Chapter 1

Tableau Method in Predicate Logic

In this chapter, we will demonstrate how to generalize the *method of analytic tableau* from propositional logic to predicate logic.[1] The method operates similarly but must be able to handle *quantifiers*.

1.1 Informal Introduction

In this section, we will informally introduce the tableau method in predicate logic. Formal definitions will follow later. We will start with two examples illustrating how the tableau method works in predicate logic and how it deals with quantifiers.

Example 1.1.1. Figure 1.1.1 shows two tableaux. These are tableau proofs (in logic, i.e., from the empty theory) of the *sentences* $(\exists x)\neg P(x) \to \neg(\forall x)P(x)$ (right) and $\neg(\forall x)P(x) \to (\exists x)\neg P(x)$ (left) in the language $L = \langle P \rangle$ (without equality), where P is a unary relation symbol. The symbol c_0 is an *auxiliary constant symbol* that we add to the language during the construction of the tableau.

Entries

Formulas in entries must always be *sentences* because we need them to have a *truth value* in a given model (independently of a variable assignment). This is not a substantial restriction; if we want to prove that a formula φ holds in a theory T, we can first replace the formula φ and all the axioms of T with their *general closures* (i.e., universally quantify all free variables). This yields a *closed* theory T' and a sentence φ', and it holds that $T' \models \varphi'$ if and only if $T \models \varphi$.

Quantifiers

The reduction of entries works the same way, using the same atomic tableaux for logical connectives (see Table **??**, where instead of propositions, φ, ψ are sentences). However, we must add four new atomic tableaux for T/F and universal/existential quantifiers. These entries can be divided into two types:

- Type "*witness*": entries of the form $\mathrm{T}(\exists x)\varphi(x)$ and $\mathrm{F}(\forall x)\varphi(x)$

[1] At this point, it is helpful to revisit the tableau method in propositional logic, see Chapter **??**.

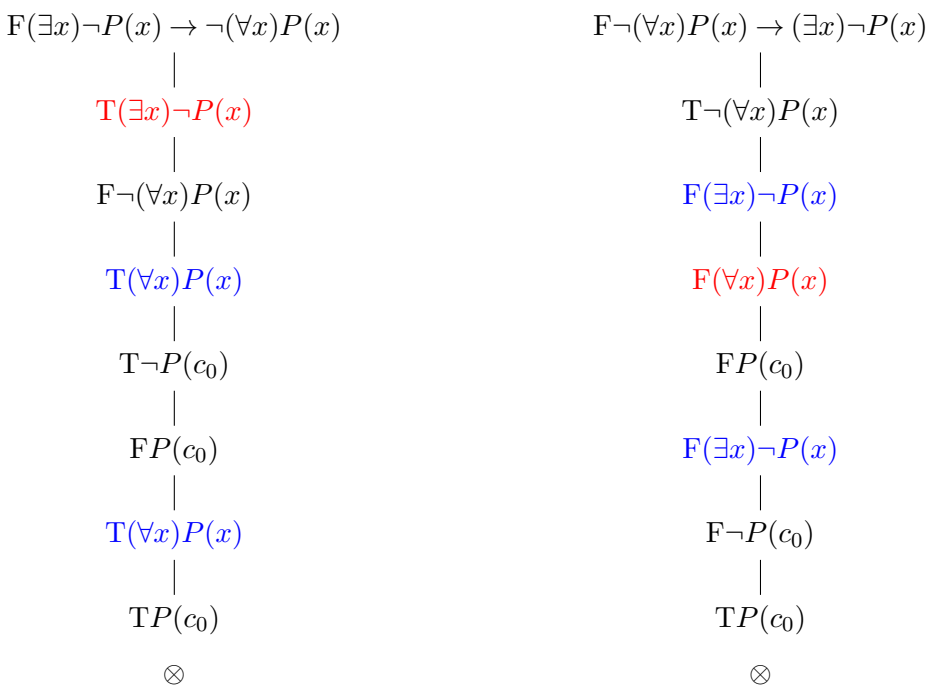

Figure 1.1: Example tableaux. Entries of the 'witness' type are shown in red, entries of the 'everyone' type in blue.

- Type "*everyone*": entries of the form $\mathrm{T}(\forall x)\varphi(x)$ and $\mathrm{F}(\exists x)\varphi(x)$

Examples can be seen in the tableaux in Figure 1.1.1 ('witnesses' are in red, 'everyone' in blue).

We cannot simply remove the quantifier because the resulting formula $\varphi(x)$ would not be a sentence. Instead, simultaneously with removing the quantifier, we *substitute* a *ground term* for x, resulting in the new entry being the *sentence* $\varphi(x/t)$. The ground term t we substitute depends on whether it is a "witness" or "everyone" type entry.

Auxiliary Constant Symbols

The language L of the theory T in which we are proving is extended with a countable number of *new (auxiliary) constant symbols* $C = \{c_0, c_1, c_2, \dots\}$ (but we will also write $c, d, \dots$), and the resulting extended language is denoted L_C. Thus, ground terms in the language L_C exist even if the original language L has no constants. When constructing the tableau, we always have some *new*, previously *unused* (neither in the theory nor in the constructed tableau) auxiliary constant symbol $c \in C$ available.

Witnesses

When reducing a "witness" type entry, we substitute for the variable one of these new, auxiliary symbols that *has not yet been used on the given branch*. For an entry $\mathrm{T}(\exists x)\varphi(x)$, we thus get $\mathrm{T}\varphi(x/c)$. This constant symbol c will represent (some) element that satisfies the formula (or refutes it, in the case of an entry of the form $\mathrm{F}(\forall x)\varphi(x)$). Compare this with the Theorem on Constants (Theorem **??**). It is important that the symbol c has not yet been

used on the branch or in the theory. Typically, we then use "everyone" type entries to learn what must *hold about this witness*.

In Figure 1.1.1, we see an example: the entry $\mathrm{T}(\exists x)\neg P(x)$ in the left tableau is reduced, with its reduction resulting in the entry $\mathrm{T}\neg P(c_0)$; $c_0 \in C$ is an auxiliary symbol that has not yet appeared on the branch (and is the first such). Similarly for the entry $\mathrm{F}(\forall x)P(x)$ and $\mathrm{F}P(c_0)$ in the right tableau.

Everyone

When reducing an "everyone" type entry, we substitute for the variable x any *ground term* t of the extended language L_C. From an entry of the form $\mathrm{T}(\forall x)\varphi(x)$, we thus get the entry $\mathrm{T}\varphi(x/t)$.

However, for a branch to be *finished*, it must contain entries $\mathrm{T}\varphi(x/t)$ for *all* ground L_C-terms t. (We must 'use' everything the entry $\mathrm{T}(\forall x)\varphi(x)$ 'says'.) The same applies for an entry of the form $\mathrm{F}(\exists x)\varphi(x)$.

In propositional logic, we used the convention of omitting the roots of atomic tableaux when appending them (otherwise, we would repeat the same entry twice on the branch). In predicate logic, we will use the same convention but *with the exception of 'everyone' type entries*. For type 'everyone', we will write the root of the appended atomic tableau as well. Why do we do this? To remind ourselves that we are not yet done with this entry and that we must append atomic tableaux with other ground terms.

