latter is consistent. If it isn't, $\Gamma_{n+1} = \Gamma_n \cup \{\neg\varphi_n\}$. We have to verify that $\Gamma_n \cup \{\neg\varphi_n\}$ is consistent. Suppose it's not. Then *both* $\Gamma_n \cup \{\varphi_n\}$ and $\Gamma_n \cup \{\neg\varphi_n\}$ are inconsistent. This means that Γ_n would be inconsistent by Propositions 6.21, 7.21, 8.21 and 9.29, contrary to the induction hypothesis.

For every n and every $i < n$, $\Gamma_i \subseteq \Gamma_n$. This follows by a simple induction on n. For $n = 0$, there are no $i < 0$, so the claim holds automatically. For the inductive step, suppose it is true for n. We have $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_n\}$ or $= \Gamma_n \cup \{\neg\varphi_n\}$ by construction. So $\Gamma_n \subseteq \Gamma_{n+1}$. If $i < n$, then $\Gamma_i \subseteq \Gamma_n$ by inductive hypothesis, and so $\subseteq \Gamma_{n+1}$ by transitivity of $\subseteq$.

From this it follows that every finite subset of Γ^* is a subset of Γ_n for some n, since each $\psi \in \Gamma^*$ not already in Γ_0 is added at some stage i. If n is the last one of these, then all ψ in the finite subset are in Γ_n. So, every finite subset of Γ^* is consistent. By Propositions 6.17, 7.17, 8.17 and 9.21, Γ^* is consistent.

Every sentence of $\mathrm{Frm}(\mathcal{L})$ appears on the list used to define Γ^*. If $\varphi_n \notin \Gamma^*$, then that is because $\Gamma_n \cup \{\varphi_n\}$ was inconsistent. But then $\neg\varphi_n \in \Gamma^*$, so Γ^* is complete. □

10.6 Construction of a Model

fol:com:mod: sec

Right now we are not concerned about $=$, i.e., we only want to show that a consistent set Γ of sentences not containing $=$ is satisfiable. We first extend Γ to a consistent, complete, and saturated set Γ^*. In this case, the definition of a model $\mathfrak{M}(\Gamma^*)$ is simple: We take the set of closed terms of $\mathcal{L}'$ as the domain. We assign every constant symbol to itself, and make sure that more generally, explanation for every closed term t, $\mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(t) = t$. The predicate symbols are assigned extensions in such a way that an atomic sentence is true in $\mathfrak{M}(\Gamma^*)$ iff it is in Γ^*. This will obviously make all the atomic sentences in Γ^* true in $\mathfrak{M}(\Gamma^*)$. The rest are true provided the Γ^* we start with is consistent, complete, and saturated.

Definition 10.9 (Term model). Let Γ^* be a complete and consistent, saturated set of sentences in a language $\mathcal{L}$. The *term model* $\mathfrak{M}(\Gamma^*)$ of Γ^* is the structure defined as follows: fol:com:mod:defn:termmodel

1. The domain $|\mathfrak{M}(\Gamma^*)|$ is the set of all closed terms of $\mathcal{L}$.
2. The interpretation of a constant symbol c is c itself: $c^{\mathfrak{M}(\Gamma^*)} = c$.
3. The function symbol f is assigned the function which, given as arguments the closed terms $t_1, \ldots, t_n$, has as value the closed term $f(t_1, \ldots, t_n)$:
$$f^{\mathfrak{M}(\Gamma^*)}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$$
4. If R is an n-place predicate symbol, then
$$\langle t_1, \ldots, t_n \rangle \in R^{\mathfrak{M}(\Gamma^*)} \text{ iff } R(t_1, \ldots, t_n) \in \Gamma^*.$$

We will now check that we indeed have $\mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(t) = t$.

Lemma 10.10. *Let $\mathfrak{M}(\Gamma^*)$ be the term model of Definition 10.9, then $\mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(t) = t$.* *fol:com:mod:lem:val-in-termmodel*

Proof. The proof is by induction on t, where the base case, when t is a constant symbol, follows directly from the definition of the term model. For the induction step assume $t_1, \ldots, t_n$ are closed terms such that $\mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(t_i) = t_i$ and that f is an n-ary function symbol. Then

$$\begin{aligned}
\mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(f(t_1, \ldots, t_n)) &= f^{\mathfrak{M}(\Gamma^*)}(\mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(t_1), \ldots, \mathrm{Val}^{\mathfrak{M}(\Gamma^*)}(t_n)) \\
&= f^{\mathfrak{M}(\Gamma^*)}(t_1, \ldots, t_n) \\
&= f(t_1, \ldots, t_n),
\end{aligned}$$

and so by induction this holds for every closed term t. □

explanation A structure $\mathfrak{M}$ may make an existentially quantified sentence $\exists x\, \varphi(x)$ true without there being an instance $\varphi(t)$ that it makes true. A structure $\mathfrak{M}$ may make all instances $\varphi(t)$ of a universally quantified sentence $\forall x\, \varphi(x)$ true, without making $\forall x\, \varphi(x)$ true. This is because in general not every element of $|\mathfrak{M}|$ is the value of a closed term ($\mathfrak{M}$ may not be covered). This is the reason the satisfaction relation is defined via variable assignments. However, for our term model $\mathfrak{M}(\Gamma^*)$ this wouldn't be necessary—because it is covered. This is the content of the next result.

fol:com:mod: prop:quant-termmodel **Proposition 10.11.** *Let $\mathfrak{M}(\Gamma^*)$ be the term model of Definition 10.9.*

1. *$\mathfrak{M}(\Gamma^*) \vDash \exists x\, \varphi(x)$ iff $\mathfrak{M}(\Gamma^*) \vDash \varphi(t)$ for at least one term t.*

2. *$\mathfrak{M}(\Gamma^*) \vDash \forall x\, \varphi(x)$ iff $\mathfrak{M}(\Gamma^*) \vDash \varphi(t)$ for all terms t.*

Proof. 1. By Proposition 3.18, $\mathfrak{M}(\Gamma^*) \vDash \exists x\, \varphi(x)$ iff for at least one variable assignment s, $\mathfrak{M}(\Gamma^*), s \vDash \varphi(x)$. As $|\mathfrak{M}(\Gamma^*)|$ consists of the closed terms of $\mathcal{L}$, this is the case iff there is at least one closed term t such that $s(x) = t$ and $\mathfrak{M}(\Gamma^*), s \vDash \varphi(x)$. By Proposition 3.22, $\mathfrak{M}(\Gamma^*), s \vDash \varphi(x)$ iff $\mathfrak{M}(\Gamma^*), s \vDash \varphi(t)$, where $s(x) = t$. By Proposition 3.17, $\mathfrak{M}(\Gamma^*), s \vDash \varphi(t)$ iff $\mathfrak{M}(\Gamma^*) \vDash \varphi(t)$, since $\varphi(t)$ is a sentence.

2. By Proposition 3.18, $\mathfrak{M}(\Gamma^*) \vDash \forall x\, \varphi(x)$ iff for every variable assignment s, $\mathfrak{M}(\Gamma^*), s \vDash \varphi(x)$. Recall that $|\mathfrak{M}(\Gamma^*)|$ consists of the closed terms of $\mathcal{L}$, so for every closed term t, $s(x) = t$ is such a variable assignment, and for any variable assignment, $s(x)$ is some closed term t. By Proposition 3.22, $\mathfrak{M}(\Gamma^*), s \vDash \varphi(x)$ iff $\mathfrak{M}(\Gamma^*), s \vDash \varphi(t)$, where $s(x) = t$. By Proposition 3.17, $\mathfrak{M}(\Gamma^*), s \vDash \varphi(t)$ iff $\mathfrak{M}(\Gamma^*) \vDash \varphi(t)$, since $\varphi(t)$ is a sentence. □

fol:com:mod: lem:truth **Lemma 10.12 (Truth Lemma).** *Suppose φ does not contain $=$. Then $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\varphi \in \Gamma^*$.*

Proof. We prove both directions simultaneously, and by induction on φ.

1. $\varphi \equiv \bot$: $\mathfrak{M}(\Gamma^*) \nvDash \bot$ by definition of satisfaction. On the other hand, $\bot \notin \Gamma^*$ since Γ^* is consistent.

2. $\varphi \equiv \top$: $\mathfrak{M}(\Gamma^*) \vDash \top$ by definition of satisfaction. On the other hand, $\top \in \Gamma^*$ since Γ^* is consistent and complete, and $\Gamma^* \vdash \top$.

3. $\varphi \equiv R(t_1, \dots, t_n)$: $\mathfrak{M}(\Gamma^*) \vDash R(t_1, \dots, t_n)$ iff $\langle t_1, \dots, t_n \rangle \in R^{\mathfrak{M}(\Gamma^*)}$ (by the definition of satisfaction) iff $R(t_1, \dots, t_n) \in \Gamma^*$ (by the construction of $\mathfrak{M}(\Gamma^*)$).

