Mathematical Logic

Hannes Leitgeb

October 2006

These lecture notes follow closely:

Ebbinghaus, H.D., Flum, J., Thomas, W., *Mathematical Logic*, New York: Springer, 1984.

Contents

1 Introduction

What is mathematical logic?

Let us consider a simple theorem in group theory:

A group is a triple $(G, \circ, e)$ with

- G is a set, $G \neq \varnothing$
- $\circ : G \times G \to G$
- $e \in G$

such that the following axioms are satisfied:

(G1)	For all $x, y, z : x \circ (y \circ z) = (x \circ y) \circ z$	(Associative Law)
(G2)	For all $x : x \circ e = x$	(Right Neutral Element)
(G3)	For all x there is a y such that $x \circ y = e$	(Right Inverse Element)

Example 1

- $(\mathbb{R}, +, 0)$ *is a group.*
- $(\mathbb{R}, \cdot, 1)$ *is not a group (there is no* y *such that* $0 \cdot y = 1$*).*

We can prove the following theorem:

Theorem 1 *Let* $(G, \circ, e)$ *be an arbitrary group:*
For every group element in G *there is a left inverse element, i.e.,*
for all x *there is a* y *such that* $y \circ x = e$.

Proof.
Let x be an arbitrary member of G:
By (G3), for this x there is a $y \in G$ such that: $x \circ y = e$.

By (G3), for this y there is a $z \in G$ such that: $y \circ z = e$.

Therefore:

$$
\begin{aligned}
y \circ x &= (y \circ x) \circ e && \text{(G2)} \\
&= \underbrace{(y \circ x)}_{(a \circ b)} \circ \underbrace{(y \circ z)}_{c} && (y \circ z = e) \\
&= \underbrace{y}_{a} \circ \underbrace{(x \circ (y \circ z))}_{(b \circ c)} && \text{(G1)} \\
&= y \circ ((x \circ y) \circ z) && \text{(G1)} \\
&= y \circ (e \circ z)) && (x \circ y = e) \\
&= (y \circ e) \circ z && \text{(G1)} \\
&= y \circ z && \text{(G2)} \\
&= e && (y \circ z = e)
\end{aligned}
$$

■

Reconsidering the above we recognize the following three "ingredients":

1. a theorem (which is nothing but a sentence in a formalised language)
2. the claim that this theorem is a logical consequence of other sentences (here: the group axioms)
3. the proof of the theorem

$\hookrightarrow$ Mathematical logic is the subdiscipline of mathematics which deals with the mathematical properties of *formal languages*, *logical consequence*, and *proofs*.

Here is another example:

An equivalence structure is a pair $(A, \approx)$ where

- A is a set, $A \neq \varnothing$
- $\approx \subseteq A \times A$

such that the following axioms are satisfied:

(A1)	For all $x : x \approx x$	(Reflexivity)
(A2)	For all $x, y :$ if $x \approx y$ then $y \approx x$	(Symmetry)
(A3)	For all $x, y, z :$ if $x \approx y$ *and* $y \approx z$ then $x \approx z$	(Transitivity)

Example 2

- *If $=$ is the equality relation on a non-empty set A, then $(A, =)$ is an equivalence structure.*
- *For $m, n \in \mathbb{Z}$ let $m \equiv n$ iff there is a $k \in \mathbb{Z}$ s.t. $m - n = 5k$. Then $(\mathbb{Z}, \equiv)$ is an equivalence structure.*

Consider the following simple theorem on equivalence structures:

Theorem 2 *Let $(A, \approx)$ be an arbitrary equivalence structure:
If two elements of A are equivalent to some common element of A, then they are equivalent to exactly the same elements of A, i.e.,
for all x, y: if there is some u such that $x \approx u, y \approx u$, then for all z:*

$$x \approx z \textit{ iff } y \approx z$$

Proof.
Let x, y be arbitrary members of A.
Assume that there is a u such that $x \approx u, y \approx u$.
It follows:

$u \approx x$	(A2, $x \approx u$)
$u \approx y$	(A2, $y \approx u$)
$y \approx x$	(A3, $y \approx u, u \approx x$)
$x \approx y$	(A2, $y \approx x$)

Now let z be an arbitrary member of A:
If $x \approx z$, then $y \approx z$ because of (A3, $y \approx x, x \approx z$).
If $y \approx z$, then $x \approx z$ because of (A3, $x \approx y, y \approx z$). ■

We recognize the same three "ingredients" as before:

1. a theorem
2. the claim that this theorem is a logical consequence of other sentences (in this case: the axioms of equivalence structures)
3. the proof of the theorem

More generally: we deal with

1. a set Φ of sentences ("axioms"), a sentence φ ("theorem")
2. the claim that φ follows logically from Φ
3. the proof of φ on the basis of Φ

In mathematical logic this is made precise:

1. sentences: members of so-called *first-order languages*
2. consequence: a first-order sentences φ follows logically from a set Φ of first-order sentences iff every *model* that satisfies all sentences in Φ also satisfies φ
3. proofs: sequences of first-order sentences which can be generated effectively on the basis of a particular set of *formal rules*

We will define 'first-order language', 'model', 'proof',... and prove theorems *about* first-order languages, models, and proofs. E.g., we will show:

- If φ is derivable from Φ on the basis of the rules in 3, then φ follows logically from Φ in the sense of 2 ("Soundness Theorem").
- If φ follows logically from Φ in the sense of 2, then φ is derivable from Φ on the basis of the rules in 3 ("Completeness Theorem").

Historical development:

Aristotle (384–322 BC)
G. Frege (AD 1848–1925)
K. Gödel (AD 1906–1978)
A. Tarski (AD 1902–1983)

G. Gentzen (AD 1909–1945)
A. Turing (AD 1912–1954)
⋮

Subdisciplines of mathematical logic:

- Model theory
- Proof theory
- Set theory
- Computability theory (Recursion theory)

Logic is at the intersection of mathematics, computer science, and philosophy.

References to (good) introductory text books:

- Ebbinghaus, H.D., Flum, J., Thomas, W., *Mathematical Logic*, New York: Springer, 1984.
- Enderton, H.B., *A Mathematical Introduction to Logic*, San Diego: Harcourt, 2001.
- Shoenfield, J.R., *Mathematical Logic*, Natick: A K Peters, 2000.

In particular, we will make use of Ebbinghaus, Flum, Thomas.

Internet: check out *http://world.logic.at/*

2 First-Order Languages

2.1 Preliminary Remarks on Formal Languages

Alphabet: non-empty set $\mathcal{A}$ the members of which are called 'symbols'.

Example 3

- $\mathcal{A}_1 = \{l_1, \ldots, l_{26}\}$, where $l_1 =$ 'a', $\ldots$, $l_{26} =$ 'z'
  (if denoted less precisely but more simply: $\mathcal{A}_1 = \{a, b, \ldots, z\}$)
- $\mathcal{A}_2 = \{n_0, n_1, \ldots, n_9\}$, where $n_0 =$ '0', $\ldots$, $n_9 =$ '9'
- $\mathcal{A}_3 = \mathbb{N}$
- $\mathcal{A}_4 = \mathbb{R}$

We are going to restrict ourselves to *countable* alphabets (i.e., finite or countably infinite ones); so e.g. $\mathcal{A}_4$ will be excluded.