In Figure 1.1.1, the entry $\mathrm{T}(\forall x)P(x)$ in the left tableau is *not reduced*. Its *first occurrence* (4th node from the top) has been reduced, substituting the term $t = c_0$, resulting in $\varphi(x/t) = P(c_0)$. We have appended an atomic tableau consisting of the same entry at the root $\mathrm{T}(\forall x)P(x)$, which we *do write* in the tableau, and the entry $\mathrm{T}P(c_0)$ below it. While the *first occurrence* of the entry $\mathrm{T}(\forall x)P(x)$ is thus reduced, the *second occurrence* (7th node from the top) is not. Similarly for the entry $\mathrm{F}(\exists x)\neg P(x)$ in the right tableau.

This somewhat technical approach to defining the *reducedness* of (occurrences of) 'everyone' type entries will be useful in defining a *systematic tableau*.

Language

We will assume that the language L is *countable*.[2] As a consequence, any L-theory T has only countably many axioms, and there are also only countably many ground terms in the language L_C. This restriction is necessary because every tableau, even an infinite one, has only countably many entries, and we must be able to use all the axioms of the given theory and substitute all the ground terms of the language L_C.

Initially, we will also assume that the language is *without equality*, which is the simpler case. The issue with equality is that *the tableau* is a purely syntactic object, but *equality* has a special semantic meaning: it must be interpreted as the identity relation in every model. How to adapt the method for languages with equality will be discussed later.

1.2 Formal Definitions

In this section, we define all the notions needed for the tableau method for languages without equality. We will return to languages with equality in Section 1.3.

[2] This is not a significant limitation from the perspective of computational logic.

Let L be a *countable* language without equality. Denote by L_C the extension of the language L with countably many new *auxiliary* constant symbols $C = \{c_i \mid i \in \mathbb{N}\}$. Let us choose some numbering of the ground terms of the language L_C, denoted by $\{t_i \mid i \in \mathbb{N}\}$.

Let T be some L-theory and φ an L-sentence.

1.2.1 Atomic Tableaux

An *entry* is a label $\mathrm{T}\varphi$ or $\mathrm{F}\varphi$, where φ is some L_C-sentence. Entries of the form $\mathrm{T}(\exists x)\varphi(x)$ and $\mathrm{F}(\forall x)\varphi(x)$ are of the *'witness' type*, while entries of the form $\mathrm{T}(\forall x)\varphi(x)$ and $\mathrm{F}(\exists x)\varphi(x)$ are of the *'everyone' type*.

Atomic tableaux are labeled trees shown in Tables 1.1 and 1.2.

	$\neg$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$
True	$\mathrm{T}\neg\varphi$ | $\mathrm{F}\varphi$	$\mathrm{T}\varphi \wedge \psi$ | $\mathrm{T}\varphi$ | $\mathrm{T}\psi$	$\mathrm{T}\varphi \vee \psi$ / \\ $\mathrm{T}\varphi$ $\mathrm{T}\psi$	$\mathrm{T}\varphi \rightarrow \psi$ / \\ $\mathrm{F}\varphi$ $\mathrm{T}\psi$	$\mathrm{T}\varphi \leftrightarrow \psi$ / \\ $\mathrm{T}\varphi$ $\mathrm{F}\varphi$ | | $\mathrm{T}\psi$ $\mathrm{F}\psi$
False	$\mathrm{F}\neg\varphi$ | $\mathrm{T}\varphi$	$\mathrm{F}\varphi \wedge \psi$ / \\ $\mathrm{F}\varphi$ $\mathrm{F}\psi$	$\mathrm{F}\varphi \vee \psi$ | $\mathrm{F}\varphi$ | $\mathrm{F}\psi$	$\mathrm{F}\varphi \rightarrow \psi$ | $\mathrm{T}\varphi$ | $\mathrm{F}\psi$	$\mathrm{F}\varphi \leftrightarrow \psi$ / \\ $\mathrm{T}\varphi$ $\mathrm{F}\varphi$ | | $\mathrm{F}\psi$ $\mathrm{T}\psi$

Table 1.1: Atomic tableaux for logical connectives; φ and ψ are any L_C-sentences.

	$\forall$	$\exists$
True	$\mathrm{T}(\forall x)\varphi(x)$ | $\mathrm{T}\varphi(x/t_i)$	$\mathrm{T}(\exists x)\varphi(x)$ | $\mathrm{T}\varphi(x/c_i)$
False	$\mathrm{F}(\forall x)\varphi(x)$ | $\mathrm{F}\varphi(x/c_i)$	$\mathrm{F}(\exists x)\varphi(x)$ | $\mathrm{F}\varphi(x/t_i)$

Table 1.2: Atomic tableaux for quantifiers; φ is an L_C-sentence, x a variable, t_i any ground L_C-term, $c_i \in C$ is a new auxiliary constant symbol (not yet occurring on the given branch of the constructed tableau).

1.2.2 Tableau Proof

The definitions in this section are almost identical to the corresponding definitions from propositional logic. The main technical issue is how to define the reducedness of 'everyone' type entries on a branch of the tableau: we want *all* ground L_C-terms t_i to be substituted for the variable.

Definition 1.2.1 (Tableau)**.** A *finite tableau from theory* T is an ordered, labeled tree constructed by applying finitely many of the following rules:

- A single-node tree labeled with any entry is a tableau from theory T,
- For any entry E on any branch B, we can append an atomic tableau for the entry E to the end of branch B, provided that if E is of 'witness' type, we use only an auxiliary constant symbol $c_i \in C$ not yet appearing on branch B (for 'everyone' type entries, we can use any ground L_C-term t_i),
- We can append the entry $\mathrm{T}\alpha$ for any axiom $\alpha \in T$ to the end of any branch.

A *tableau from theory* T can be either finite or *infinite*: in that case it is constructed in countably many steps. It can be formally expressed as the union $\tau = \bigcup_{i \geq 0} \tau_i$, where τ_i are finite tableaux from T, τ_0 is a single-node tableau, and τ_{i+1} is constructed from τ_i in one step.[3]

A tableau *for entry* E is a tableau with the entry E at its root.

Recall the convention that if E is *not* of the 'everyone' type, we omit the root of the atomic tableau (since the node with entry E is already in the tableau).

Exercise 1.1. Show step-by-step how the tableaux from Figure 1.1.1 were constructed.

Definition 1.2.2 (Tableau Proof)**.** A *tableau proof* of a sentence φ from a theory T is a *contradictory* tableau from theory T with the entry $\mathrm{F}\varphi$ at the root. If such a proof exists, then φ is *(tableau) provable* from T, denoted by $T \vdash \varphi$. (We also define a *tableau refutation* as a contradictory tableau with $\mathrm{T}\varphi$ at the root. If such a proof exists, then φ is *(tableau) refutable* from T, i.e., $T \vdash \neg\varphi$ holds.)