4. $\varphi \equiv \neg\psi$: $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{M}(\Gamma^*) \nvDash \psi$ (by definition of satisfaction). By induction hypothesis, $\mathfrak{M}(\Gamma^*) \nvDash \psi$ iff $\psi \notin \Gamma^*$. Since Γ^* is consistent and complete, $\psi \notin \Gamma^*$ iff $\neg\psi \in \Gamma^*$.

5. $\varphi \equiv \psi \wedge \chi$: $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff we have both $\mathfrak{M}(\Gamma^*) \vDash \psi$ and $\mathfrak{M}(\Gamma^*) \vDash \chi$ (by definition of satisfaction) iff both $\psi \in \Gamma^*$ and $\chi \in \Gamma^*$ (by the induction hypothesis). By Proposition 10.2(2), this is the case iff $(\psi \wedge \chi) \in \Gamma^*$.

6. $\varphi \equiv \psi \vee \chi$: $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{M}(\Gamma^*) \vDash \psi$ or $\mathfrak{M}(\Gamma^*) \vDash \chi$ (by definition of satisfaction) iff $\psi \in \Gamma^*$ or $\chi \in \Gamma^*$ (by induction hypothesis). This is the case iff $(\psi \vee \chi) \in \Gamma^*$ (by Proposition 10.2(3)).

7. $\varphi \equiv \psi \to \chi$: $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{M}(\Gamma^*) \nvDash \psi$ or $\mathfrak{M}(\Gamma^*) \vDash \chi$ (by definition of satisfaction) iff $\psi \notin \Gamma^*$ or $\chi \in \Gamma^*$ (by induction hypothesis). This is the case iff $(\psi \to \chi) \in \Gamma^*$ (by Proposition 10.2(4)).

8. $\varphi \equiv \forall x\, \psi(x)$: $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{M}(\Gamma^*) \vDash \psi(t)$ for all terms t (Proposition 10.11). By induction hypothesis, this is the case iff $\psi(t) \in \Gamma^*$ for all terms t, by Proposition 10.7, this in turn is the case iff $\forall x\, \varphi(x) \in \Gamma^*$.

9. $\varphi \equiv \exists x\, \psi(x)$: $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\mathfrak{M}(\Gamma^*) \vDash \psi(t)$ for at least one term t (Proposition 10.11). By induction hypothesis, this is the case iff $\psi(t) \in \Gamma^*$ for at least one term t. By Proposition 10.7, this in turn is the case iff $\exists x\, \psi(x) \in \Gamma^*$. □

10.7 Identity

explanation The construction of the term model given in the preceding section is enough to establish completeness for first-order logic for sets Γ that do not contain $=$. The term model satisfies every $\varphi \in \Gamma^*$ which does not contain $=$ (and hence all $\varphi \in \Gamma$). It does not work, however, if $=$ is present. The reason is that Γ^* then may contain a sentence $t = t'$, but in the term model the value of any term is that term itself. Hence, if t and t' are different terms, their values in the term model—i.e., t and t', respectively—are different, and so $t = t'$ is false. We can fix this, however, using a construction known as "factoring." fol:com:ide:sec

Definition 10.13. Let Γ^* be a consistent and complete set of sentences in $\mathcal{L}$. We define the relation $\approx$ on the set of closed terms of $\mathcal{L}$ by

$$t \approx t' \quad \text{iff} \quad t = t' \in \Gamma^*$$

Proposition 10.14. *The relation $\approx$ has the following properties:* *fol:com:ide:prop:approx-equiv*

1. *$\approx$ is reflexive.*

2. *$\approx$ is symmetric.*

3. *$\approx$ is transitive.*

4. *If $t \approx t'$, f is a function symbol, and $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$ are terms, then*

$$f(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n) \approx f(t_1, \ldots, t_{i-1}, t', t_{i+1}, \ldots, t_n).$$

5. *If $t \approx t'$, R is a predicate symbol, and $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$ are terms, then*

$$R(t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_n) \in \Gamma^* \text{ iff } R(t_1, \ldots, t_{i-1}, t', t_{i+1}, \ldots, t_n) \in \Gamma^*.$$

Proof. Since Γ^* is consistent and complete, $t = t' \in \Gamma^*$ iff $\Gamma^* \vdash t = t'$. Thus it is enough to show the following:

1. $\Gamma^* \vdash t = t$ for all terms t.

2. If $\Gamma^* \vdash t = t'$ then $\Gamma^* \vdash t' = t$.

3. If $\Gamma^* \vdash t = t'$ and $\Gamma^* \vdash t' = t''$, then $\Gamma^* \vdash t = t''$.

4. If $\Gamma^* \vdash t = t'$, then

$$\Gamma^* \vdash f(t_1, \dots, t_{i-1}, t, t_{i+1}, , \dots, t_n) = f(t_1, \dots, t_{i-1}, t', t_{i+1}, \dots, t_n)$$

   for every n-place function symbol f and terms $t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n$.

5. If $\Gamma^* \vdash t = t'$ and $\Gamma^* \vdash R(t_1, \dots, t_{i-1}, t, t_{i+1}, \dots, t_n)$, then $\Gamma^* \vdash R(t_1, \dots, t_{i-1}, t', t_{i+1}, \dots, t_n)$ for every n-place predicate symbol R and terms $t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n$.
   □

Problem 10.2. Complete the proof of Proposition 10.14.

Definition 10.15. Suppose Γ^* is a consistent and complete set in a language $\mathcal{L}$, t is a term, and $\approx$ as in the previous definition. Then:

$$[t]_\approx = \{t' : t' \in \mathrm{Trm}(\mathcal{L}), t \approx t'\}$$

and $\mathrm{Trm}(\mathcal{L})/_\approx = \{[t]_\approx : t \in \mathrm{Trm}(\mathcal{L})\}$.

fol:com:ide:
defn:term-model-factor **Definition 10.16.** Let $\mathfrak{M} = \mathfrak{M}(\Gamma^*)$ be the term model for Γ^* from Definition 10.9. Then $\mathfrak{M}/_\approx$ is the following structure:

1. $|\mathfrak{M}/_\approx| = \mathrm{Trm}(\mathcal{L})/_\approx$.

2. $c^{\mathfrak{M}/_\approx} = [c]_\approx$

3. $f^{\mathfrak{M}/_\approx}([t_1]_\approx, \dots, [t_n]_\approx) = [f(t_1, \dots, t_n)]_\approx$

4. $\langle [t_1]_\approx, \dots, [t_n]_\approx \rangle \in R^{\mathfrak{M}/_\approx}$ iff $\mathfrak{M} \vDash R(t_1, \dots, t_n)$, i.e., iff $R(t_1, \dots, t_n) \in \Gamma^*$.

Note that we have defined $f^{\mathfrak{M}/_\approx}$ and $R^{\mathfrak{M}/_\approx}$ for elements of $\mathrm{Trm}(\mathcal{L})/_\approx$ by referring to them as $[t]_\approx$, i.e., via *representatives* $t \in [t]_\approx$. We have to make sure that these definitions do not depend on the choice of these representatives, i.e., that for some other choices t' which determine the same equivalence classes ($[t]_\approx = [t']_\approx$), the definitions yield the same result. For instance, if R is a one-place predicate symbol, the last clause of the definition says that $[t]_\approx \in R^{\mathfrak{M}/_\approx}$ iff $\mathfrak{M} \vDash R(t)$. If for some other term t' with $t \approx t'$, $\mathfrak{M} \nvDash R(t)$, then the definition would require $[t']_\approx \notin R^{\mathfrak{M}/_\approx}$. If $t \approx t'$, then $[t]_\approx = [t']_\approx$, but we can't have both $[t]_\approx \in R^{\mathfrak{M}/_\approx}$ and $[t]_\approx \notin R^{\mathfrak{M}/_\approx}$. However, Proposition 10.14 guarantees that this cannot happen. explanation

Proposition 10.17. *$\mathfrak{M}/_\approx$ is well defined, i.e., if $t_1, \dots, t_n, t'_1, \dots, t'_n$ are terms, and $t_i \approx t'_i$ then*

1. *$[f(t_1, \dots, t_n)]_\approx = [f(t'_1, \dots, t'_n)]_\approx$, i.e.,*

$$f(t_1, \dots, t_n) \approx f(t'_1, \dots, t'_n)$$

and

2. *$\mathfrak{M} \vDash R(t_1, \dots, t_n)$ iff $\mathfrak{M} \vDash R(t'_1, \dots, t'_n)$, i.e.,*

$$R(t_1, \dots, t_n) \in \Gamma^* \text{ iff } R(t'_1, \dots, t'_n) \in \Gamma^*.$$

Proof. Follows from Proposition 10.14 by induction on n. □

As in the case of the term model, before proving the truth lemma we need the following lemma.