We recall:

X is countable iff
$X = \{x_1, x_2, x_3, \ldots\}$ iff
there is a function $f : \mathbb{N} \to X$, such that f is surjective (onto) iff
there is a function $f : X \to \mathbb{N}$, such that f is injective (one-to-one).

(Note that for *finite* sets X it holds that for some n:
$x_{n+1} = x_n, x_{n+2} = x_n, \ldots$)

String over $\mathcal{A}$: finite linear sequence $\underbrace{\zeta_1, \zeta_2, \ldots, \zeta_n}_{(\zeta_1, \ldots, \zeta_n)}$ of members of $\mathcal{A}$

(i.e., $\zeta_i \in \mathcal{A}$ for $1 \leq i \leq n$).

Example 4

- *abudabi, goedel: strings over* $\mathcal{A}_1$
- *4711, 007: strings over* $\mathcal{A}_2$

- *123: string over* $\mathcal{A}_3$

  $\hookrightarrow$ *But* which *string?*

  $123 \approx (1,2,3)$ or $123 \approx (12,3)$ or $123 \approx (1,23)$ or $123 \approx (123)$*?*

We have to take care to choose alphabets and notations for strings in a way such that every string can be reconstructed uniquely as a sequence of members of the alphabet.

Let $\mathcal{A}^*$ be the *set of strings* over the alphabet $\mathcal{A}$.

Here is a simple lemma on the cardinality of $\mathcal{A}^*$:

Lemma 1 *If* $\mathcal{A}$ *is countable, then* $\mathcal{A}^*$ *is countable as well; indeed* $\mathcal{A}^*$ *is countably infinite.*

Proof. Recall the fundamental theorem of elementary number theory:
Every natural number $n > 1$ can be represented uniquely as a product of prime numbers $p_1, \ldots, p_k$ with $p_1 < \ldots < p_k$, such that

$$n = p_1^{i_1} \cdot \ldots \cdot p_k^{i_k}, \qquad \text{where } i_j > 0 \text{ for } 1 \leq j \leq k$$

E.g.: $14 = 2 \cdot 7 = 2^1 \cdot 7^1$, $120 = 2 \cdot 2 \cdot 2 \cdot 3 \cdot 5 = 2^3 \cdot 3^1 \cdot 5^1$

Now let q_1 be the first prime number (i.e., 2), q_2 be the second prime number (i.e., 3),..., q_n the n-th prime number,...
Since $\mathcal{A}$ is countable by assumption, either of the following is the case:

- $\mathcal{A}$ is finite: $\mathcal{A} = \{a_1, \ldots, a_n\}$ with pairwise distinct a_i
- $\mathcal{A}$ is (countably) infinite: $\mathcal{A} = \{a_1, a_2, a_3, \ldots\}$ with pairwise distinct a_i

Let

$$\begin{array}{rrcl} f: & \mathcal{A}^* & \to & \mathbb{N} \\ & a_{i_1} a_{i_2} \ldots a_{i_k} & \mapsto & q_1^{i_1} \cdot q_2^{i_2} \cdot \ldots \cdot q_k^{i_k} \end{array}$$

E.g.: $f(a_3 a_2 a_2) = 2^3 \cdot 3^2 \cdot 5^2 = 1800$
(coding functions of such a type are often referred to as ‘Gödelisations’).

Claim: f is injective (hence this “coding” is unique).

For assume $\underbrace{f(a_{i_1}a_{i_2}\dots a_{i_k})}_{q_1^{i_1}\cdot q_2^{i_2}\cdot\ldots\cdot q_k^{i_k}} = \underbrace{f(a_{j_1}a_{j_2}\dots a_{j_l})}_{q_1^{j_1}\cdot q_2^{j_2}\cdot\ldots\cdot q_l^{j_l}}$

Since the prime factor representation of natural numbers is unique, it follows:
$\Rightarrow k = l$
$\Rightarrow i_1 = j_1, \dots, i_k = j_l$
$\Rightarrow a_{i_1} \dots a_{i_k} = a_{j_1} \dots a_{j_l}$ ✓
$\Longrightarrow$ according to what we said before about countability: $\mathcal{A}^*$ is *countable*.

Indeed, $\mathcal{A}^*$ is countably *infinite*: $a_1, a_1a_1, a_1a_1a_1, \ldots \in \mathcal{A}^*$. ■

2.2 Vocabulary of First-Order Languages

Now we consider the alphabets of formal languages of a particular kind:
the vocabularies of *first-order languages*.

Take our axioms (G3) and (A3) from section 1 as examples:

- (G3) For all x there is a y such that $x \circ y = e$
- (A3) For all x, y, z: if $x \approx y$ *and* $y \approx z$ then $x \approx z$

In these sentences and in other ones about groups and equivalence structures
we find symbols of the following types:

Propositional connectives:
"and": $\wedge$
"if-then": $\rightarrow$
"not": $\neg$
"or": $\vee$
"iff" ("if and only if"): $\leftrightarrow$

Quantifiers:
"for all": $\forall$
"there is': $\exists$

Variables:
"x","y","z": $v_0, v_1, v_2, \ldots$

Equality sign:
"$=$": $\equiv$ (binary predicate)

Predicates:
"$\approx$": P_0^2 (binary predicate)

Function signs:
"$\circ$": f_0^2 (binary function sign)

Constants:
"e": c_0

Parentheses are used as auxiliary symbols.

More generally:

Definition 1 *The* vocabulary *or* alphabet of a first-order language *contains the following symbols (and only the following symbols):*

1. $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
2. $\forall, \exists$
3. $v_0, v_1, v_2, \ldots$
4. $\equiv$
5. *for every $n \geq 1$ a (possibly empty) set of n-ary predicates P_i^n* $(i = 0, 1, 2 \ldots)$
6. *for every $n \geq 1$ a (possibly empty) set of n-ary function signs f_i^n* $(i = 0, 1, 2 \ldots)$
7. *a (possibly empty) set of constants c_i* $(i = 0, 1, 2, \ldots)$

and parentheses as auxiliary symbols.

Note:

- $\mathcal{A} = \{(1), (2), (3), (4)\}$ is fixed: these symbols are contained in *every* first-order language.
- $\mathcal{S} = \{(5), (6), (7)\}$ is optional: the choice of $\mathcal{S}$ determines the specific character of a first-order language.
- $\mathcal{A}_{\mathcal{S}} = \mathcal{A} \cup \mathcal{S}$ is the actual alphabet of the first-order language that is determined by $\mathcal{S}$.

Example 5

- $\mathcal{S}_{\mathrm{Gr}} = \{\underbrace{\circ}_{f_0^2}, \underbrace{e}_{c_0}\}$ *determines the first-order language of group theory.*
- $\mathcal{S}_{\mathrm{Equ}} = \{\underbrace{\approx}_{P_0^2}\}$ *determines the first-order language of equivalence structures.*

In future we will use the following conventions for "metavariables":
"P","Q","R" (with or without indices) denote predicates.
"f","g","h" (with or without indices) denote function signs.
"c" (with or without indices) denote constants.
"x","y","z" (with or without indices) denote variables.