- A tableau is *contradictory* if each of its branches is contradictory.
- A branch is *contradictory* if it contains entries $\mathrm{T}\psi$ and $\mathrm{F}\psi$ for some sentence ψ, otherwise it is *non-contradictory*.
- A tableau is *finished* if each of its branches is finished.
- A branch is *finished* if
  - it is contradictory, or
  - each entry on this branch is *reduced* and the branch contains the entry $\mathrm{T}\alpha$ for each axiom $\alpha \in T$.
- An entry E is *reduced* on a branch B passing through this entry if

[3] Union because in each step, we add new nodes to the tableau, so τ_i is a subtree of τ_{i+1}.

- it is of the form Tψ or Fψ for an *atomic sentence* ψ (i.e., $R(t_1, \ldots, t_n)$, where t_i are *ground* L_C-terms), or
- it is not of the 'everyone' type and appears on B as the root of an atomic tableau[4] (i.e., typically, the entry has already been 'developed' on B during the tableau construction), or
- it is of the 'everyone' type and all its *occurrences* on B are reduced on the branch B.

- An occurrence of an 'everyone' type entry E on branch B is *i-th* if it has exactly $i-1$ predecessors labeled with this entry on B, and the i-th occurrence is *reduced* on B if
  - the entry E has the $(i+1)$-th occurrence on B, and at the same time
  - the branch B contains the entry T$\varphi(x/t_i)$ (if $E = \mathrm{T}(\forall x)\varphi(x)$) or F$\varphi(x/t_i)$ (if $E = \mathrm{F}(\exists x)\varphi(x)$), where t_i is the i-th ground L_C-term.[5]

Note that if an 'everyone' type entry is reduced on some branch, it must have infinitely many occurrences on that branch, and we must have used all possibilities, i.e., all ground L_C-terms, in the substitutions.

Example 1.2.3. As an example, let us construct tableau proofs *in logic* (from the empty theory) of the following sentences:

(a) $(\forall x)(P(x) \to Q(x)) \to ((\forall x)P(x) \to (\forall x)Q(x))$, where P, Q are unary relation symbols.

(b) $(\forall x)(\varphi(x) \wedge \psi(x)) \leftrightarrow ((\forall x)\varphi(x) \wedge (\forall x)\psi(x))$, where $\varphi(x), \psi(x)$ are arbitrary formulas with a single free variable x.

The resulting tableaux are in Figures 1.2 and 1.3. The pairs of contradictory entries are shown in red. Consider how the tableaux were constructed step-by-step.

1.2.3 Systematic Tableau and Finiteness of Proofs

In Section **??**, we showed that if we do not extend contradictory branches (and there is no reason to do that), then a contradictory tableau, in particular a tableau proof, will always be finite. The same argument applies to predicate logic.

Corollary 1.2.4 (Finiteness of Proofs)**.** *If $T \vdash \varphi$, then there also exists a* finite *tableau proof of φ from T.*

Proof. The same as in propositional logic, see the proof of Corollary **??**. □

In the same section, we also showed the construction of the *systematic tableau*. This too can be easily adapted to predicate logic. We must ensure that we eventually reduce each entry, use every axiom, and now, in predicate logic, also substitute every L_C-term t_i for the variable in entries of type 'everyone'.

[4] Even though by convention we do not write this root.

[5] I.e., (typically) we have already substituted the term t_i for x.

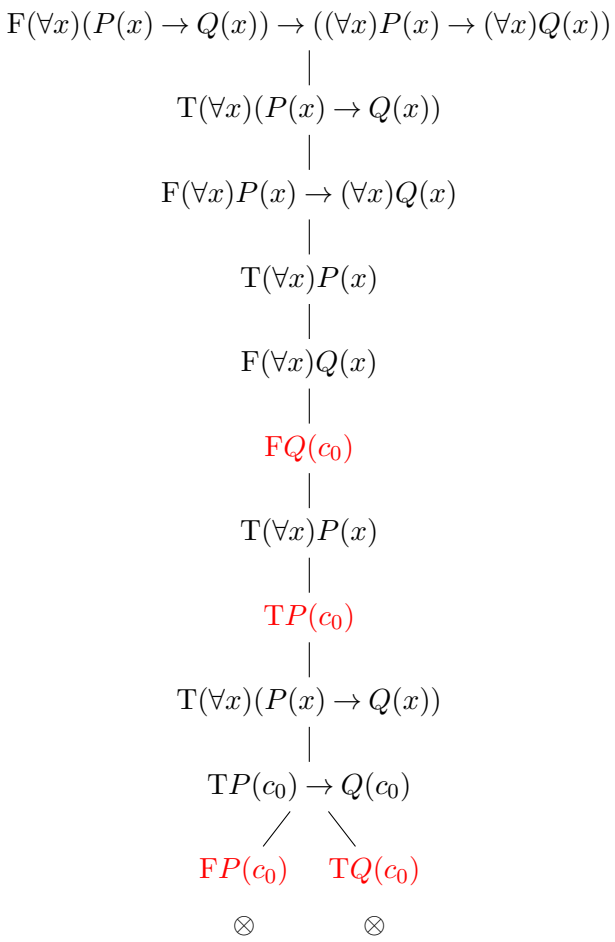

Figure 1.2: Tableau proof from Example 1.2.3 (a).

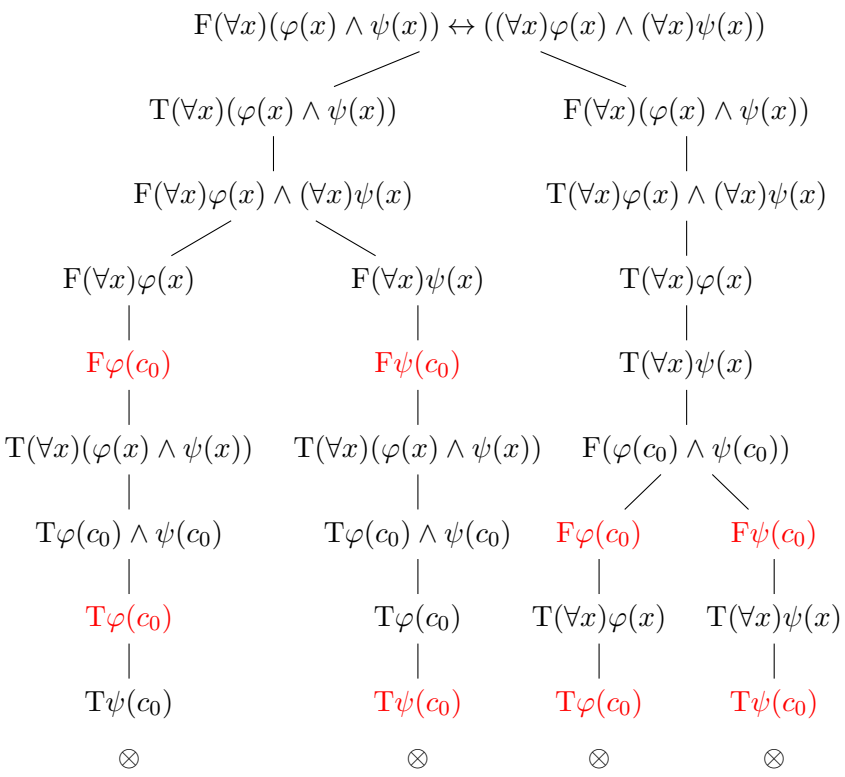

Figure 1.3: Tableau proof from Example 1.2.3 (b). The constant c_0 can be used as *new* in all three cases. It suffices that it does not yet occur *on the given branch*.