Lemma 10.18. *Let $\mathfrak{M} = \mathfrak{M}(\Gamma^*)$, then $\mathrm{Val}^{\mathfrak{M}/_\approx}(t) = [t]_\approx$.* fol:com:ide: lem:val-in-termmodel-factored

Proof. The proof is similar to that of Lemma 10.10. □

Problem 10.3. Complete the proof of Lemma 10.18.

Lemma 10.19. *$\mathfrak{M}/_\approx \vDash \varphi$ iff $\varphi \in \Gamma^*$ for all sentences φ.* fol:com:ide: lem:truth

Proof. By induction on φ, just as in the proof of Lemma 10.12. The only case that needs additional attention is when $\varphi \equiv t = t'$.

$$\begin{aligned} \mathfrak{M}/_\approx \vDash t = t' & \text{ iff } [t]_\approx = [t']_\approx \text{ (by definition of } \mathfrak{M}/_\approx) \\ & \text{ iff } t \approx t' \text{ (by definition of } [t]_\approx) \\ & \text{ iff } t = t' \in \Gamma^* \text{ (by definition of } \approx). \end{aligned}$$ □

digression Note that while $\mathfrak{M}(\Gamma^*)$ is always enumerable and infinite, $\mathfrak{M}/_\approx$ may be finite, since it may turn out that there are only finitely many classes $[t]_\approx$. This is to be expected, since Γ may contain sentences which require any structure in which they are true to be finite. For instance, $\forall x\, \forall y\, x = y$ is a consistent sentence, but is satisfied only in structures with a domain that contains exactly one element.

10.8 The Completeness Theorem

explanation Let's combine our results: we arrive at the completeness theorem. fol:com:cth: sec

Theorem 10.20 (Completeness Theorem). *Let Γ be a set of sentences. If Γ is consistent, it is satisfiable.* fol:com:cth: thm:completeness

Proof. Suppose Γ is consistent. By Lemma 10.6, there is a saturated consistent set $\Gamma' \supseteq \Gamma$. By Lemma 10.8, there is a $\Gamma^* \supseteq \Gamma'$ which is consistent and complete. Since $\Gamma' \subseteq \Gamma^*$, for each formula $\varphi(x)$, Γ^* contains a sentence of the form $\exists x\, \varphi(x) \rightarrow \varphi(c)$ and so Γ^* is saturated. If Γ does not contain $=$, then by Lemma 10.12, $\mathfrak{M}(\Gamma^*) \vDash \varphi$ iff $\varphi \in \Gamma^*$. From this it follows in particular that for all $\varphi \in \Gamma$, $\mathfrak{M}(\Gamma^*) \vDash \varphi$, so Γ is satisfiable. If Γ does contain $=$, then by Lemma 10.19, for all sentences φ, $\mathfrak{M}/_{\approx} \vDash \varphi$ iff $\varphi \in \Gamma^*$. In particular, $\mathfrak{M}/_{\approx} \vDash \varphi$ for all $\varphi \in \Gamma$, so Γ is satisfiable. □

fol:com:cth:cor:completeness **Corollary 10.21 (Completeness Theorem, Second Version).** *For all Γ and sentences φ: if $\Gamma \vDash \varphi$ then $\Gamma \vdash \varphi$.*

Proof. Note that the Γ's in Corollary 10.21 and Theorem 10.20 are universally quantified. To make sure we do not confuse ourselves, let us restate Theorem 10.20 using a different variable: for any set of sentences Δ, if Δ is consistent, it is satisfiable. By contraposition, if Δ is not satisfiable, then Δ is inconsistent. We will use this to prove the corollary.

Suppose that $\Gamma \vDash \varphi$. Then $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable by Proposition 3.27. Taking $\Gamma \cup \{\neg\varphi\}$ as our Δ, the previous version of Theorem 10.20 gives us that $\Gamma \cup \{\neg\varphi\}$ is inconsistent. By Propositions 6.19, 7.19, 8.19 and 9.27, $\Gamma \vdash \varphi$. □

Problem 10.4. Use Corollary 10.21 to prove Theorem 10.20, thus showing that the two formulations of the completeness theorem are equivalent.

Problem 10.5. In order for a derivation system to be complete, its rules must be strong enough to prove every unsatisfiable set inconsistent. Which of the rules of derivation were necessary to prove completeness? Are any of these rules not used anywhere in the proof? In order to answer these questions, make a list or diagram that shows which of the rules of derivation were used in which results that lead up to the proof of Theorem 10.20. Be sure to note any tacit uses of rules in these proofs.

10.9 The Compactness Theorem

fol:com:com:sec One important consequence of the completeness theorem is the compactness theorem. The compactness theorem states that if each *finite* subset of a set of sentences is satisfiable, the entire set is satisfiable—even if the set itself is infinite. This is far from obvious. There is nothing that seems to rule out, at first glance at least, the possibility of there being infinite sets of sentences which are contradictory, but the contradiction only arises, so to speak, from the infinite number. The compactness theorem says that such a scenario can be ruled out: there are no unsatisfiable infinite sets of sentences each finite subset of which is satisfiable. Like the completeness theorem, it has a version related to entailment: if an infinite set of sentences entails something, already a finite subset does.

Definition 10.22. A set Γ of formulas is *finitely satisfiable* iff every finite $\Gamma_0 \subseteq \Gamma$ is satisfiable.

Theorem 10.23 (Compactness Theorem). *The following hold for any sentences Γ and φ:* fol:com:com:thm:compactness

1. *$\Gamma \vDash \varphi$ iff there is a finite $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vDash \varphi$.*

2. *Γ is satisfiable iff it is finitely satisfiable.*

Proof. We prove (2). If Γ is satisfiable, then there is a structure $\mathfrak{M}$ such that $\mathfrak{M} \vDash \varphi$ for all $\varphi \in \Gamma$. Of course, this $\mathfrak{M}$ also satisfies every finite subset of Γ, so Γ is finitely satisfiable.

Now suppose that Γ is finitely satisfiable. Then every finite subset $\Gamma_0 \subseteq \Gamma$ is satisfiable. By soundness (Corollaries 7.29, 6.31, 8.31 and 9.38), every finite subset is consistent. Then Γ itself must be consistent by Propositions 6.17, 7.17, 8.17 and 9.21. By completeness (Theorem 10.20), since Γ is consistent, it is satisfiable. $\square$

Problem 10.6. Prove (1) of Theorem 10.23.

Example 10.24. In every model $\mathfrak{M}$ of a theory Γ, each term t of course picks out an element of $|\mathfrak{M}|$. Can we guarantee that it is also true that every element of $|\mathfrak{M}|$ is picked out by some term or other? In other words, are there theories Γ all models of which are covered? The compactness theorem shows that this is not the case if Γ has infinite models. Here's how to see this: Let $\mathfrak{M}$ be an infinite model of Γ, and let c be a constant symbol not in the language of Γ. Let Δ be the set of all sentences $c \neq t$ for t a term in the language $\mathcal{L}$ of Γ, i.e.,

$$\Delta = \{c \neq t : t \in \mathrm{Trm}(\mathcal{L})\}.$$

A finite subset of $\Gamma \cup \Delta$ can be written as $\Gamma' \cup \Delta'$, with $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$. Since Δ' is finite, it can contain only finitely many terms. Let $a \in |\mathfrak{M}|$ be an element of $|\mathfrak{M}|$ not picked out by any of them, and let $\mathfrak{M}'$ be the structure that is just like $\mathfrak{M}$, but also $c^{\mathfrak{M}'} = a$. Since $a \neq \mathrm{Val}^{\mathfrak{M}}(t)$ for all t occuring in Δ', $\mathfrak{M}' \vDash \Delta'$. Since $\mathfrak{M} \vDash \Gamma$, $\Gamma' \subseteq \Gamma$, and c does not occur in Γ, also $\mathfrak{M}' \vDash \Gamma'$. Together, $\mathfrak{M}' \vDash \Gamma' \cup \Delta'$ for every finite subset $\Gamma' \cup \Delta'$ of $\Gamma \cup \Delta$. So every finite subset of $\Gamma \cup \Delta$ is satisfiable. By compactness, $\Gamma \cup \Delta$ itself is satisfiable. So there are models $\mathfrak{M} \vDash \Gamma \cup \Delta$. Every such $\mathfrak{M}$ is a model of Γ, but is not covered, since $\mathrm{Val}^{\mathfrak{M}}(c) \neq \mathrm{Val}^{\mathfrak{M}}(t)$ for all terms t of $\mathcal{L}$.