Remark 1 *Let $\mathcal{S}$ be the specific symbol set of a first-order language (such that $\mathcal{A}_{\mathcal{S}} = \mathcal{A} \cup \mathcal{S}$ is the alphabet of that language):*

$\mathcal{A}_{\mathcal{S}}$ is countable (in fact countably infinite because of $v_0, v_1, \ldots \in \mathcal{A}$)
By lemma 1, $\mathcal{A}_{\mathcal{S}}^*$ is also countably infinite.

2.3 Terms and Formulas of First-Order Languages

Compare natural languages:

'This is lame and extremely boring' is well-formed ("grammatical").
'and extremely this lame is boring' is *not* well-formed.

Accordingly, we will now characterise particular strings over some alphabet $\mathcal{A}_\mathcal{S}$ of a first-order language as being well-formed. E.g.:

$e \wedge) \rightarrow \forall$: *not* well-formed, although it is a member of $\mathcal{A}^*_{\mathcal{S}_{\mathrm{Gr}}}$.
$e \circ v_1 \equiv v_2$: *well-formed* and a member of $\mathcal{A}^*_{\mathcal{S}_{\mathrm{Gr}}}$.

We are going to build up well-formed expressions in a step-by-step manner; e.g.:

$$\underbrace{\underbrace{e \circ v_1}_{\text{term}} \equiv \underbrace{v_2}_{\text{term}}}_{\text{formula}} \in \mathcal{A}^*_{\mathcal{S}_{\mathrm{Gr}}}$$

Definition 2 *Let $\mathcal{S}$ be the specific symbol set of a first-order language. $\mathcal{S}$-terms are precisely those strings over $\mathcal{A}_\mathcal{S}$ that can be generated according to the following rules:*

(T1) Every variable is an $\mathcal{S}$-term.

(T2) Every constant in $\mathcal{S}$ is an $\mathcal{S}$-term.

(T3) If $t_1, \ldots, t_n$ are $\mathcal{S}$-terms and f is an n-ary function sign in $\mathcal{S}$, then $f(t_1, \ldots, t_n)$ is an $\mathcal{S}$-term.

Abbreviated:

(T1) $\dfrac{}{x}$

(T2) $\dfrac{}{c}$

(T3) $\dfrac{t_1,\ldots,t_n}{f(t_1,\ldots,t_n)}$ $\begin{matrix}\}\text{premises}\\ \}\text{conclusion}\end{matrix}$

This means:

$t \in \mathcal{A}_{\mathcal{S}}^*$ is an $\mathcal{S}$-Term iff there is a sequence $u_1, \ldots, u_k$ of elements of $\mathcal{A}_{\mathcal{S}}^*$, such that

$$u_k = t$$

and for all u_i with $1 \leq i \leq k$ *it is the case that:*

- *u_i is a variable or*
- *u_i is a constant in $\mathcal{S}$ or*
- $u_i = f\,(t_1, \ldots, t_n)$ and $t_1, \ldots, t_n \in \{u_1, \ldots, u_{i-1}\}$.

We denote the set of $\mathcal{S}$-terms by: $\mathcal{T}_{\mathcal{S}}$.

Example 6 *Let $\mathcal{S} = \{f, g, c\}$ where f, g are binary function signs, c is a constant.*
It follows that $g\big(f(c, v_0), c\big)$ is an $\mathcal{S}$-term:

(1)	c	*(T2)*
(2)	v_0	*(T1)*
(3)	$f(c, v_0)$	*(T3, with 1., 2.)*
(4)	$g\big(f(c, v_0), c\big)$	*(T3, with 3., 1.)*

Remark 2

- *In order to denote arbitrary terms we use: "t" (with or without an index).*
- *Traditionally, one writes functions signs* between *terms: e.g. $t_1 + t_2 := +(t_1, t_2)$, $t_1 \circ t_2 := \circ(t_1, t_2)$, and so forth.*

Definition 3 *Let $\mathcal{S}$ be the specific symbol set of a first-order language. $\mathcal{S}$-formulas are precisely those strings over $\mathcal{A}_{\mathcal{S}}$ that can be generated according to the following rules:*

(F1) $\dfrac{}{\equiv(t_1,t_2)}$ *(for $\mathcal{S}$-terms t_1, t_2)*

(F2) $\dfrac{}{P(t_1,\ldots,t_n)}$ *(for $\mathcal{S}$-terms $t_1, \ldots, t_n$, for n-ary $P \in \mathcal{S}$)*

(Formulas which can be generated solely on basis of (F1) and (F2) are called atomic.*)*

(F3) $\underbrace{\dfrac{\varphi}{\neg\varphi}}_{\text{negation}}$

(F4) $\underbrace{\dfrac{\varphi,\psi}{(\varphi \vee \psi)}}_{\text{disjunction}}$ $\underbrace{\dfrac{\varphi,\psi}{(\varphi \wedge \psi)}}_{\text{conjunction}}$ $\underbrace{\dfrac{\varphi,\psi}{(\varphi \rightarrow \psi)}}_{\text{implication}}$ $\underbrace{\dfrac{\varphi,\psi}{(\varphi \leftrightarrow \psi)}}_{\text{equivalence}}$

(F5) $\underbrace{\dfrac{\varphi}{\forall x\, \varphi}}_{\text{universally quantified}}$ $\underbrace{\dfrac{\varphi}{\exists x\, \varphi}}_{\text{existentially quantified}}$ *(for arbitrary variables x)*

This means: … (analogous to the case of terms).

We denote the set of $\mathcal{S}$-formulas by: $\mathcal{F}_{\mathcal{S}}$.

Example 7 *Let $\mathcal{S} = \{R\}$ where R is a binary predicate.*

It follows that $\Big(\big(R(v_0,v_1) \wedge R(v_1,v_2)\big) \rightarrow R(v_0,v_2)\Big)$ is an $\mathcal{S}$-term:

(1)	$R(v_0,v_1)$	*(F2)*
(2)	$R(v_1,v_2)$	*(F2)*
(3)	$\big(R(v_0,v_1) \wedge R(v_1,v_2)\big)$	*(F4, with 1., 2.)*
(4)	$R(v_0,v_2)$	*(F2)*
(5)	$\Big(\big(R(v_0,v_1) \wedge R(v_1,v_2)\big) \rightarrow R(v_0,v_2)\Big)$	*(F4, with 3., 4.)*

Remark 3

- *In order to denote arbitrary formulas we use: "φ", "ψ", "ρ", … (with or without an index).*

- *Often one writes predicates* between *terms:*
  e.g. $t_1 \equiv t_2 := \ \equiv (t_1, t_2)$, $t_1 \approx t_2 := \ \approx (t_1, t_2)$, *and so forth.*