Definition 1.2.5. Let R be an entry and $T = \{\alpha_0, \alpha_1, \alpha_2, \dots\}$ a theory. The *systematic tableau* from theory T for entry R is the tableau $\tau = \bigcup_{i \geq 0} \tau_i$, where τ_0 is a single-node tableau with entry R, and for each $i \geq 0$:

Let E be an entry at the leftmost node v at the smallest level of the tableau τ_i that is not reduced on some non-contradictory branch passing through E (or, if the entry is of the 'everyone' type, its *occurrence* in this node is not reduced). Then τ_i' is the tableau constructed from τ_i by appending an atomic tableau for E to each non-contradictory branch passing through v, where

- If E is of the type 'everyone' and has its k-th occurrence in the node v, then we substitute the k-th L_C-term t_k for the variable.
- If E is of the 'witness' type, then on the given branch B, we substitute $c_i \in C$ with the smallest possible i (such that c_i does not yet occur on B).

Otherwise, if such an entry E and node v do not exist, i.e., all entries are reduced, we define $\tau_i' = \tau_i$.

The tableau τ_{i+1} is then the tableau constructed from τ_i' by appending $\mathrm{T}\alpha_i$ to each non-contradictory branch of τ_i', if $i \leq |T|$. Otherwise (if T is finite and we have used all the axioms), we skip this step and define $\tau_{i+1} = \tau_i'$.

Just as in propositional logic, it holds that the systematic tableau is always finished and provides a finite proof:

Lemma 1.2.6. *The systematic tableau is finished.*

Proof. Similar to the proof in propositional logic (Lemma **??**). For 'everyone' type entries, note that we reduce the k-th occurrence when we encounter it in the construction: by appending a node with the $(k+1)$-th occurrence and substituting the k-th L_C-term t_k. □

Corollary 1.2.7 (Systematicity of Proofs)**.** *If $T \vdash \varphi$, then the systematic tableau is (a finite) tableau proof of φ from T.*

Proof. The same as the proof in propositional logic (Corollary **??**). □

1.3 Languages with Equality

Now we will show how to apply the tableau method to languages with equality. What is equality? In mathematics, it can mean different relations in different contexts. Does $1 + 0 = 0 + 1$ hold? If we are talking about integers, then yes, but if we mean arithmetic expressions (or e.g., terms in the language of fields), then the left and right sides are not equal: they are different expressions. [6]

Imagine we have a theory T in a language with equality containing constant symbols c_1, c_2, a unary function symbol f, and a unary relation symbol P. Let us have some finished tableau from this theory, and in it a non-contradictory branch, on which we find the entry $\mathrm{T}c_1 = c_2$. We aim to construct a *canonical model* $\mathcal{A}$ for this branch, similar to propositional logic. The entry will mean that in the canonical model we have $c_1^{\mathcal{A}} =^{\mathcal{A}} c_2^{\mathcal{A}}$, i.e., $(c_1^{\mathcal{A}}, c_2^{\mathcal{A}}) \in =^{\mathcal{A}}$. But this is not enough, we also want, for example:

[6] Similarly, $t_1 = t_2$ in Prolog does not mean they are the same term but that the terms t_1 and t_2 are *unifiable*, see the next chapter, Section **??**.

- $c_2^{\mathcal{A}} =^{\mathcal{A}} c_1^{\mathcal{A}}$,
- $f^{\mathcal{A}}(c_1^{\mathcal{A}}) =^{\mathcal{A}} f^{\mathcal{A}}(c_2^{\mathcal{A}})$,
- $c_1^{\mathcal{A}} \in P^{\mathcal{A}}$ if and only if $c_2^{\mathcal{A}} \in P^{\mathcal{A}}$.

In general, we want the relation $=^{\mathcal{A}}$ to be a so-called *congruence*,[7] i.e., an equivalence that behaves 'well' with respect to the functions and relations of the structure $\mathcal{A}$. We achieve this by adding so-called *axioms of equality* to the theory T, which enforce these properties, and construct the tableau from the resulting theory T^*.

In the model $\mathcal{A}$, the relation $=^{\mathcal{A}}$ will then be a congruence. But this is not enough for us; we want equality to be *identity*, i.e., $(a, b) \in =^{\mathcal{A}}$ only if a and b are the same element of the domain. We achieve this by identifying all $=^{\mathcal{A}}$-equivalent elements into a single element. This construction is called a *quotient (structure)* by the congruence $=^{\mathcal{A}}$.[8] Now we formalize these concepts.

Definition 1.3.1 (Congruence)**.** Let $\sim$ be an equivalence on the set A, $f\colon A^n \to A$ be a function, and $R \subseteq A^n$ be a relation. We say that $\sim$ is

- *a congruence for the function* f if for all $a_i, b_i \in A$ such that $a_i \sim b_i$ $(1 \leq i \leq n)$ it holds that $f(a_1, \ldots, a_n) \sim f(b_1, \ldots, b_n)$,
- *a congruence for the relation* R if for all $a_i, b_i \in A$ such that $a_i \sim b_i$ $(1 \leq i \leq n)$ it holds that $(a_1, \ldots, a_n) \in R$ if and only if $(b_1, \ldots, b_n) \in R$.

A congruence of the structure $\mathcal{A}$ is an equivalence $\sim$ on the set A that is a congruence for all functions and relations of $\mathcal{A}$.

Definition 1.3.2 (Quotient Structure)**.** Let $\mathcal{A}$ be a structure and $\sim$ be its congruence. The *quotient structure* of $\mathcal{A}$ by $\sim$ is the structure $\mathcal{A}/_\sim$ in the same language, whose universe $A/_\sim$ is the set of all equivalence classes of A by $\sim$, and whose functions and relations are defined *using representatives*, i.e.:

- $f^{\mathcal{A}/_\sim}([a_1]_\sim, \ldots, [a_n]_\sim) = [f^{\mathcal{A}}(a_1, \ldots, a_n)]_\sim$, for each ($n$-ary) function symbol f, and
- $R^{\mathcal{A}/_\sim}([a_1]_\sim, \ldots, [a_n]_\sim)$ if and only if $R^{\mathcal{A}}(a_1, \ldots, a_n)$, for each ($n$-ary) relation symbol R.

Definition 1.3.3 (Axioms of Equality)**.** The *axioms of equality* for a language L with equality are as follows:

(i) $x = x$,

(ii) $x_1 = y_1 \wedge \cdots \wedge x_n = y_n \to f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$ for every n-ary function symbol f of the language L,

(iii) $x_1 = y_1 \wedge \cdots \wedge x_n = y_n \to (R(x_1, \ldots, x_n) \to R(y_1, \ldots, y_n))$ for every n-ary relation symbol R of the language L *including the equality symbol*.

[7] The name comes from *congruence modulo n*, which is a congruence in this sense on the set of all integers, e.g., it satisfies: $a + b \equiv c + d \pmod{n}$ whenever $a \equiv c \pmod{n}$ and $b \equiv d \pmod{n}$.