Example 10.25. Consider a language $\mathcal{L}$ containing the predicate symbol $<$, constant symbols $\mathtt{0}$, $\mathtt{1}$, and function symbols $+$, $\times$, $-$, $\div$. Let Γ be the set of all sentences in this language true in $\mathfrak{Q}$ with domain $\mathbb{Q}$ and the obvious interpretations. Γ is the set of all sentences of $\mathcal{L}$ true about the rational numbers. Of course, in $\mathbb{Q}$ (and even in $\mathbb{R}$), there are no numbers which are greater than 0 but less than $1/k$ for all $k \in \mathbb{Z}^+$. Such a number, if it existed, would be an *infinitesimal:* non-zero, but infinitely small. The compactness theorem shows that there are models of Γ in which infinitesimals exist: Let Δ be $\{0 < c\} \cup \{c < (\mathtt{1} \div \overline{k}) : k \in \mathbb{Z}^+\}$ (where $\overline{k} = (\mathtt{1} + (\mathtt{1} + \dots + (\mathtt{1} + \mathtt{1})\dots))$ with k $\mathtt{1}$'s). For any finite subset Δ_0 of Δ there is a K such that all the sentences $c < (\mathtt{1} \div \overline{k})$ in Δ_0 have $k < K$. If we expand $\mathfrak{Q}$ to $\mathfrak{Q}'$ with $c^{\mathfrak{Q}'} = 1/K$ we have that $\mathfrak{Q}' \vDash \Gamma \cup \Delta_0$, and so $\Gamma \cup \Delta$ is finitely satisfiable (Exercise: prove this in detail). By compactness, $\Gamma \cup \Delta$ is satisfiable. Any model $\mathfrak{S}$ of $\Gamma \cup \Delta$ contains an infinitesimal, namely $c^{\mathfrak{S}}$.

Problem 10.7. In the standard model of arithmetic $\mathfrak{N}$, there is no element $k \in |\mathfrak{N}|$ which satisfies every formula $\overline{n} < x$ (where $\overline{n}$ is $\mathtt{0}'^{\dots\prime}$ with n $\prime$'s). Use the compactness theorem to show that the set of sentences in the language of arithmetic which are true in the standard model of arithmetic $\mathfrak{N}$ are also true in a structure $\mathfrak{N}'$ that contains an element which *does* satisfy every formula $\overline{n} < x$.

Example 10.26. We know that first-order logic with identity predicate can express that the size of the domain must have some minimal size: The sentence $\varphi_{\geq n}$ (which says "there are at least n distinct objects") is true only in structures where $|\mathfrak{M}|$ has at least n objects. So if we take

$$\Delta = \{\varphi_{\geq n} : n \geq 1\}$$

then any model of Δ must be infinite. Thus, we can guarantee that a theory only has infinite models by adding Δ to it: the models of $\Gamma \cup \Delta$ are all and only the infinite models of Γ.

So first-order logic can express infinitude. The compactness theorem shows that it cannot express finitude, however. For suppose some set of sentences Λ were satisfied in all and only finite structures. Then $\Delta \cup \Lambda$ is finitely satisfiable. Why? Suppose $\Delta' \cup \Lambda' \subseteq \Delta \cup \Lambda$ is finite with $\Delta' \subseteq \Delta$ and $\Lambda' \subseteq \Lambda$. Let n be the largest number such that $\varphi_{\geq n} \in \Delta'$. Λ, being satisfied in all finite structures, has a model $\mathfrak{M}$ with finitely many but $\geq n$ elements. But then $\mathfrak{M} \vDash \Delta' \cup \Lambda'$. By compactness, $\Delta \cup \Lambda$ has an infinite model, contradicting the assumption that Λ is satisfied only in finite structures.

10.10 A Direct Proof of the Compactness Theorem

fol:com:cpd:sec We can prove the Compactness Theorem directly, without appealing to the Completeness Theorem, using the same ideas as in the proof of the completeness theorem. In the proof of the Completeness Theorem we started with a consistent set Γ of sentences, expanded it to a consistent, saturated, and complete set Γ^* of sentences, and then showed that in the term model $\mathfrak{M}(\Gamma^*)$ constructed from Γ^*, all sentences of Γ are true, so Γ is satisfiable.

We can use the same method to show that a finitely satisfiable set of sentences is satisfiable. We just have to prove the corresponding versions of the results leading to the truth lemma where we replace "consistent" with "finitely satisfiable."

Proposition 10.27. *Suppose Γ is complete and finitely satisfiable. Then:* fol:com:cpd: prop:fsat-ccs

1. $(\varphi \wedge \psi) \in \Gamma$ *iff both* $\varphi \in \Gamma$ *and* $\psi \in \Gamma$.
2. $(\varphi \vee \psi) \in \Gamma$ *iff either* $\varphi \in \Gamma$ *or* $\psi \in \Gamma$.
3. $(\varphi \rightarrow \psi) \in \Gamma$ *iff either* $\varphi \notin \Gamma$ *or* $\psi \in \Gamma$.

Problem 10.8. Prove Proposition 10.27. Avoid the use of $\vdash$.

Lemma 10.28. *Every finitely satisfiable set Γ can be extended to a saturated finitely satisfiable set Γ'.* fol:com:cpd: lem:fsat-henkin

Problem 10.9. Prove Lemma 10.28. (Hint: The crucial step is to show that if Γ_n is finitely satisfiable, so is $\Gamma_n \cup \{\theta_n\}$, without any appeal to derivations or consistency.)

Proposition 10.29. *Suppose Γ is complete, finitely satisfiable, and saturated.* fol:com:cpd: prop:fsat-instances

1. $\exists x\, \varphi(x) \in \Gamma$ *iff* $\varphi(t) \in \Gamma$ *for at least one closed term* t.
2. $\forall x\, \varphi(x) \in \Gamma$ *iff* $\varphi(t) \in \Gamma$ *for all closed terms* t.

Problem 10.10. Prove Proposition 10.29.

Lemma 10.30. *Every finitely satisfiable set Γ can be extended to a complete and finitely satisfiable set Γ^*.* fol:com:cpd: lem:fsat-lindenbaum

Problem 10.11. Prove Lemma 10.30. (Hint: the crucial step is to show that if Γ_n is finitely satisfiable, then either $\Gamma_n \cup \{\varphi_n\}$ or $\Gamma_n \cup \{\neg\varphi_n\}$ is finitely satisfiable.)

Theorem 10.31 (Compactness). *Γ is satisfiable if and only if it is finitely satisfiable.* fol:com:cpd: thm:compactness-direct

Proof. If Γ is satisfiable, then there is a structure $\mathfrak{M}$ such that $\mathfrak{M} \vDash \varphi$ for all $\varphi \in \Gamma$. Of course, this $\mathfrak{M}$ also satisfies every finite subset of Γ, so Γ is finitely satisfiable.

Now suppose that Γ is finitely satisfiable. By Lemma 10.28, there is a finitely satisfiable, saturated set $\Gamma' \supseteq \Gamma$. By Lemma 10.30, Γ' can be extended to a complete and finitely satisfiable set Γ^*, and Γ^* is still saturated. Construct the term model $\mathfrak{M}(\Gamma^*)$ as in Definition 10.9. Note that Proposition 10.11 did not rely on the fact that Γ^* is consistent (or complete or saturated, for that matter), but just on the fact that $\mathfrak{M}(\Gamma^*)$ is covered. The proof of the Truth Lemma (Lemma 10.12) goes through if we replace references to Proposition 10.2 and Proposition 10.7 by references to Proposition 10.27 and Proposition 10.29 □

Problem 10.12. Write out the complete proof of the Truth Lemma (Lemma 10.12) in the version required for the proof of Theorem 10.31.

10.11 The Löwenheim-Skolem Theorem

fol:com:dls: sec

The Löwenheim-Skolem Theorem says that if a theory has an infinite model, then it also has a model that is at most denumerable. An immediate consequence of this fact is that first-order logic cannot express that the size of a structure is non-enumerable: any sentence or set of sentences satisfied in all non-enumerable structures is also satisfied in some enumerable structure.

fol:com:dls: thm:downward-ls

Theorem 10.32. *If Γ is consistent then it has an enumerable model, i.e., it is satisfiable in a structure whose domain is either finite or denumerable.*

Proof. If Γ is consistent, the structure $\mathfrak{M}$ delivered by the proof of the completeness theorem has a domain $|\mathfrak{M}|$ that is no larger than the set of the terms of the language $\mathcal{L}$. So $\mathfrak{M}$ is at most denumerable. □

fol:com:dls: noidentity-ls

Theorem 10.33. *If Γ is a consistent set of sentences in the language of first-order logic without identity, then it has a denumerable model, i.e., it is satisfiable in a structure whose domain is infinite and enumerable.*

Proof. If Γ is consistent and contains no sentences in which identity appears, then the structure $\mathfrak{M}$ delivered by the proof of the completness theorem has a domain $|\mathfrak{M}|$ identical to the set of terms of the language $\mathcal{L}'$. So $\mathfrak{M}$ is denumerable, since $\mathrm{Trm}(\mathcal{L}')$ is. □

Example 10.34 (Skolem's Paradox). Zermelo-Fraenkel set theory **ZFC** is a very powerful framework in which practically all mathematical statements can be expressed, including facts about the sizes of sets. So for instance, **ZFC** can prove that the set $\mathbb{R}$ of real numbers is non-enumerable, it can prove Cantor's Theorem that the power set of any set is larger than the set itself, etc. If **ZFC** is consistent, its models are all infinite, and moreover, they all contain elements about which the theory says that they are non-enumerable, such as the element that makes true the theorem of **ZFC** that the power set of the natural numbers exists. By the Löwenheim-Skolem Theorem, **ZFC** also has enumerable models—models that contain "non-enumerable" sets but which themselves are enumerable.