- *"(" , ")" are needed in (F4) to guarantee unique readability:*
  otherwise $\varphi \wedge \psi \vee \rho$ *could either be* $(\varphi \wedge \psi) \vee \rho$ *or be* $\varphi \wedge (\psi \vee \rho)$ *!?*

Lemma 2 *For all symbol sets* $\mathcal{S}$ *that specify first-order languages:*
$\mathcal{T}_\mathcal{S}, \mathcal{F}_\mathcal{S}$ *are countably infinite.*

Proof. We have already seen that $\mathcal{A}^*_\mathcal{S}$ is countable.
Since $\mathcal{T}_\mathcal{S}, \mathcal{F}_\mathcal{S} \subseteq \mathcal{A}^*_\mathcal{S}$, it follows that $\mathcal{T}_\mathcal{S}, \mathcal{F}_\mathcal{S}$ are countable, too.
$v_0, v_1, v_2, \ldots \in \mathcal{T}_\mathcal{S}$, thus $\mathcal{T}_\mathcal{S}$ is countably infinite.
$v_0 \equiv v_0, v_1 \equiv v_1, v_2 \equiv v_2, \ldots \in \mathcal{F}_\mathcal{S}$, therefore $\mathcal{F}_\mathcal{S}$ is countably infinite. ■

Remark 4 *Our definition of formulas is quite "liberal": if* $\mathcal{S}$ *is chosen accordingly, then*

- $\forall x \, (P(y) \to Q(y)) \in \mathcal{F}_\mathcal{S}$, *although* x *does not occur in* $(P(y) \to Q(y))$

- $\forall x \, (P(y) \to Q(y)) \in \mathcal{F}_\mathcal{S}$, *although* y *occurs freely in* $(P(y) \to Q(y))$, *i.e., it occurs not in the range of ("bound by")* $\forall y$ *or* $\exists y$

- $\exists x (x \equiv x \wedge x \equiv x) \in \mathcal{F}_\mathcal{S}$, *although this is obviously redundant*

- $\exists x \exists x P(x) \in \mathcal{F}_\mathcal{S}$, *although the* x *in* $\exists x P(x)$ *is bound and therefore the first occurrence of* $\exists x$ *is useless.*

But being tolerant in these ways does not do any harm and makes things much simpler. So let's not care!

At some points we will need to sort out those variables which occur freely in a formula, i.e., which occur not bound by any quantifier:

Definition 4 *Let* $\varphi \in \mathcal{F}_\mathcal{S}$ *(for arbitrary symbol set* $\mathcal{S}$*):*
$free(\varphi)$, *i.e., the set of variables which occur freely in* φ, *can be defined recursively as follows:*

- $free(t_1 \equiv t_2) := var(t_1) \cup var(t_2)$

  (let $var(t)$ *be the set of variables in* t*)*

- $free\,(P(t_1,\ldots,t_n) := var(t_1) \cup \ldots \cup var(t_n)$
- $free(\neg\varphi) := free(\varphi)$
- $free\,((\varphi \wedge \psi)) := free(\varphi) \cup free(\psi)$
  (analogously for $\vee, \rightarrow, \leftrightarrow$*)*
- $free(\forall x\, \varphi) := free(\varphi) \backslash \{x\}$
  (analogously for $\exists$*).*

Example 8

$free\,(\forall x\,(P(y) \rightarrow Q(y))) = \{y\}$

$free\,(\exists x P(x)) = \varnothing$

$free\,((\exists x P(x) \wedge P(x))) = \{x\}$

We see that according to our definition, $free(\varphi)$ is the set of variables that occur freely at *some* place within φ.

A formula φ without free variables, i.e., for which $free(\varphi) = \varnothing$, is called a *sentence*.

Remark 5 *For simplicity, we will sometimes omit parentheses:*

- *Outer parentheses: e.g.,* $\varphi \wedge \psi := (\varphi \wedge \psi)$
- $\neg$ *binds more strongly than* $\wedge, \vee$*, i.e.:*
  $\neg\varphi \wedge \psi := (\neg\varphi \wedge \psi) \qquad [\neq \neg(\varphi \wedge \psi)]$
- $\wedge, \vee$ *bind more strongly than* $\rightarrow, \leftrightarrow$*, i.e.:*
  $\varphi \wedge \psi \rightarrow \varphi \vee \psi := (\varphi \wedge \psi) \rightarrow (\varphi \vee \psi)$

  $[\neq (\varphi \wedge (\psi \rightarrow \varphi)) \vee \psi]$
- *in the case of multiple conjunctions or disjunctions:*
  $\varphi \wedge \psi \wedge \rho = (\varphi \wedge \psi) \wedge \rho$
  $\varphi \vee \psi \vee \rho = (\varphi \vee \psi) \vee \rho$

2.4 Induction over Terms and Formulas

Throughout this course we will use the following method of proof over and over again:

1. We show: All terms/formulas ζ with $\frac{}{\zeta}$ have the property P.
2. We show: Assume terms/formulas $\zeta_1, \ldots, \zeta_n$ have the property P. Then all terms/formulas ζ with $\frac{\zeta_1,\ldots,\zeta_n}{\zeta}$ have the property P.

1. and 2. imply: *All* terms/formulas ζ have the property P.

Remark 6 *This is simply a version of* proof by complete induction over natural numbers*. One actually shows by complete induction over n:*

For all $n \in \mathbb{N}$: For all ζ: if ζ is derivable in the term-/formula-calculus in n steps, then ζ has the property P.

Example 9
Claim: All terms in $\mathcal{T}_\mathcal{S}$ contain as many opening parentheses as they contain closing parantheses; i.e.: all terms ζ have the property of

$$\underbrace{\textit{containing as many opening parentheses as closing parentheses}}_{P}$$

Proof.
Induction basis:
Variables have the property P (since they do not contain parentheses at all).✓
Constants have the property P (they do not contain parentheses at all).✓

Inductive assumption:
Assume terms $t_1, \ldots, t_n$ have the property P.
So t_1 contains as many opening parentheses as closing parentheses, t_2 contains as many opening parentheses as closing parentheses,... But then also

$$f(t_1, \ldots, t_n)$$

does so (where f is an arbitrary function sign in $\mathcal{S}$).

In short: $\underbrace{f(\underbrace{t_1, \ldots, t_n}_{IA})}_{\text{also has property P}}$ ✓ $\Longrightarrow$ by induction, we are done. ■

(Analogously for formulas!)

Now let us consider a slightly more complex application of induction over terms:

We call $X \subseteq \mathcal{A}_{\mathcal{S}}^*$ *closed under the rules of the term calculus* (for a given symbol set $\mathcal{S}$) iff

1. all variables are contained in X,
2. all constants in $\mathcal{S}$ are contained in X,
3. if $t_1, \ldots, t_n$ are in X, then $f(t_1, \ldots, t_n)$ is in X
   (where f is an arbitrary function sign in $\mathcal{S}$).