[8] Just as the group $\mathbb{Z}_n$ is a quotient structure of the group $\mathbb{Z}$ over $\equiv \pmod{n}$; for example, the element $2 \in \mathbb{Z}_n$ represents the set of all integers whose remainder after division by n is 2.

Exercise 1.2. The first of the axioms of equality expresses the reflexivity of the relation $=^{\mathcal{A}}$. What about symmetry and transitivity? Show that they follow from axiom (iii) for the equality symbol $=$.

Thus, from axioms (i) and (iii) it follows that the relation $=^{\mathcal{A}}$ is an equivalence on A, and axioms (ii) and (iii) express that $=^{\mathcal{A}}$ is a congruence of $\mathcal{A}$. In the tableau method for a language with equality, we implicitly add all the axioms of equality:

Definition 1.3.4 (Tableau Proof with Equality)**.** If T is a theory in a language L with equality, then denote as T^* the extension of the theory T by the general closures[9] of the axioms of equality for the language L. A *tableau proof* from the theory T is a *tableau proof* from T^*, similarly for tableau refutation (and generally any tableau).

The following simple observation holds:

Observation 1.3.5. *If $\mathcal{A} \models T^*$, then it holds that $\mathcal{A}/_{=^{\mathcal{A}}} \models T^*$, and in the structure $\mathcal{A}/_{=^{\mathcal{A}}}$ the equality symbol is interpreted as identity. On the other hand, in any model where the equality symbol is interpreted as identity, the axioms of equality hold.*

We will use this observation in the construction of the *canonical model*, which we will need to prove the Completeness Theorem. But first, we will prove the Soundness Theorem.

1.4 Soundness and Completeness

In this section, we will prove that the tableau method is sound and complete in predicate logic as well. The proofs of both theorems have the same structure as in propositional logic, differing only in the implementation details.

1.4.1 Soundness Theorem

A model (structure) $\mathcal{A}$ *agrees* with an entry E if $E = \mathrm{T}\varphi$ and $\mathcal{A} \models \varphi$, or $E = \mathrm{F}\varphi$ and $\mathcal{A} \not\models \varphi$. Further, $\mathcal{A}$ agrees with a branch B if it agrees with every entry on that branch.

First, we show an auxiliary lemma analogous to Lemma **??**:

Lemma 1.4.1. *If a model $\mathcal{A}$ of the theory T agrees with the entry at the root of a tableau from the theory T (in the language L), then $\mathcal{A}$ can be expanded to the language L_C such that it agrees with some branch in the tableau.*

Note that it suffices to expand $\mathcal{A}$ by those new constants $c^{\mathcal{A}}$ where c appears on the branch B. Other constant symbols can be interpreted arbitrarily.

Proof. Let $\tau = \bigcup_{i\geq 0} \tau_i$ be a tableau from the theory T and $\mathcal{A} \in \mathrm{M}_L(T)$ a model agreeing with the root of τ, i.e., with the (single-element) branch B_0 in the (single-element) τ_0.

By induction on i, we find a sequence of branches B_i and expansions $\mathcal{A}_i$ of the model $\mathcal{A}$ by constants $c^{\mathcal{A}} \in C$ appearing on B_i such that B_i is a branch in the tableau τ_i agreeing with the model $\mathcal{A}_i$, B_{i+1} is an extension of B_i, and $\mathcal{A}_{i+1}$ is an expansion of $\mathcal{A}_i$ (they might be the same structure). The desired branch of the tableau τ is then $B = \bigcup_{i\geq 0} B_i$. The expansion of the model $\mathcal{A}$ to the language L_C is obtained as the 'limit' of the expansions $\mathcal{A}_i$, i.e., if a symbol $c \in C$ appears on B, it appears on some branch B_i and we interpret it the same as in $\mathcal{A}_i$ (other auxiliary symbols are interpreted arbitrarily).

[9] Because for the tableau method we need to have *sentences*.

- If τ_{i+1} was created from τ_i without extending the branch B_i, we define $B_{i+1} = B_i$ and $\mathcal{A}_{i+1} = \mathcal{A}_i$.
- If τ_{i+1} was created from τ_i by adding the entry $\mathrm{T}\alpha$ (for some axiom $\alpha \in T$) to the end of the branch B_i, we define B_{i+1} as this extended branch and $\mathcal{A}_{i+1} = \mathcal{A}_i$ (we did not add any new auxiliary constant symbol). Since $\mathcal{A}_{i+1}$ is a model of T, the axiom α holds in it, so it agrees with the new entry $\mathrm{T}\alpha$.
- Suppose τ_{i+1} was created from τ_i by appending an atomic tableau for some entry E to the end of the branch B_i. Since the model $\mathcal{A}_i$ agrees with the entry E (which lies on the branch B_i), it also agrees with the root of the appended atomic tableau.
  - If we appended an atomic tableau for a logical connective, we set $\mathcal{A}_{i+1} = \mathcal{A}_i$ (we did not add a new auxiliary symbol). Since $\mathcal{A}_{i+1}$ agrees with the root of the atomic tableau, it also agrees with one of its branches (just as in propositional logic); we define B_{i+1} as the extension of B_i by this branch.
  - If the entry E is of the type 'witness': If $E = \mathrm{T}(\exists x)\varphi(x)$, then $\mathcal{A}_i \models (\exists x)\varphi(x)$, so there exists $a \in A$ such that $\mathcal{A}_i \models \varphi(x)[e(x/a)]$. We define the branch B_{i+1} as the extension of B_i by the newly added entry $\mathrm{T}\varphi(x/c)$ and the model $\mathcal{A}_{i+1}$ as the expansion of $\mathcal{A}_i$ by the constant $c^A = a$. The case $E = \mathrm{F}(\forall x)\varphi(x)$ is analogous.
  - If the entry E is of the type 'everyone', we define the branch B_{i+1} as the extension of B_i by the atomic tableau. The newly added entry is $\mathrm{T}\varphi(x/t)$ or $\mathrm{F}\varphi(x/t)$ for some L_C-term t. Suppose it is the first of these two possibilities, for the second the proof is analogous. We define the model $\mathcal{A}_{i+1}$ as *any* expansion of $\mathcal{A}_i$ by the new constants appearing in t. Since $\mathcal{A}_i \models (\forall x)\varphi(x)$, it also holds $\mathcal{A}_{i+1} \models (\forall x)\varphi(x)$ and thus $\mathcal{A}_{i+1} \models \varphi(x/t)$; the model $\mathcal{A}_{i+1}$ thus agrees with the branch B_i.

□

Let us briefly recall the idea of the proof of the Soundness Theorem: If there existed a proof and at the same time a counterexample, the counterexample would have to agree with some branch of the proof, but they are all contradictory. The proof is thus almost the same as in propositional logic.