Chapter 11

Beyond First-order Logic

This chapter, adapted from Jeremy Avigad's logic notes, gives the briefest of glimpses into which other logical systems there are. It is intended as a chapter suggesting further topics for study in a course that does not cover them. Each one of the topics mentioned here will—hopefully—eventually receive its own part-level treatment in the Open Logic Project.

11.1 Overview

First-order logic is not the only system of logic of interest: there are many extensions and variations of first-order logic. A logic typically consists of the formal specification of a language, usually, but not always, a deductive system, and usually, but not always, an intended semantics. But the technical use of the term raises an obvious question: what do logics that are not first-order logic have to do with the word "logic," used in the intuitive or philosophical sense? All of the systems described below are designed to model reasoning of some form or another; can we say what makes them logical? fol:byd:int:sec

No easy answers are forthcoming. The word "logic" is used in different ways and in different contexts, and the notion, like that of "truth," has been analyzed from numerous philosophical stances. For example, one might take the goal of logical reasoning to be the determination of which statements are necessarily true, true a priori, true independent of the interpretation of the nonlogical terms, true by virtue of their form, or true by linguistic convention; and each of these conceptions requires a good deal of clarification. Even if one restricts one's attention to the kind of logic used in mathematics, there is little agreement as to its scope. For example, in the *Principia Mathematica*, Russell and Whitehead tried to develop mathematics on the basis of logic, in the *logicist* tradition begun by Frege. Their system of logic was a form of higher-type logic similar to the one described below. In the end they were forced to introduce axioms which, by most standards, do not seem purely logical (notably, the axiom of infinity, and the axiom of reducibility), but one might nonetheless hold that some forms of higher-order reasoning should be accepted as logical. In contrast, Quine, whose ontology does not admit "propositions" as legitimate objects of discourse, argues that second-order and higher-order logic are really manifestations of set theory in sheep's clothing; in other words, systems involving quantification over predicates are not purely logical.

For now, it is best to leave such philosophical issues for a rainy day, and simply think of the systems below as formal idealizations of various kinds of reasoning, logical or otherwise.

11.2 Many-Sorted Logic

fol:byd:msl:sec

In first-order logic, variables and quantifiers range over a single domain. But it is often useful to have multiple (disjoint) domains: for example, you might want to have a domain of numbers, a domain of geometric objects, a domain of functions from numbers to numbers, a domain of abelian groups, and so on.

Many-sorted logic provides this kind of framework. One starts with a list of "sorts"—the "sort" of an object indicates the "domain" it is supposed to inhabit. One then has variables and quantifiers for each sort, and (usually) an identity predicate for each sort. Functions and relations are also "typed" by the sorts of objects they can take as arguments. Otherwise, one keeps the usual rules of first-order logic, with versions of the quantifier-rules repeated for each sort.

For example, to study international relations we might choose a language with two sorts of objects, French citizens and German citizens. We might have a unary relation, "drinks wine," for objects of the first sort; another unary relation, "eats wurst," for objects of the second sort; and a binary relation, "forms a multinational married couple," which takes two arguments, where the first argument is of the first sort and the second argument is of the second sort. If we use variables a, b, c to range over French citizens and x, y, z to range over German citizens, then

$$\forall a\, \forall x[(\mathit{MarriedTo}(a, x) \to (\mathit{DrinksWine}(a) \lor \lnot \mathit{EatsWurst}(x))]]$$

asserts that if any French person is married to a German, either the French person drinks wine or the German doesn't eat wurst.

Many-sorted logic can be embedded in first-order logic in a natural way, by lumping all the objects of the many-sorted domains together into one first-order domain, using unary predicate symbols to keep track of the sorts, and relativizing quantifiers. For example, the first-order language corresponding to the example above would have unary predicate symbols "German" and "French," in addition to the other relations described, with the sort requirements erased. A sorted quantifier $\forall x\, \varphi$, where x is a variable of the German sort, translates to

$$\forall x\, (\mathit{German}(x) \to \varphi).$$

We need to add axioms that insure that the sorts are separate—e.g., $\forall x\, \lnot(\mathit{German}(x) \land \mathit{French}(x))$—as well as axioms that guarantee that "drinks wine" only holds of objects satisfying the predicate $French(x)$, etc. With these conventions and axioms, it is not difficult to show that many-sorted sentences translate to first-order sentences, and many-sorted derivations translate to first-order derivations. Also, many-sorted structures "translate" to corresponding first-order structures and vice-versa, so we also have a completeness theorem for many-sorted logic.

11.3 Second-Order logic

The language of second-order logic allows one to quantify not just over a domain of individuals, but over relations on that domain as well. Given a first-order language $\mathcal{L}$, for each k one adds variables R which range over k-ary relations, and allows quantification over those variables. If R is a variable for a k-ary relation, and $t_1, \dots, t_k$ are ordinary (first-order) terms, $R(t_1, \dots, t_k)$ is an atomic formula. Otherwise, the set of formulas is defined just as in the case of first-order logic, with additional clauses for second-order quantification. Note that we only have the identity predicate for first-order terms: if R and S are relation variables of the same arity k, we can define $R = S$ to be an abbreviation for fol:byd:sol:sec

$$\forall x_1 \dots \forall x_k\, (R(x_1, \dots, x_k) \leftrightarrow S(x_1, \dots, x_k)).$$

The rules for second-order logic simply extend the quantifier rules to the new second order variables. Here, however, one has to be a little bit careful to explain how these variables interact with the predicate symbols of $\mathcal{L}$, and with formulas of $\mathcal{L}$ more generally. At the bare minimum, relation variables count as terms, so one has inferences of the form

$$\varphi(R) \vdash \exists R\, \varphi(R)$$

But if $\mathcal{L}$ is the language of arithmetic with a constant relation symbol $<$, one would also expect the following inference to be valid:

$$x < y \vdash \exists R\, R(x, y)$$

or for a given formula φ,

$$\varphi(x_1, \dots, x_k) \vdash \exists R\, R(x_1, \dots, x_k)$$

More generally, we might want to allow inferences of the form

$$\varphi[\lambda \vec{x}.\, \psi(\vec{x})/R] \vdash \exists R\, \varphi$$

where $\varphi[\lambda \vec{x}.\, \psi(\vec{x})/R]$ denotes the result of replacing every atomic formula of the form $Rt_1, \dots, t_k$ in φ by $\psi(t_1, \dots, t_k)$. This last rule is equivalent to having a *comprehension schema*, i.e., an axiom of the form

$$\exists R\, \forall x_1, \dots, x_k\, (\varphi(x_1, \dots, x_k) \leftrightarrow R(x_1, \dots, x_k)),$$

one for each formula φ in the second-order language, in which R is not a free variable. (Exercise: show that if R is allowed to occur in φ, this schema is inconsistent!)

When logicians refer to the "axioms of second-order logic" they usually mean the minimal extension of first-order logic by second-order quantifier rules together with the comprehension schema. But it is often interesting to study weaker subsystems of these axioms and rules. For example, note that in its full generality the axiom schema of comprehension is *impredicative*: it allows one to assert the existence of a relation $R(x_1, \dots, x_k)$ that is "defined" by a formula with second-order quantifiers; and these quantifiers range over the set of all such relations—a set which includes R itself! Around the turn of the twentieth century, a common reaction to Russell's paradox was to lay the blame on such definitions, and to avoid them in developing the foundations of mathematics. If one prohibits the use of second-order quantifiers in the formula φ, one has a *predicative* form of comprehension, which is somewhat weaker.

From the semantic point of view, one can think of a second-order structure as consisting of a first-order structure for the language, coupled with a set of relations on the domain over which the second-order quantifiers range (more precisely, for each k there is a set of relations of arity k). Of course, if comprehension is included in the derivation system, then we have the added requirement that there are enough relations in the "second-order part" to satisfy the comprehension axioms—otherwise the derivation system is not sound! One easy way to insure that there are enough relations around is to take the second-order part to consist of *all* the relations on the first-order part. Such a structure is called *full*, and, in a sense, is really the "intended structure" for the language. If we restrict our attention to full structures we have what is known as the *full* second-order semantics. In that case, specifying a structure boils down to specifying the first-order part, since the contents of the second-order part follow from that implicitly.