Example 10

- *$\mathcal{A}_{\mathcal{S}}^*$ is closed under the rules of the term calculus.*
- *$\mathcal{T}_{\mathcal{S}}$ is closed under the rules of the term calculus.*

Lemma 3 *$\mathcal{T}_{\mathcal{S}}$ is the intersection of all subsets of $\mathcal{A}_{\mathcal{S}}^*$ that are closed under the rules of the term calculus, i.e.,*

$$\mathcal{T}_{\mathcal{S}} = \bigcap_{\substack{X \subseteq \mathcal{A}_{\mathcal{S}}^*, \\ X\,closed}} X$$

(and thus $\mathcal{T}_{\mathcal{S}}$ is the least subset of $\mathcal{A}_{\mathcal{S}}^*$ that is closed under the rules of the term calculus).

Proof.

1. If X is closed under the rules of the term calculus, then $\mathcal{T}_{\mathcal{S}} \subseteq X$:

Since: all $t \in \mathcal{T}_\mathcal{S}$ $\underbrace{\text{are contained in } X}_{\text{property P}}$, by induction:

(a) variables and constants are contained in X ✓
(b) assume $t_1, \ldots, t_n \in \mathcal{T}_\mathcal{S}$ are contained in X:
$\Longrightarrow$ also $f(t_1, \ldots, t_n)$ is a member of X ✓
(both by assumption that X is closed under the term rules).

2. Therefore: $\mathcal{T}_\mathcal{S} \subseteq \bigcap_{\text{X closed}} X$

3. $\mathcal{T}_\mathcal{S}$ is itself closed under the rules of the term calculus.

Since: (a) variables and constants are in $\mathcal{T}_\mathcal{S}$ ✓
(b) assume $t_1, \ldots, t_n$ are in $\mathcal{T}_\mathcal{S}$
[let us do this very precisely for once...]
$\Longrightarrow$ there are term derivations of the form

$$
\begin{aligned}
&u_1^1, \ldots, u_{k_1}^1 \text{ with } u_{k_1}^1 = t_1\\
&u_1^2, \ldots, u_{k_2}^2 \text{ with } u_{k_2}^2 = t_2\\
&\quad\vdots\\
&u_1^n, \ldots, u_{k_n}^n \text{ with } u_{k_n}^n = t_n
\end{aligned}
$$

But then

$$u_1^1, \ldots, \underbrace{u_{k_1}^1}_{t_1}, u_1^2, \ldots, \underbrace{u_{k_2}^2}_{t_2}, \ldots, u_1^n, \ldots, \underbrace{u_{k_n}^n}_{t_n}, f(t_1, \ldots, t_n)$$

is a derivation of $f(t_1, \ldots, t_n)$ in the term calculus
(where the last derivation step is an application of (T3))
$\Longrightarrow$ $f(t_1, \ldots, t_n)$ is in $\mathcal{T}_\mathcal{S}$✓

4. $\bigcap_{\text{X closed}} X \subseteq \mathcal{T}_\mathcal{S}$ since 3 implies that $\mathcal{T}_\mathcal{S}$ is itself one of the closed sets X.

From 2 and 4 follows: $\mathcal{T}_\mathcal{S} = \bigcap_{\text{X closed}} X$

■

An analogous statement can be proved for $\mathcal{F}_\mathcal{S}$ and the formula calculus.

2.5 Problem Set 1

1. (a) Show (this is a recapitulation of something you should know about countable sets):

   If the sets $M_0, M_1, M_2, \ldots$ are countable,
   then $\bigcup_{n\in\mathbb{N}} M_n$ is countable as well.

   (b) Prove the following lemma (lemma 1 from above) by means of 1a:

   If $\mathcal{A}$ is a countable alphabet, then the set $\mathcal{A}^*$ (of finite strings over $\mathcal{A}$) is countable, too.

2. Let $\mathcal{S}$ be an arbitrary symbol set. We consider the following calculus $\mathcal{C}$ of rules:

   - $\dfrac{}{x \;\; x}$

   (for arbitrary variables x)

   - $\dfrac{x \;\; t_i}{x \;\; f(t_1, \ldots, t_n)}$

   (for arbitrary variables x, for arbitrary $\mathcal{S}$-terms $t_1, \ldots, t_n$, for arbitrary n-ary function signs $f \in \mathcal{S}$, for arbitrary $i \in \{1, \ldots, n\}$).

   Show that for all variables x and all $\mathcal{S}$-terms t holds: The string

   $$x \;\; t$$

   is derivable in $\mathcal{C}$ if and only if $x \in var(t)$ (i.e., x is a variable in t).

3. Prove that the following strings are $\mathcal{S}$-terms (for given $\mathcal{S}$ with $c, f, g \in \mathcal{S}$, where f is a binary function sign, g is a unary function sign, x and y are variables):

   (a) $f(x, c)$

   (b) $g(f(x, c))$

   (c) $f(f(x, c), f(x, f(x, y)))$

4. Prove that the following strings are $\mathcal{S}$-formulas (with x, y, c, f, g as in 3 and where P, $Q \in \mathcal{S}$, such that P is a unary predicate and Q is a binary predicate):

   (a) $\neg P(f(x,c))$

   (b) $\exists x \forall y (P(g(f(x,c))) \to Q(y,y))$

   (c) $(\forall x \neg P(f(x,c)) \vee Q(f(x,c), f(f(x,c), f(x, f(x,y)))))$

5. Prove by induction: the string $\forall x f(x,c)$ is not an $\mathcal{S}$-term (where $\mathcal{S}$ is an arbitrary symbol set).

6. Let x, y, z be variables, $f \in \mathcal{S}$ a unary function sign, P, Q, $R \in \mathcal{S}$ with P being a binary predicate, Q a unary predicate, and R a ternary predicate. Determine for the following $\mathcal{S}$-formulas φ the corresponding set of variables that occur freely in φ (i.e., the sets $free(\varphi)$):

   (a) $\forall x \exists y (P(x,z) \to \neg Q(y)) \to \neg Q(y)$

   (b) $\forall x \forall y (Q(c) \wedge Q(f(x))) \to \forall y \forall x (Q(y) \wedge R(x,x,y))$

   (c) $Q(z) \leftrightarrow \exists z (P(x,y) \wedge R(c,x,y))$

   Which of these formulas are *sentences* (have no free variables at all)?

   [Note that we omitted parentheses in 6 as explained in remark 5.]