Theorem 1.4.2 (On Soundness)**.** *If a sentence φ is tableau provable from a theory T, then φ is valid in T, i.e., $T \vdash \varphi \;\Rightarrow\; T \models \varphi$.*

Proof. Assume for contradiction that $T \not\models \varphi$, i.e., there exists $\mathcal{A} \in \mathrm{M}(T)$ such that $\mathcal{A} \not\models \varphi$. Since $T \vdash \varphi$, there exists a contradictory tableau from T with $\mathrm{F}\varphi$ at the root. The model $\mathcal{A}$ agrees with $\mathrm{F}\varphi$, so by Lemma 1.4.1 it can be expanded to the language L_C such that the expansion agrees with some branch B. But all branches are contradictory. □

1.4.2 Completeness Theorem

Same as in propositional logic, we will show that a *non-contradictory* branch in a *finished* tableau proof provides a counterexample: a model of the theory T that agrees with the entry $\mathrm{F}\varphi$ at the root of the tableau, i.e., φ does not hold in it. There can be more such models, so we again define one specific, *canonical* model.

The model must have some domain. How do we obtain it from the tableau, which is a purely syntactic object? We use a standard trick in mathematics: we turn syntactic objects into semantic ones. In particular, we choose the set of all *ground terms* of the language L_C as the domain.[10] We understand these as finite strings. In the following exposition, we will sometimes (informally) write "t" instead of the term t to emphasize that at that place we view t as a string of characters, not e.g., as a term function to be evaluated.[11]

Definition 1.4.3 (Canonical Model)**.** Let T be a theory in a language $L = \langle \mathcal{F}, \mathcal{R} \rangle$ and let B be a non-contradictory branch of some finished tableau from the theory T. Then the *canonical model* for B is the L_C-structure $\mathcal{A} = \langle A, \mathcal{F}^{\mathcal{A}} \cup C^{\mathcal{A}}, \mathcal{R}^{\mathcal{A}} \rangle$ defined as follows:

If the language L is without equality, then:

- The domain A is the set of all ground L_C-terms.
- For each n-ary relation symbol $R \in \mathcal{R}$ and "s_1", ..., "s_n" from A:

  ("s_1", ..., "s_n") $\in R^{\mathcal{A}}$ if and only if the branch B contains the entry $\mathrm{T}R(s_1, \ldots, s_n)$
- For each n-ary function symbol $f \in \mathcal{F}$ and "s_1", ..., "s_n" from A:

  $$f^{\mathcal{A}}(\text{“}s_1\text{”}, \ldots, \text{“}s_n\text{”}) = \text{“}f(s_1, \ldots, s_n)\text{”}$$

  In particular, for a constant symbol c, we have $c^{\mathcal{A}} =$ "c".

Thus, we interpret the function $f^{\mathcal{A}}$ as the 'creation' of a new term from the symbol f and the input terms (strings).

Let L be a language with equality. Recall that our tableau is now from the theory T^*, i.e., from the extension of T by the axioms of equality for L. First, we create the canonical model $\mathcal{D}$ for the branch B as if L were without equality (its domain D is thus the set of all ground L_C-terms). Then we define the relation $=^D$ just like for other relation symbols:

$$\text{“}s_1\text{”} =^D \text{“}s_2\text{”} \text{ if and only if the branch } B \text{ contains the entry } \mathrm{T}s_1 = s_2$$

The *canonical model* for B is then defined as the quotient structure $\mathcal{A} = \mathcal{D}/_{=^D}$.

As follows from the discussion in Section 1.3, the relation $=^D$ is indeed a congruence of the structure $\mathcal{D}$, so the definition is sound, and the relation $=^{\mathcal{A}}$ is the identity on A. The following simple observation holds:

Observation 1.4.4. *For any formula φ we have $\mathcal{D} \models \varphi$ (where the symbol $=$ is interpreted as the binary relation $=^D$) if and only if $\mathcal{A} = \mathcal{D}/_{=^D} \models \varphi$ (where $=$ is interpreted as identity).*

Note that in a language without equality, the canonical model is always countably infinite. In a language with equality, it can be finite, as we will see in the following examples.

[10] That is, terms constructed by applying the function symbols of the language L to the constant symbols of the language L (if it has any) and auxiliary constant symbols from C.

[11] Compare the arithmetic expression "1+1" and 1+1=2.

Example 1.4.5. First, let us show an example of a canonical model in a language without equality. Let $T = \{(\forall x)R(f(x))\}$ be a theory in the language $L = \langle R, f, d\rangle$ without equality, where R is a unary relation, f a unary function, and d a constant symbol. Let us find a counterexample showing that $T \not\models \neg R(d)$.

The systematic tableau from T with the entry $\mathrm{F}\neg R(d)$ at the root is not contradictory; it contains a single branch B, which is non-contradictory. (Construct the tableau yourself!) The canonical model for B is the L_C-structure $\mathcal{A} = \langle A, R^{\mathcal{A}}, f^{\mathcal{A}}, d^{\mathcal{A}}, c_0^{\mathcal{A}}, c_1^{\mathcal{A}}, c_2^{\mathcal{A}}, \dots\rangle$, its domain is

$$A = \{\text{“}d\text{”}, \text{“}f(d)\text{”}, \text{“}f(f(d))\text{”}, \dots, \text{“}c_0\text{”}, \text{“}f(c_0)\text{”}, \text{“}f(f(c_0))\text{”}, \dots, \text{“}c_1\text{”}, \text{“}f(c_1)\text{”}, \text{“}f(f(c_1))\text{”}, \dots\}$$

and the interpretations of the symbols are as follows:

- $d^{\mathcal{A}} = \text{“}d\text{”}$,
- $c_i^{\mathcal{A}} = \text{“}c_i\text{”}$ for all $i \in \mathbb{N}$,
- $f^{\mathcal{A}}(\text{“}d\text{”}) = \text{“}f(d)\text{”}$, $f^{\mathcal{A}}(\text{“}f(d)\text{”}) = \text{“}f(f(d))\text{”}$, ...
- $R^{\mathcal{A}} = A\backslash C = \{\text{“}d\text{”}, \text{“}f(d)\text{”}, \text{“}f(f(d))\text{”}, \dots, \text{“}f(c_0)\text{”}, \text{“}f(f(c_0))\text{”}, \dots, \text{“}f(c_1)\text{”}, \text{“}f(f(c_1))\text{”}, \dots\}$.

The reduct of the canonical model $\mathcal{A}$ to the original language L is then $\mathcal{A}' = \langle A, R^{\mathcal{A}}, f^{\mathcal{A}}, d^{\mathcal{A}}\rangle$.

Example 1.4.6. Now an example in a language with equality: Let $T = \{(\forall x)R(f(x)), (\forall x)(x = f(f(x)))\}$ be in the language $L = \langle R, f, d\rangle$ with equality. Again, let us find a counterexample showing that $T \not\models \neg R(d)$.