To summarize, there is some ambiguity when talking about second-order logic. In terms of the derivation system, one might have in mind either

1. A "minimal" second-order derivation system, together with some comprehension axioms.

2. The "standard" second-order derivation system, with full comprehension.

In terms of the semantics, one might be interested in either

1. The "weak" semantics, where a structure consists of a first-order part, together with a second-order part big enough to satisfy the comprehension axioms.

2. The "standard" second-order semantics, in which one considers full structures only.

When logicians do not specify the derivation system or the semantics they have in mind, they are usually refering to the second item on each list. The advantage to using this semantics is that, as we will see, it gives us categorical descriptions of many natural mathematical structures; at the same time, the derivation system is quite strong, and sound for this semantics. The drawback is that the derivation system is *not* complete for the semantics; in fact, *no* effectively given derivation system is complete for the full second-order semantics. On the other hand, we will see that the derivation system *is* complete for the weakened semantics; this implies that if a sentence is not provable, then there is *some* structure, not necessarily the full one, in which it is false.

The language of second-order logic is quite rich. One can identify unary relations with subsets of the domain, and so in particular you can quantify over these sets; for example, one can express induction for the natural numbers with a single axiom

$$\forall R\, ((R(\mathsf{o}) \land \forall x\, (R(x) \to R(x'))) \to \forall x\, R(x)).$$

If one takes the language of arithmetic to have symbols $\mathsf{o}, {\prime}, +, \times$ and $<$, one can add the following axioms to describe their behavior:

1. $\forall x\, \neg x' = \mathsf{o}$
2. $\forall x\, \forall y\, (s(x) = s(y) \to x = y)$
3. $\forall x\, (x + \mathsf{o}) = x$
4. $\forall x\, \forall y\, (x + y') = (x + y)'$
5. $\forall x\, (x \times \mathsf{o}) = \mathsf{o}$
6. $\forall x\, \forall y\, (x \times y') = ((x \times y) + x)$
7. $\forall x\, \forall y\, (x < y \leftrightarrow \exists z\, y = (x + z'))$

It is not difficult to show that these axioms, together with the axiom of induction above, provide a categorical description of the structure $\mathfrak{N}$, the standard model of arithmetic, provided we are using the full second-order semantics. Given any structure $\mathfrak{M}$ in which these axioms are true, define a function f from $\mathbb{N}$ to the domain of $\mathfrak{M}$ using ordinary recursion on $\mathbb{N}$, so that $f(0) = \mathsf{o}^{\mathfrak{M}}$ and $f(x + 1) = {\prime}^{\mathfrak{M}}(f(x))$. Using ordinary induction on $\mathbb{N}$ and the fact that axioms (1) and (2) hold in $\mathfrak{M}$, we see that f is injective. To see that f is surjective, let P be the set of elements of $|\mathfrak{M}|$ that are in the range of f. Since $\mathfrak{M}$ is full, P is in the second-order domain. By the construction of f, we know that $\mathsf{o}^{\mathfrak{M}}$ is in P, and that P is closed under ${\prime}^{\mathfrak{M}}$. The fact that the induction axiom holds in $\mathfrak{M}$ (in particular, for P) guarantees that P is equal to the entire first-order domain of $\mathfrak{M}$. This shows that f is a bijection. Showing that f is a homomorphism is no more difficult, using ordinary induction on $\mathbb{N}$ repeatedly.

In set-theoretic terms, a function is just a special kind of relation; for example, a unary function f can be identified with a binary relation R satisfying $\forall x\, \exists! y\, R(x, y)$. As a result, one can quantify over functions too. Using the full semantics, one can then define the class of infinite structures to be the class of structures $\mathfrak{M}$ for which there is an injective function from the domain of $\mathfrak{M}$ to a proper subset of itself:

$$\exists f\,(\forall x\,\forall y\,(f(x) = f(y) \rightarrow x = y) \land \exists y\,\forall x\, f(x) \neq y).$$

The negation of this sentence then defines the class of finite structures.

In addition, one can define the class of well-orderings, by adding the following to the definition of a linear ordering:

$$\forall P\,(\exists x\, P(x) \rightarrow \exists x\,(P(x) \land \forall y\,(y < x \rightarrow \neg P(y)))).$$

This asserts that every non-empty set has a least element, modulo the identification of "set" with "one-place relation". For another example, one can express the notion of connectedness for graphs, by saying that there is no nontrivial separation of the vertices into disconnected parts:

$$\neg\exists A\,(\exists x\, A(x) \land \exists y\, \neg A(y) \land \forall w\, \forall z\,((A(w) \land \neg A(z)) \rightarrow \neg R(w, z))).$$

For yet another example, you might try as an exercise to define the class of finite structures whose domain has even size. More strikingly, one can provide a categorical description of the real numbers as a complete ordered field containing the rationals.

In short, second-order logic is much more expressive than first-order logic. That's the good news; now for the bad. We have already mentioned that there is no effective derivation system that is complete for the full second-order semantics. For better or for worse, many of the properties of first-order logic are absent, including compactness and the Löwenheim-Skolem theorems.

On the other hand, if one is willing to give up the full second-order semantics in terms of the weaker one, then the minimal second-order derivation system is complete for this semantics. In other words, if we read $\vdash$ as "proves in the minimal system" and $\vDash$ as "logically implies in the weaker semantics", we can show that whenever $\Gamma \vDash \varphi$ then $\Gamma \vdash \varphi$. If one wants to include specific comprehension axioms in the derivation system, one has to restrict the semantics to second-order structures that satisfy these axioms: for example, if Δ consists of a set of comprehension axioms (possibly all of them), we have that if $\Gamma \cup \Delta \vDash \varphi$, then $\Gamma \cup \Delta \vdash \varphi$. In particular, if φ is not provable using the comprehension axioms we are considering, then there is a model of $\neg\varphi$ in which these comprehension axioms nonetheless hold.

The easiest way to see that the completeness theorem holds for the weaker semantics is to think of second-order logic as a many-sorted logic, as follows. One sort is interpreted as the ordinary "first-order" domain, and then for each k we have a domain of "relations of arity k." We take the language to have built-in relation symbols "$\mathit{true}_k(R, x_1, \ldots, x_k)$" which is meant to assert that R holds of x_1, …, x_k, where R is a variable of the sort "k-ary relation" and x_1, …, x_k are objects of the first-order sort.

With this identification, the weak second-order semantics is essentially the usual semantics for many-sorted logic; and we have already observed that many-sorted logic can be embedded in first-order logic. Modulo the translations back and forth, then, the weaker conception of second-order logic is really a form of first-order logic in disguise, where the domain contains both "objects" and "relations" governed by the appropriate axioms.

11.4 Higher-Order logic

Passing from first-order logic to second-order logic enabled us to talk about sets of objects in the first-order domain, within the formal language. Why stop there? For example, third-order logic should enable us to deal with sets of sets of objects, or perhaps even sets which contain both objects and sets of objects. And fourth-order logic will let us talk about sets of objects of that kind. As you may have guessed, one can iterate this idea arbitrarily. fol:byd:hol:sec

In practice, higher-order logic is often formulated in terms of functions instead of relations. (Modulo the natural identifications, this difference is inessential.) Given some basic "sorts" A, B, C, ... (which we will now call "types"), we can create new ones by stipulating

> If σ and τ are finite types then so is $\sigma \to \tau$.

Think of types as syntactic "labels," which classify the objects we want in our domain; $\sigma \to \tau$ describes those objects that are functions which take objects of type σ to objects of type τ. For example, we might want to have a type Ω of truth values, "true" and "false," and a type $\mathbb{N}$ of natural numbers. In that case, you can think of objects of type $\mathbb{N} \to \Omega$ as unary relations, or subsets of $\mathbb{N}$; objects of type $\mathbb{N} \to \mathbb{N}$ are functions from natural numers to natural numbers; and objects of type $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ are "functionals," that is, higher-type functions that take functions to numbers.

As in the case of second-order logic, one can think of higher-order logic as a kind of many-sorted logic, where there is a sort for each type of object we want to consider. But it is usually clearer just to define the syntax of higher-type logic from the ground up. For example, we can define a set of finite types inductively, as follows:

1. $\mathbb{N}$ is a finite type.
2. If σ and τ are finite types, then so is $\sigma \to \tau$.
3. If σ and τ are finite types, so is $\sigma \times \tau$.