3 Semantics of First-Order Languages

3.1 Models, Variable Assignments, and Semantic Values

Example 11 *Consider* $\mathcal{S}_{Gr} = \{\circ, e\}$*:*
The formula $\forall x \exists y\, x \circ y \equiv e$ *is a member of* $\mathcal{F}_{\mathcal{S}_{Gr}}$*.*

But what does this formula mean*?*

- *If* $\forall x\, \exists y$ *quantify over the set* $\mathbb{Z}$ *of integers, if* $\circ$ *stands for addition in* $\mathbb{Z}$*, and if* e *denotes the integer* 0*, then the formula means:*

  "for every integer there is another integer such that the addition of both is equal to 0*"*

  $\hookrightarrow$ *if interpreted in this way: the formula is true!*

- *If* $\forall x\, \exists y$ *quantify over* $\mathbb{Z}$*, if* $\circ$ *stands for multiplication in* $\mathbb{Z}$*, and if* e *denotes the integer* 1*, then the formula means:*

  "for every integer there is another integer such that the product of both is equal to 1*"*

  $\hookrightarrow$ *if interpreted in this way: the formula is false!*

Let us now make precise what we understand by such *interpretations* of symbol sets of first-order languages:

Definition 5 *Let* $\mathcal{S}$ *be an arbitrary symbol set:*

An $\mathcal{S}$*-model (or* $\mathcal{S}$*-structure) is an ordered pair* $\mathfrak{M} = (D, \mathfrak{I})$*, such that:*

1. D is a set, $D \neq \varnothing$ *("domain", "universe of discourse", "range of the quantifiers")*

2. $\mathfrak{I}$ *is defined on* $\mathcal{S}$ *as follows ("interpretation of* $\mathcal{S}$*"):*

   - *for n-ary predicates* P *in* $\mathcal{S}$*:* $\mathfrak{I}(P) \subseteq \underbrace{D \times \ldots \times D}_{n \text{ times}} = D^n$

   - *for n-ary function signs* f *in* $\mathcal{S}$*:* $\mathfrak{I}(f) : D^n \to D$
     (i.e., $\mathfrak{I}(f)$ *is a mapping from* D^n *to* D*)*

- *for every constant c in $\mathcal{S}$: $\mathfrak{I}(c) \in D$*

Example 12

1. $\mathcal{S} = \mathcal{S}_{Gr} = \{\circ, e\}$,
   $D = \mathbb{Z}$,
   $\mathfrak{I}(\circ) :=$ + on $\mathbb{Z}$, i.e., $\mathfrak{I}(\circ) : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ *with* $(m, n) \mapsto m + n$,
   $\mathfrak{I}(e) := 0$,
   $\Longrightarrow \mathfrak{M} = (D, \mathfrak{I})$ *is the model of the set of integers viewed as additive group.*

2. $\mathcal{S} = \mathcal{S}_{\text{Fie}} = \{\bar{+}, \bar{\cdot}, \bar{0}, \bar{1}\}$,
   $D = \mathbb{R}$,
   $\mathfrak{I}(\bar{+}) :=$ + on $\mathbb{R}$,
   $\mathfrak{I}(\bar{\cdot}) :=$ · on $\mathbb{R}$,
   $\mathfrak{I}(\bar{0}) := 0$,
   $\mathfrak{I}(\bar{1}) := 1$,
   $\Longrightarrow \mathfrak{M} = (D, \mathfrak{I})$ *is the model of the real number field.*

3. *Now we add* $\bar{<}$ *to* $\mathcal{S}_{\text{Fie}}$*:* $\mathcal{S}_{\text{OrdFie}} = \{\bar{+}, \bar{\cdot}, \bar{0}, \bar{1}, \bar{<}\}$ *with*
   $D = \mathbb{R}, \ldots$ *(as in 2)*
   Moreover: $\mathfrak{I}(\bar{<}) \subseteq \mathbb{R} \times \mathbb{R}$ *such that* $(d_1, d_2) \in \mathfrak{I}(\bar{<}) \Leftrightarrow d_1 < d_2$
   (i.e., $\mathfrak{I}(\bar{<})$ *is the less-than relation on real numbers).*
   $\Longrightarrow \mathfrak{M} = (D, \mathfrak{I})$ *is the model of the ordered real number field.*

Just as an interpretation assigns meaning to predicates, function signs, and constants, we need a way of assigning values to variables:

Definition 6 *A variable assignment over a model* $\mathfrak{M} = (D, \mathfrak{I})$ *is a function* $s : \{v_0, v_1, \ldots\} \to D$.

Remark 7 *As we will see, we need variable assignments in order to define the truth values of quantified formulas. Here is the idea:*
$\forall x \varphi$ *is true* $\Leftrightarrow$ *whatever* $d \in D$ *a variable assignment assigns to* x, φ *turns out to be true under this assignment.*

Analogously for $\exists x \varphi$ and the existence of an element $d \in D$.

It is also useful to have a formal way of changing *variable assignments:*
Let s be a variable assignment over $\mathfrak{M} = (D, \mathfrak{I})$, let $d \in D$:

We define

$$s\frac{d}{x} : \ \{v_0, v_1, \ldots\} \to D$$

$$s\frac{d}{x}(y) := \begin{cases} d, & y = x \\ s(y), & y \neq x \end{cases}$$

(where x is some variable in $\{v_0, v_1, \ldots\}$).

E.g., $s\frac{4}{v_0}(v_0) = 4, s\frac{4}{v_0}(v_1) = s(v_1)$.

Given an $\mathcal{S}$-model together with a variable assignment over this model, we can define the *semantic value* of a term/formula:

Definition 7 *Let $\mathfrak{M} = (D, \mathfrak{I})$ be an $\mathcal{S}$-model.*
Let s be a variable assignment over $\mathfrak{M}$:
$Val_{\mathfrak{M},s}$ ("semantic value function") is defined on $\mathcal{T}_{\mathcal{S}} \cup \mathcal{F}_{\mathcal{S}}$, such that:

(V1) $Val_{\mathfrak{M},s}(x) := s(x)$

(V2) $Val_{\mathfrak{M},s}(c) := \mathfrak{I}(c)$

(V3) $Val_{\mathfrak{M},s}(f(t_1, \ldots, t_n)) := \mathfrak{I}(f)\,(Val_{\mathfrak{M},s}(t_1), \ldots, Val_{\mathfrak{M},s}(t_n))$

(V4) $Val_{\mathfrak{M},s}(t_1 \equiv t_2) := 1 \Longleftrightarrow Val_{\mathfrak{M},s}(t_1) = Val_{\mathfrak{M},s}(t_2)$

(V5) $Val_{\mathfrak{M},s}\,(P(t_1, \ldots, t_n)) := 1 \Longleftrightarrow (Val_{\mathfrak{M},s}(t_1), \ldots, Val_{\mathfrak{M},s}(t_n)) \in \mathfrak{I}(P)$

(V6) $Val_{\mathfrak{M},s}(\neg \varphi) := 1 \Longleftrightarrow Val_{\mathfrak{M},s}(\varphi) = 0$

(V7) $Val_{\mathfrak{M},s}(\varphi \wedge \psi) := 1 \Longleftrightarrow Val_{\mathfrak{M},s}(\varphi) = Val_{\mathfrak{M},s}(\psi) = 1$

(V8) $Val_{\mathfrak{M},s}(\varphi \vee \psi) := 1 \Longleftrightarrow Val_{\mathfrak{M},s}(\varphi) = 1$ or $Val_{\mathfrak{M},s}(\psi) = 1$ *(or both)*

(V9) $Val_{\mathfrak{M},s}(\varphi \to \psi) := 1 \Longleftrightarrow Val_{\mathfrak{M},s}(\varphi) = 0$ or $Val_{\mathfrak{M},s}(\psi) = 1$ *(or both)*