The systematic tableau from the theory T^* (i.e., from T extended by the axioms of equality for L) with the entry $\mathrm{F}\neg R(d)$ at the root contains a non-contradictory branch B. (Construct the tableau yourself!) First, we construct the canonical model $\mathcal{D}$ for this branch as if the language were without equality:

$$\mathcal{D} = \langle D, R^{\mathcal{D}}, f^{\mathcal{D}}, d^{\mathcal{D}}, c_0^{\mathcal{D}}, c_1^{\mathcal{D}}, c_2^{\mathcal{D}}, \dots\rangle$$

where D is the set of all ground L_C-terms. The relation $=^D$ is defined as if the symbol ‘=’ were an ‘ordinary’ relation symbol in L. It is a congruence of the structure $\mathcal{D}$, and it holds that $s_1 =^D s_2$ if and only if $s_1 = f(\cdots(f(s_2))\cdots)$ or $s_2 = f(\cdots(f(s_1))\cdots)$ for an even number of applications of f. Thus, we can choose terms with zero or one occurrence of the symbol f as representatives of individual equivalence classes:

$$D/_{=^D} = \{[\text{“}d\text{”}]_{=^D}, [\text{“}f(d)\text{”}]_{=^D}, [\text{“}c_0\text{”}]_{=^D}, [\text{“}f(c_0)\text{”}]_{=^D}, [\text{“}c_1\text{”}]_{=^D}, [\text{“}f(c_1)\text{”}]_{=^D}, \dots\}$$

The canonical model for the branch B is then the L_C-structure

$$\mathcal{A} = \mathcal{D}/_{=^D} = \langle A, R^{\mathcal{A}}, f^{\mathcal{A}}, d^{\mathcal{A}}, c_0^{\mathcal{A}}, c_1^{\mathcal{A}}, c_2^{\mathcal{A}}, \dots\rangle$$

where $A = D/_{=^D}$ and the interpretations of the symbols are as follows:

- $d^{\mathcal{A}} = [\text{“}d\text{”}]_{=^D}$,
- $c_i^{\mathcal{A}} = [\text{“}c_i\text{”}]_{=^D}$ for all $i \in \mathbb{N}$,
- $f^{\mathcal{A}}([\text{“}d\text{”}]_{=^D}) = [\text{“}f(d)\text{”}]_{=^D}$, $f^{\mathcal{A}}([\text{“}f(d)\text{”}]_{=^D}) = [\text{“}f(f(d))\text{”}]_{=^D} = [\text{“}d\text{”}]_{=^D}$, ...
- $R^{\mathcal{A}} = A = D/_{=^D}$.

The reduct of the canonical model $\mathcal{A}$ to the original language L is again $\mathcal{A}' = \langle A, R^{\mathcal{A}}, f^{\mathcal{A}}, d^{\mathcal{A}} \rangle$.

Exercise 1.3. (a) Construct a finished tableau with the entry $\mathrm{T}(\forall x)(\forall y)(x = y)$ at the root. Construct the canonical model for the (only, non-contradictory) branch of this tableau.

(b) Construct a finished tableau with the entry $\mathrm{T}(\forall x)(\forall y)(\forall z)(x = y \vee x = z \vee y = z)$ at the root. Construct canonical models for several of its non-contradictory branches, and compare them.

We are now ready to prove the Completeness Theorem. We again use the following auxiliary lemma, whose wording is exactly the same as Lemma **??** and whose proof differs only in technical details.

Lemma 1.4.7. *The canonical model for a (non-contradictory, finished) branch B agrees with B.*

Proof. First, consider languages without equality. We will show by induction on the structure of sentences in the entries that the canonical model $\mathcal{A}$ agrees with all entries E on the branch B.

The base case, i.e., when $\varphi = R(s_1, \ldots, s_n)$ is an atomic sentence, is simple: If there is an entry $\mathrm{T}\varphi$ on B, then $(s_1, \ldots, s_n) \in R^{\mathcal{A}}$ directly follows from the definition of the canonical model, so we have $\mathcal{A} \models \varphi$. If there is an entry $\mathrm{F}\varphi$ on B, then there is no entry $\mathrm{T}\varphi$ on B (since B is non-contradictory), $(s_1, \ldots, s_n) \notin R^{\mathcal{A}}$, and $\mathcal{A} \not\models \varphi$.

Now the induction step. We will discuss only some of the cases, the remaining cases are proved similarly.

For logical connectives, the proof is exactly the same as in propositional logic, for example, if $E = \mathrm{F}\varphi \wedge \psi$, then since E on B is reduced, there is an entry $\mathrm{F}\varphi$ or an entry $\mathrm{F}\psi$ on B. Thus, $\mathcal{A} \not\models \varphi$ or $\mathcal{A} \not\models \psi$, from which it follows that $\mathcal{A} \not\models \varphi \wedge \psi$ and $\mathcal{A}$ agrees with E.

If we have an entry of type 'everyone', for example, $E = \mathrm{T}(\forall x)\varphi(x)$ (the case $E = \mathrm{F}(\exists x)\varphi(x)$ is analogous), then there are entries $\mathrm{T}\varphi(x/t)$ for each ground L_C-term, i.e., for each element "t" $\in A$ on B. By the induction hypothesis, $\mathcal{A} \models \varphi(x/t)$ for each "t" $\in A$, so $\mathcal{A} \models (\forall x)\varphi(x)$.

If we have an entry of type 'witness', for example, $E = \mathrm{T}(\exists x)\varphi(x)$ (the case $E = \mathrm{F}(\forall x)\varphi(x)$ is agan analogous), then there is an entry $\mathrm{T}\varphi(x/c)$ for some "c" $\in A$ on B. By the induction hypothesis, $\mathcal{A} \models \varphi(x/c)$, so $\mathcal{A} \models (\exists x)\varphi(x)$.

If the language is with equality, we have the canonical model $\mathcal{A} = \mathcal{D}/_{=_D}$; the proof above applies to $\mathcal{D}$, and the rest follows from Observation 1.4.4. □

Exercise 1.4. Verify the remaining cases in the proof of Lemma 1.4.7.

The proof of the Completeness Theorem is also analogous to its version for propositional logic:

Theorem 1.4.8 (On Completeness)**.** *If a sentence φ is valid in a theory T, then it is tableau provable from T, i.e., $T \models \varphi \;\Rightarrow\; T \vdash \varphi$.*

Proof. We will show that any *finished* tableau from T with the entry $\mathrm{F}\varphi$ at the root must be contradictory. We prove this by contradiction: if such a tableau were not contradictory, there would be an non-contradictory (finished) branch B in it. Consider the canonical model $\mathcal{A}$ for this branch and denote its reduct to the language L as $\mathcal{A}'$. Since B is finished, it contains $\mathrm{T}\alpha$ for all axioms $\alpha \in T$. The model $\mathcal{A}$ agrees with all entries on B by Lemma 1.4.7, so it

satisfies all axioms, and we have $\mathcal{A}' \models T$. But since $\mathcal{A}$ also agrees with the entry $\mathrm{F}\varphi$ at the root, it holds that $\mathcal{A}' \not\models \varphi$, which means that $\mathcal{A}' \in \mathrm{M}_L(T) \setminus \mathrm{M}_L(\varphi)$, thus $T \not\models \varphi$, and this is a contradiction. Therefore, the tableau must have been contradictory, i.e., it must have been a tableau proof of φ from T. □

1.5 Consequences of Soundness and Completeness

Same as in propositional logic, the Soundness and Completeness Theorems together state that *provability* is the same as *validity*. This allows us to similarly formulate syntactic analogs of semantic concepts and properties.