Intuitively, $\mathbb{N}$ denotes the type of the natural numbers, $\sigma \to \tau$ denotes the type of functions from σ to τ, and $\sigma \times \tau$ denotes the type of pairs of objects, one from σ and one from τ. We can then define a set of terms inductively, as follows:

1. For each type σ, there is a stock of variables x, y, z, ... of type σ
2. $\mathsf{0}$ is a term of type $\mathbb{N}$

3. S (successor) is a term of type $\mathbb{N} \to \mathbb{N}$

4. If s is a term of type σ, and t is a term of type $\mathbb{N} \to (\sigma \to \sigma)$, then R_{st} is a term of type $\mathbb{N} \to \sigma$

5. If s is a term of type $\tau \to \sigma$ and t is a term of type τ, then $s(t)$ is a term of type σ

6. If s is a term of type σ and x is a variable of type τ, then $\lambda x. s$ is a term of type $\tau \to \sigma$.

7. If s is a term of type σ and t is a term of type τ, then $\langle s, t \rangle$ is a term of type $\sigma \times \tau$.

8. If s is a term of type $\sigma \times \tau$ then $p_1(s)$ is a term of type σ and $p_2(s)$ is a term of type τ.

Intuitively, R_{st} denotes the function defined recursively by

$$\mathsf{R}_{st}(0) = s$$
$$\mathsf{R}_{st}(x+1) = t(x, \mathsf{R}_{st}(x)),$$

$\langle s, t \rangle$ denotes the pair whose first component is s and whose second component is t, and $p_1(s)$ and $p_2(s)$ denote the first and second elements ("projections") of s. Finally, $\lambda x. s$ denotes the function f defined by

$$f(x) = s$$

for any x of type σ; so item (6) gives us a form of comprehension, enabling us to define functions using terms. Formulas are built up from identity predicate statements $s = t$ between terms of the same type, the usual propositional connectives, and higher-type quantification. One can then take the axioms of the system to be the basic equations governing the terms defined above, together with the usual rules of logic with quantifiers and identity predicate.

If one augments the finite type system with a type Ω of truth values, one has to include axioms which govern its use as well. In fact, if one is clever, one can get rid of complex formulas entirely, replacing them with terms of type Ω! The proof system can then be modified accordingly. The result is essentially the *simple theory of types* set forth by Alonzo Church in the 1930s.

As in the case of second-order logic, there are different versions of higher-type semantics that one might want to use. In the full version, variables of type $\sigma \to \tau$ range over the set of *all* functions from the objects of type σ to objects of type τ. As you might expect, this semantics is too strong to admit a complete, effective derivation system. But one can consider a weaker semantics, in which a structure consists of sets of elements T_τ for each type τ, together with appropriate operations for application, projection, etc. If the details are carried out correctly, one can obtain completeness theorems for the kinds of derivation systems described above.

Higher-type logic is attractive because it provides a framework in which we can embed a good deal of mathematics in a natural way: starting with $\mathbb{N}$, one can define real numbers, continuous functions, and so on. It is also particularly attractive in the context of intuitionistic logic, since the types have clear "constructive" intepretations. In fact, one can develop constructive versions of higher-type semantics (based on intuitionistic, rather than classical logic) that clarify these constructive interpretations quite nicely, and are, in many ways, more interesting than the classical counterparts.

11.5 Intuitionistic Logic

In constrast to second-order and higher-order logic, intuitionistic first-order logic represents a restriction of the classical version, intended to model a more "constructive" kind of reasoning. The following examples may serve to illustrate some of the underlying motivations. fol:byd:il:sec

Suppose someone came up to you one day and announced that they had determined a natural number x, with the property that if x is prime, the Riemann hypothesis is true, and if x is composite, the Riemann hypothesis is false. Great news! Whether the Riemann hypothesis is true or not is one of the big open questions of mathematics, and here they seem to have reduced the problem to one of calculation, that is, to the determination of whether a specific number is prime or not.

What is the magic value of x? They describe it as follows: x is the natural number that is equal to 7 if the Riemann hypothesis is true, and 9 otherwise.

Angrily, you demand your money back. From a classical point of view, the description above does in fact determine a unique value of x; but what you really want is a value of x that is given *explicitly.*

To take another, perhaps less contrived example, consider the following question. We know that it is possible to raise an irrational number to a rational power, and get a rational result. For example, $\sqrt{2}^2 = 2$. What is less clear is whether or not it is possible to raise an irrational number to an *irrational* power, and get a rational result. The following theorem answers this in the affirmative:

Theorem 11.1. *There are irrational numbers a and b such that a^b is rational.*

Proof. Consider $\sqrt{2}^{\sqrt{2}}$. If this is rational, we are done: we can let $a = b = \sqrt{2}$. Otherwise, it is irrational. Then we have

$$(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2}\cdot\sqrt{2}} = \sqrt{2}^2 = 2,$$

which is certainly rational. So, in this case, let a be $\sqrt{2}^{\sqrt{2}}$, and let b be $\sqrt{2}$. □

Does this constitute a valid proof? Most mathematicians feel that it does. But again, there is something a little bit unsatisfying here: we have proved the existence of a pair of real numbers with a certain property, without being able to say *which* pair of numbers it is. It is possible to prove the same result, but in such a way that the pair a, b *is* given in the proof: take $a = \sqrt{3}$ and $b = \log_3 4$. Then

$$a^b = \sqrt{3}^{\log_3 4} = 3^{1/2 \cdot \log_3 4} = (3^{\log_3 4})^{1/2} = 4^{1/2} = 2,$$

since $3^{\log_3 x} = x$.

Intuitionistic logic is designed to model a kind of reasoning where moves like the one in the first proof are disallowed. Proving the existence of an x satisfying $\varphi(x)$ means that you have to give a specific x, and a proof that it satisfies φ, like in the second proof. Proving that φ or ψ holds requires that you can prove one or the other.

Formally speaking, intuitionistic first-order logic is what you get if you restrict a derivation system for first-order logic in a certain way. Similarly, there are intuitionistic versions of second-order or higher-order logic. From the mathematical point of view, these are just formal deductive systems, but, as already noted, they are intended to model a kind of mathematical reasoning. One can take this to be the kind of reasoning that is justified on a certain philosophical view of mathematics (such as Brouwer's intuitionism); one can take it to be a kind of mathematical reasoning which is more "concrete" and satisfying (along the lines of Bishop's constructivism); and one can argue about whether or not the formal description captures the informal motivation. But whatever philosophical positions we may hold, we can study intuitionistic logic as a formally presented logic; and for whatever reasons, many mathematical logicians find it interesting to do so.

There is an informal constructive interpretation of the intuitionist connectives, usually known as the BHK interpretation (named after Brouwer, Heyting, and Kolmogorov). It runs as follows: a proof of $\varphi \wedge \psi$ consists of a proof of φ paired with a proof of ψ; a proof of $\varphi \vee \psi$ consists of either a proof of φ, or a proof of ψ, where we have explicit information as to which is the case; a proof of $\varphi \to \psi$ consists of a procedure, which transforms a proof of φ to a proof of ψ; a proof of $\forall x\, \varphi(x)$ consists of a procedure which returns a proof of $\varphi(x)$ for any value of x; and a proof of $\exists x\, \varphi(x)$ consists of a value of x, together with a proof that this value satisfies φ. One can describe the interpretation in computational terms known as the "Curry-Howard isomorphism" or the "formulas-as-types paradigm": think of a formula as specifying a certain kind of data type, and proofs as computational objects of these data types that enable us to see that the corresponding formula is true.

Intuitionistic logic is often thought of as being classical logic "minus" the law of the excluded middle. This following theorem makes this more precise.

Theorem 11.2. *Intuitionistically, the following axiom schemata are equivalent:*

1. $(\varphi \to \bot) \to \neg\varphi$.

2. $\varphi \lor \neg\varphi$

3. $\neg\neg\varphi \to \varphi$

Obtaining instances of one schema from either of the others is a good exercise in intuitionistic logic.

The first deductive systems for intuitionistic propositional logic, put forth as formalizations of Brouwer's intuitionism, are due, independently, to Kolmogorov, Glivenko, and Heyting. The first formalization of intuitionistic first-order logic (and parts of intuitionist mathematics) is due to Heyting. Though a number of classically valid schemata are not intuitionistically valid, many are.

The *double-negation translation* describes an important relationship between classical and intuitionist logic. It is defined inductively follows (think of φ^N as the "intuitionist" translation of the classical formula φ):

$$\begin{aligned}
\varphi^N &\equiv \neg\neg\varphi \quad \text{for atomic formulas } \varphi \\
(\varphi \land \psi)^N &\equiv (\varphi^N \land \psi^N) \\
(\varphi \lor \psi)^N &\equiv \neg\neg(\varphi^N \lor \psi^N) \\
(\varphi \to \psi)^N &\equiv (\varphi^N \to \psi^N) \\
(\forall x\, \varphi)^N &\equiv \forall x\, \varphi^N \\
(\exists x\, \varphi)^N &\equiv \neg\neg\exists x\, \varphi^N
\end{aligned}$$

Kolmogorov and Glivenko had versions of this translation for propositional logic; for predicate logic, it is due to Gödel and Gentzen, independently. We have

Theorem 11.3. *1. $\varphi \leftrightarrow \varphi^N$ is provable classically*

2. If φ is provable classically, then φ^N is provable intuitionistically.

We can now envision the following dialogue. Classical mathematician: "I've proved φ!" Intuitionist mathematician: "Your proof isn't valid. What you've really proved is φ^N." Classical mathematician: "Fine by me!" As far as the classical mathematician is concerned, the intuitionist is just splitting hairs, since the two are equivalent. But the intuitionist insists there is a difference.