(V10) $Val_{\mathfrak{M},s}(\varphi \leftrightarrow \psi) := 1 \Longleftrightarrow Val_{\mathfrak{M},s}(\varphi) = Val_{\mathfrak{M},s}(\psi)$

(V11) $Val_{\mathfrak{M},s}(\forall x\, \varphi) := 1 \Longleftrightarrow$ for all $d \in D$: $Val_{\mathfrak{M},s\frac{d}{x}}(\varphi) = 1$

(V12) $Val_{\mathfrak{M},s}(\exists x\, \varphi) := 1 \Longleftrightarrow$ there is a $d \in D$, such that: $Val_{\mathfrak{M},s\frac{d}{x}}(\varphi) = 1$

For (V4)–(V12): in case the "iff" condition is not satisfied, the corresponding semantic value is defined to be 0*.*

Terminology:

$Val_{\mathfrak{M},s}(t)$ and $Val_{\mathfrak{M},s}(\varphi)$ are the semantic values of t and φ respectively (relative to $\mathfrak{M}, s$), where

- $Val_{\mathfrak{M},s}(t) \in D$,
- $Val_{\mathfrak{M},s}(\varphi) \in \{1, 0\} \widehat{=} \{\text{T,F}\}$.

Instead of writing that $Val_{\mathfrak{M},s}(\varphi) = 1$, we may also say:

- φ is true at $\mathfrak{M}, s$
- $\mathfrak{M}, s$ make φ true
- $\mathfrak{M}, s$ satisfy φ
- briefly: $\mathfrak{M}, s \vDash \varphi$ ("$\vDash$" is called the "semantic turnstile")

We will also write for sets Φ of formulas:
$\mathfrak{M}, s \vDash \Phi \;\Leftrightarrow\;$ for all $\varphi \in \Phi$: $\mathfrak{M}, s \vDash \varphi$

Example 13 *Let* $\mathfrak{M}$ *be the model of the ordered real number field (i.e.,* $\mathcal{S} = \mathcal{S}_{\text{OrdFie}} = \{\overline{+}, \overline{\cdot}, \overline{0}, \overline{1}, \overline{<}\}$*,* $D = \mathbb{R}$*,* $\mathfrak{I}$ *is as described on p.25,* $\mathfrak{M} = (D, \mathfrak{I})$*).*
Let s *be a variable assignment over* $\mathfrak{M}$*, such that* $s(v_1) = 3$*:*

$Val_{\mathfrak{M},s}\left(\exists v_0\; v_1 \overline{<} v_0 \overline{+} \overline{1}\right) = 1$
$\Longleftrightarrow$ there is a $d \in D = \mathbb{R}$, such that:
$Val_{\mathfrak{M},s\frac{d}{v_0}}\left(v_1 \overline{<} v_0 \overline{+} \overline{1}\right) = 1$ (V12)
$\Longleftrightarrow$ there is a $d \in D = \mathbb{R}$, such that:
$\left(Val_{\mathfrak{M},s\frac{d}{v_0}}(v_1), Val_{\mathfrak{M},s\frac{d}{v_0}}(v_0 \overline{+} \overline{1})\right) \in \mathfrak{I}(\overline{<})$ (V5)

$\iff$ there is a $d \in D = \mathbb{R}$, such that:
$\left(s\frac{d}{v_0}(v_1), \mathfrak{I}(\overline{+})\left(Val_{\mathfrak{M},s\frac{d}{v_0}}(v_0), Val_{\mathfrak{M},s\frac{d}{v_0}}(\overline{1})\right)\right) \in \mathfrak{I}(\overline{<})$ (V1),(V3)
$\iff$ there is a $d \in D = \mathbb{R}$, such that:
$(s\frac{d}{v_0}(v_1), \mathfrak{I}(\overline{+})(s\frac{d}{v_0}(v_0), \mathfrak{I}(\overline{1}))) \in \mathfrak{I}(\overline{<})$ (V1),(V2)
$\iff$ there is a $d \in D = \mathbb{R}$, such that:
$(s(v_1), \mathfrak{I}(\overline{+})\,(d, 1)) \in \mathfrak{I}(\overline{<})$ *(Def.* $s\frac{d}{v_0}, \mathfrak{I}$*)*
$\iff$ there is a $d \in D = \mathbb{R}$, such that:
$(3, d+1) \in \mathfrak{I}(\overline{<})$ *(Def.* $s, \mathfrak{I}$*)*
$\iff$ there is a $d \in D = \mathbb{R}$, such that:
$3 < d + 1$ *(Def.* $\mathfrak{I}$*)*

Such a $d \in \mathbb{R} = D$ *exists: e.g.,* $d = 3, 4, ...$

$\Longrightarrow Val_{\mathfrak{M},s}\left(\exists v_0\ v_1 \ \overline{<}\ v_0 \ \overline{+}\ \overline{1}\right) = 1$ (equivalently: $\mathfrak{M}, s \vDash \exists v_0\ v_1 \ \overline{<}\ v_0 \ \overline{+}\ \overline{1}$)

Examples like these tell us:

Remark 8

1. *The semantic value of a term* t *only depends (i) on the interpretation of the constants and functions signs that occur in* t *and (ii) on the values the assignment function assigns to the variables that occur in* t*.*

2. *The semantic value of a formula* φ *only depends (i) on the interpretation of the constants, functions signs, and predicates that occur in* φ *and (ii) on the values the assignment function assigns to the variables that occur in* φ *freely (the assignment of values to bound occurrences of variables are "erased" by the quantifiers which bind these occurrences).*

In formal terms:

Lemma 4 *(Coincidence Lemma)*
Let $\mathcal{S}_1, \mathcal{S}_2$ be two symbol sets. Let $\mathfrak{M}_1 = (D, \mathfrak{I}_1)$ be an $\mathcal{S}_1$-model, let $\mathfrak{M}_2 = (D, \mathfrak{I}_2)$ be an $\mathcal{S}_2$-model.
Let s_1 be a variable assignment over $\mathfrak{M}_1$, s_2 a variable assignment over $\mathfrak{M}_2$.
Finally, let $\mathcal{S} = \mathcal{S}_1 \cap \mathcal{S}_2$:

1. *For all terms $t \in \mathcal{T}_{\mathcal{S}}$:*

   If $\mathfrak{I}_1(c) = \mathfrak{I}_2(c)$ for all c *in* t
   $\mathfrak{I}_1(f) = \mathfrak{I}_2(f)$ for all f *in* t
   $s_1(x) = s_2(x)$ for all x *in* t

   then: $Val_{\mathfrak{M}_1, s_1}(t) = Val_{\mathfrak{M}_2, s_2}(t)$ *(compare 1 in the remark above).*

2. *For all formulas $\varphi \in \mathcal{F}_{\mathcal{S}}$:*

   If $\mathfrak{I}_1(c) = \mathfrak{I}_2(c)$ for all c *in* φ
   $\mathfrak{I}_1(f) = \mathfrak{I}_2(f)$ for all f *in* φ
   $\mathfrak{I}_1(P) = \mathfrak{I}_2(P)$ for all P *in* φ
   $s_1(x) = s_2(x)$ for all $x \in free(\varphi)$

   then: $Val_{\mathfrak{M}_1, s_1}(\varphi) = Val_{\mathfrak{M}_2, s_2}(\varphi)$ *(compare 2 in the remark above).*

Proof. (By standard induction over terms and formulas: see the next problem sheet!) ■

Corollary 1 *Let φ be an $\mathcal{S}$-sentence, let s_1, s_2 be variable assignments over an $\mathcal{S}$-Modell M:*

It follows that $Val_{\mathfrak{M}, s_1}(\varphi) = Val_{\mathfrak{M}, s_2}(\varphi)$.