The analogs of *consequences* are *theorems* of the theory T:

$$\mathrm{Thm}_L(T) = \{\varphi \mid \varphi \text{ is an } L\text{-sentence and } T \vdash \varphi\}$$

Corollary 1.5.1 (Provability = Validity)**.** *For any theory T and sentences φ, ψ, the following holds:*

- $T \vdash \varphi$ *if and only if* $T \models \varphi$
- $\mathrm{Thm}_L(T) = \mathrm{Csq}_L(T)$

For example, it holds that:

- A theory is *inconsistent* if the contradiction is provable in it (i.e., $T \vdash \bot$).
- A theory is *complete* if for every sentence φ either $T \vdash \varphi$ or $T \vdash \neg\varphi$ (but not both, otherwise it would be inconsistent).
- Deduction Theorem: For a theory T and sentences φ, ψ, it holds that $T, \varphi \vdash \psi$ if and only if $T \vdash \varphi \to \psi$.

At the end of this section, we will look at several applications of the Soundness and Completeness Theorems.

1.5.1 Löwenheim-Skolem Theorem

Theorem 1.5.2 (Löwenheim-Skolem)**.** *If L is a countable language without equality, then every consistent L-theory has a countably infinite model.*

Proof. Take any finished (e.g., systematic) tableau from the theory T with the entry $\mathrm{F}\bot$ at the root. Since T is consistent, a contradiction is not provable in it, so the tableau must contain a non-contradictory branch. The desired countably infinite model is the L-reduct of the canonical model for this branch. □

We will return to this theorem in Chapter **??**, where we will show a stronger version including languages with equality (for those, the canonical model is countable but it can also be finite).

1.5.2 Compactness Theorem

Just as in propositional logic, the Compactness Theorem holds, and its proof is the same:

Theorem 1.5.3 (On Compactness)**.** *A theory has a model if and only if every finite part of it has a model.*

Proof. A model of a theory is obviously a model of each of its parts. Conversely, if T has no model, it is contradictory, so $T \vdash \bot$. Take any *finite* tableau proof of $\bot$ from T. To construct it, finitely many axioms of T suffice, and they form a finite subtheory $T' \subseteq T$, which has no model. □

1.5.3 Nonstandard Model of Natural Numbers

At the very end of this section, we will show that there exists a so-called *nonstandard model* of natural numbers. The key is the Compactness Theorem.

Let $\underline{\mathbb{N}} = \langle \mathbb{N}, S, +, \cdot, 0, \leq \rangle$ be the standard model of natural numbers. Denote by $\mathrm{Th}(\underline{\mathbb{N}})$ the set of all sentences *valid* in the structure $\underline{\mathbb{N}}$ (the so-called *theory of the structure* $\underline{\mathbb{N}}$). For $n \in \mathbb{N}$, define the *n-th numeral* as the term $\underline{n} = S(S(\cdots(S(0)\cdots))$, where S is applied n times.

Let us take a new constant symbol c and express that it is strictly greater than each n-th numeral:

$$T = \mathrm{Th}(\underline{\mathbb{N}}) \cup \{\underline{n} < c \mid n \in \mathbb{N}\}$$

Note that every finite part of the theory T has a model. Thus, it follows from the Compactness Theorem that the theory T also has a model. We call it a *nonstandard model* (denote it by $\mathcal{A}$). It satisfies the same sentences as the standard model but also contains an element $c^{\mathcal{A}}$ that is greater than every $n \in \mathbb{N}$ (by which we mean the value of the term $\underline{n}$ in the nonstandard model $\mathcal{A}$).

1.6 Hilbert Calculus in Predicate Logic

At the end of the chapter, we will show how Hilbert's calculus, introduced in Section **??**, can be adapted for use in predicate logic. This is not difficult; to deal with quantifiers, it is sufficient to add two new schemes of logical axioms and one new inference rule. We will again show the soundness of this proof system and only state without proof that it is also complete.

Proofs will consist of arbitrary formulas, not just sentences. Recall that Hilbert's calculus uses only the connectives $\neg$ and $\rightarrow$. We will have similar logical axioms as in propositional logic; in the case of a language with equality, we will additionally include the *axioms of equality*.

Definition 1.6.1 (Axiom Schemes in Hilbert Calculus in Predicate Logic)**.** For any formulas φ, ψ, χ, term t, and variable x, the following formulas are logical axioms:

(i) $\varphi \rightarrow (\psi \rightarrow \varphi)$

(ii) $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$

(iii) $(\neg\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \varphi)$

(iv) $(\forall x)\varphi \rightarrow \varphi(x/t)$, if t is substitutable for x in φ

(v) $(\forall x)(\varphi \to \psi) \to (\varphi \to (\forall x)\psi)$, if x is not free in φ

If the language is with equality, then the logical axioms also include the *axioms of equality* for the given language.

Note that all logical axioms are indeed tautologies. The inference rules are *modus ponens* and *generalization*:

Definition 1.6.2 (Modus Ponens)**.** *Modus ponens* states that if we have already proved φ and also $\varphi \to \psi$, we can derive the formula ψ:

$$\frac{\varphi, \varphi \to \psi}{\psi}$$

Definition 1.6.3 (Generalization Rule)**.** The *generalization rule* states that if we have proved φ, we can derive the formula $(\forall x)\varphi$ (for any variable x):

$$\frac{\varphi}{(\forall x)\varphi}$$

Note that both inference rules are *sound*, i.e., if $T \models \varphi$ and $T \models \varphi \to \psi$ in some theory, we also have $T \models \psi$, and similarly if $T \models \varphi$, then $T \models (\forall x)\varphi$.

Just as in propositional logic, a *proof* will be a finite sequence of formulas, in which each newly written formula is either an axiom (logical, including the axiom of equality, or from the theory we are proving in), or can be derived from previous ones using one of the inference rules:

Definition 1.6.4 (Hilbert-style Proof)**.** A *Hilbert-style proof* of the formula φ from the theory T is a *finite* sequence of formulas $\varphi_0, \ldots, \varphi_n = \varphi$, in which for each $i \leq n$:

- φ_i is a logical axiom (including axioms of equality, if the language is with equality), or
- φ_i is an axiom of the theory ($\varphi_i \in T$), or
- φ_i can be derived from some previous formulas φ_j, φ_k (where $j, k < i$) using modus ponens, or
- φ_i can be derived from some previous formula φ_j (where $j < i$) using the generalization rule.

If there is a Hilbert-style proof, φ is *(Hilbert-style) provable*, and we write $T \vdash_H \varphi$.

In predicate logic, Hilbert's calculus is again a sound and complete proof system.

Theorem 1.6.5 (On the Soundness of Hilbert Calculus)**.** *For any theory T and formula φ, the following holds:*

$$T \vdash_H \varphi \;\Rightarrow\; T \models \varphi$$

Proof. By induction on the index i, we show that each formula φ_i in the proof (and thus also $\varphi_n = \varphi$) is true in T.

If φ_i is a logical axiom (including the axioms of equality), $T \models \varphi_i$ holds because logical axioms are tautologies. If $\varphi_i \in T$, certainly also $T \models \varphi_i$. The rest follows from the soundness of the inference rules. □

For the sake of completeness, we state completeness as well, but we will omit a proof.

Theorem 1.6.6 (On the Completeness of Hilbert Calculus)**.** *For any theory T and formula φ, the following holds:*

$$T \models \varphi \;\Rightarrow\; T \vdash_H \varphi$$

Bibliography