Note that the above translation concerns pure logic only; it does not address the question as to what the appropriate *nonlogical* axioms are for classical and intuitionistic mathematics, or what the relationship is between them. But the following slight extension of the theorem above provides some useful information:

Theorem 11.4. *If Γ proves φ classically, Γ^N proves φ^N intuitionistically.*

In other words, if φ is provable from some hypotheses classically, then φ^N is provable from their double-negation translations.

To show that a sentence or propositional formula is intuitionistically valid, all you have to do is provide a proof. But how can you show that it is not valid? For that purpose, we need a semantics that is sound, and preferrably complete. A semantics due to Kripke nicely fits the bill.

We can play the same game we did for classical logic: define the semantics, and prove soundness and completeness. It is worthwhile, however, to note the following distinction. In the case of classical logic, the semantics was the "obvious" one, in a sense implicit in the meaning of the connectives. Though one can provide some intuitive motivation for Kripke semantics, the latter does not offer the same feeling of inevitability. In addition, the notion of a classical structure is a natural mathematical one, so we can either take the notion of a structure to be a tool for studying classical first-order logic, or take classical first-order logic to be a tool for studying mathematical structures. In contrast, Kripke structures can only be viewed as a logical construct; they don't seem to have independent mathematical interest.

A Kripke structure $\mathfrak{M} = \langle W, R, V \rangle$ for a propositional language consists of a set W, partial order R on W with a least element, and an "monotone" assignment of propositional variables to the elements of W. The intuition is that the elements of W represent "worlds," or "states of knowledge"; an element $v \geq u$ represents a "possible future state" of u; and the propositional variables assigned to u are the propositions that are known to be true in state u. The forcing relation $\mathfrak{M}, w \Vdash \varphi$ then extends this relationship to arbitrary formulas in the language; read $\mathfrak{M}, w \Vdash \varphi$ as "φ is true in state w." The relationship is defined inductively, as follows:

1. $\mathfrak{M}, w \Vdash p_i$ iff p_i is one of the propositional variables assigned to w.

2. $\mathfrak{M}, w \nVdash \bot$.

3. $\mathfrak{M}, w \Vdash (\varphi \wedge \psi)$ iff $\mathfrak{M}, w \Vdash \varphi$ and $\mathfrak{M}, w \Vdash \psi$.

4. $\mathfrak{M}, w \Vdash (\varphi \vee \psi)$ iff $\mathfrak{M}, w \Vdash \varphi$ or $\mathfrak{M}, w \Vdash \psi$.

5. $\mathfrak{M}, w \Vdash (\varphi \rightarrow \psi)$ iff, whenever $w' \geq w$ and $\mathfrak{M}, w' \Vdash \varphi$, then $\mathfrak{M}, w' \Vdash \psi$.

It is a good exercise to try to show that $\neg(p \wedge q) \rightarrow (\neg p \vee \neg q)$ is not intuitionistically valid, by cooking up a Kripke structure that provides a counterexample.

11.6 Modal Logics

fol:byd:mod:sec Consider the following example of a conditional sentence:

> If Jeremy is alone in that room, then he is drunk and naked and dancing on the chairs.

This is an example of a conditional assertion that may be materially true but nonetheless misleading, since it seems to suggest that there is a stronger link between the antecedent and conclusion other than simply that either the antecedent is false or the consequent true. That is, the wording suggests that the claim is not only true in this particular world (where it may be trivially true, because Jeremy is not alone in the room), but that, moreover, the conclusion *would have* been true *had* the antecedent been true. In other words, one can take the assertion to mean that the claim is true not just in this world, but in any "possible" world; or that it is *necessarily* true, as opposed to just true in this particular world.

Modal logic was designed to make sense of this kind of necessity. One obtains modal propositional logic from ordinary propositional logic by adding a box operator; which is to say, if φ is a formula, so is $\Box\varphi$. Intuitively, $\Box\varphi$ asserts that φ is *necessarily* true, or true in any possible world. $\Diamond\varphi$ is usually taken to be an abbreviation for $\neg\Box\neg\varphi$, and can be read as asserting that φ is *possibly* true. Of course, modality can be added to predicate logic as well.

Kripke structures can be used to provide a semantics for modal logic; in fact, Kripke first designed this semantics with modal logic in mind. Rather than restricting to partial orders, more generally one has a set of "possible worlds," P, and a binary "accessibility" relation $R(x, y)$ between worlds. Intuitively, $R(p, q)$ asserts that the world q is compatible with p; i.e., if we are "in" world p, we have to entertain the possibility that the world could have been like q.

Modal logic is sometimes called an "intensional" logic, as opposed to an "extensional" one. The intended semantics for an extensional logic, like classical logic, will only refer to a single world, the "actual" one; while the semantics for an "intensional" logic relies on a more elaborate ontology. In addition to structureing necessity, one can use modality to structure other linguistic constructions, reinterpreting $\Box$ and $\Diamond$ according to the application. For example:

1. In provability logic, $\Box\varphi$ is read "φ is provable" and $\Diamond\varphi$ is read "φ is consistent."
2. In epistemic logic, one might read $\Box\varphi$ as "I know φ" or "I believe φ."
3. In temporal logic, one can read $\Box\varphi$ as "φ is always true" and $\Diamond\varphi$ as "φ is sometimes true."

One would like to augment logic with rules and axioms dealing with modality. For example, the system **S4** consists of the ordinary axioms and rules of propositional logic, together with the following axioms:

$$\Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$$
$$\Box\varphi \rightarrow \varphi$$
$$\Box\varphi \rightarrow \Box\Box\varphi$$

as well as a rule, "from φ conclude $\Box\varphi$." **S5** adds the following axiom:

$$\Diamond\varphi \rightarrow \Box\Diamond\varphi$$

Variations of these axioms may be suitable for different applications; for example, S5 is usually taken to characterize the notion of logical necessity. And the nice thing is that one can usually find a semantics for which the derivation system is sound and complete by restricting the accessibility relation in the Kripke structures in natural ways. For example, **S4** corresponds to the class of Kripke structures in which the accessibility relation is reflexive and transitive. **S5** corresponds to the class of Kripke structures in which the accessibility relation is *universal*, which is to say that every world is accessible from every other; so $\Box\varphi$ holds if and only if φ holds in every world.

11.7 Other Logics

fol:byd:oth:sec As you may have gathered by now, it is not hard to design a new logic. You too can create your own a syntax, make up a deductive system, and fashion a semantics to go with it. You might have to be a bit clever if you want the derivation system to be complete for the semantics, and it might take some effort to convince the world at large that your logic is truly interesting. But, in return, you can enjoy hours of good, clean fun, exploring your logic's mathematical and computational properties.

Recent decades have witnessed a veritable explosion of formal logics. Fuzzy logic is designed to model reasoning about vague properties. Probabilistic logic is designed to model reasoning about uncertainty. Default logics and nonmonotonic logics are designed to model defeasible forms of reasoning, which is to say, "reasonable" inferences that can later be overturned in the face of new information. There are epistemic logics, designed to model reasoning about knowledge; causal logics, designed to model reasoning about causal relationships; and even "deontic" logics, which are designed to model reasoning about moral and ethical obligations. Depending on whether the primary motivation for introducing these systems is philosophical, mathematical, or computational, you may find such creatures studies under the rubric of mathematical logic, philosophical logic, artificial intelligence, cognitive science, or elsewhere.

The list goes on and on, and the possibilities seem endless. We may never attain Leibniz' dream of reducing all of human reason to calculation—but that can't stop us from trying.

Photo Credits

Bibliography

Magnus, P. D., Tim Button, J. Robert Loftis, Aaron Thomas-Bolduc, Robert Trueman, and Richard Zach. 2021. *Forall* x*: Calgary. An Introduction to Formal Logic*. Calgary: Open Logic Project, f21 ed. URL `https://forallx.openlogicproject.org/`.

Smullyan, Raymond M. 1968. *First-Order Logic*. New York, NY: Springer. Corrected reprint, New York, NY: Dover, 1995.

Zuckerman, Martin M. 1973. Formation sequences for propositional formulas. *Notre Dame Journal of Formal Logic* 14(1): 134–138.