Proof. Since φ is assumed to be a sentence, $free(\varphi) = \varnothing$. Therefore, trivially, $s_1(x) = s_2(x)$ for all $x \in free(\varphi)$.
So we can apply the coincidence lemma, where in this case $\mathfrak{M}_1 = \mathfrak{M}_2 = \mathfrak{M}$, and we are done. ■

Remark 9 *We see that as far as sentences are concerned, it is irrelevant which variable assignment we choose in order to evaluate them:*
a sentence φ is true in a model $\mathfrak{M}$ relative to some variable assignment over $\mathfrak{M}$ iff φ is true in $\mathfrak{M}$ relative to all variable assignments over $\mathfrak{M}$.

Therefore we are entitled to write for sentences φ and sets Φ of sentences:

$$\mathfrak{M} \vDash \varphi \qquad \text{and} \qquad \mathfrak{M} \vDash \Phi$$

without mentioning a variable assignment s at all.

Here is an example of what a typical set of sentences satisfied by a model can look like:

Example 14 *(We use again $\mathcal{S}_{\text{OrdFie}} = \{\overline{+}, \bar{\cdot}, \overline{0}, \overline{1}, \overline{<}\}$ as our symbol set.)*

Let Φ be the following set of sentences:

- $\forall x \forall y \forall z \quad x \,\overline{+}\, (y \,\overline{+}\, z) \equiv (x \,\overline{+}\, y) \,\overline{+}\, z$
- $\forall x \quad x \,\overline{+}\, \overline{0} \equiv x$
- $\forall x \exists y \quad x \,\overline{+}\, y \equiv \; \overline{0}$
- $\forall x \forall y \quad x \,\overline{+}\, y \equiv y \,\overline{+}\, x$

$\hookrightarrow$ *These axioms describe the Abelian group* $(\mathbb{R}, +)$.

- $\forall x \forall y \forall z \quad x \,\bar{\cdot}\, (y \,\bar{\cdot}\, z) \equiv (x \,\bar{\cdot}\, y) \,\bar{\cdot}\, z$
- $\forall x \quad x \,\bar{\cdot}\, \overline{1} \equiv x$
- $\forall x \forall y \quad x \,\bar{\cdot}\, y \equiv y \,\bar{\cdot}\, x$
- $\forall x \; (\neg\, x \equiv \overline{0} \;\rightarrow\; \exists y \; x \,\bar{\cdot}\, y \equiv \; \overline{1})$

$\hookrightarrow$ *These axioms describe the Abelian group* $(\mathbb{R}\backslash\{0\}, \cdot)$.

- $\forall x \forall y \forall z \quad x \,\bar{\cdot}\, (y \,\overline{+}\, z) \equiv (x \,\bar{\cdot}\, y) \,\overline{+}\, (x \,\bar{\cdot}\, z)$

- $\neg\ \overline{0} \equiv \overline{1}$

  $\hookrightarrow$ *All axioms up to here taken together describe the real field* $(\mathbb{R}, +, \cdot)$.

- $\forall x\ \neg\, x \mathrel{\overline{<}} x$
- $\forall x \forall y \forall z\ (x \mathrel{\overline{<}} y \land y \mathrel{\overline{<}} z \to x \mathrel{\overline{<}} z)$
- $\forall x \forall y\ (x \mathrel{\overline{<}} y \lor x \equiv y \lor y \mathrel{\overline{<}} x)$
- $\forall x \forall y \forall z\ (x \mathrel{\overline{<}} y \to x \mathbin{\overline{+}} z \mathrel{\overline{<}} y \mathbin{\overline{+}} z)$
- $\forall x \forall y \forall z\ (x \mathrel{\overline{<}} y \land\ \overline{0} \mathrel{\overline{<}} z \to x \mathbin{\overline{\cdot}} z \mathrel{\overline{<}} y \mathbin{\overline{\cdot}} z)$

  $\hookrightarrow$ *All axioms taken together describe the real ordered field* $(\mathbb{R}, +, \cdot)$.

  Now let $\mathfrak{M}$ *be the model of the real ordered field (see p. 25): then* $\mathfrak{M} \models \Phi$.

Remark 10 *Why are these languages called* first-order*?*
Because there are also second-order *languages:*

- *first order:*
  $\forall x, \exists x$*: for all* members *of* D*, there is a* member *of* D
- *second-order:*
  $\forall x, \exists x$*: for all* members *of* D*, there is a* member *of* D
  But these languages have additional quantifiers of the form:
  $\forall X, \exists X$*: for all* subsets *of* D*, there is a* subset *of* D*.*

Final remark: From now on no lines above signs anymore (fortunately...).

3.2 Problem Set 2

1. Let $\mathcal{S} = \{P, R, f, g, c_0, c_1\}$, where P is a unary predicate, R is a binary predicate, and f and g are binary function signs. Let $\mathfrak{M} = (D, \mathfrak{I})$ be an $\mathcal{S}$-model with $D = \mathbb{R}$, such that $\mathfrak{I}(P) = \mathbb{N}$, $\mathfrak{I}(R)$ is the "larger than" ($>$) relation on $\mathbb{R}$, $\mathfrak{I}(f)$ is the addition mapping on $\mathbb{R}$, $\mathfrak{I}(g)$ is the multiplication mapping on $\mathbb{R}$, $\mathfrak{I}(c_0) = 0$, and $\mathfrak{I}(c_1) = 1$. Finally, let s be a variable assignment over $\mathfrak{M}$ with the property that $s(x) = 5$ and $s(y) = 3$ (where x, y, and z from below, are fixed pairwise distinct variables).

   Determine the following semantic values by step-by-step application of the definition clauses for $Val_{\mathfrak{M},s}$; subsequently, translate the terms/formulas into our usual mathematical "everyday" language:

   (a) $Val_{\mathfrak{M},s}(g(x, f(y, c_1)))$

   (b) $Val_{\mathfrak{M},s}(f(g(x, y), g(x, c_1)))$

   (c) $Val_{\mathfrak{M},s}(\forall x \forall y(R(x, c_0) \rightarrow \exists z(P(z) \wedge R(g(z, x), y))))$

   For which variable assignments s over $\mathfrak{M}$ is it the case that

   $P(z) \wedge R(z, c_1) \wedge \forall x(P(x) \wedge \exists y(P(y) \wedge g(x, y) \equiv z) \rightarrow x \equiv c_1 \vee x \equiv z)$

   is true at $\mathfrak{M}$ and s?

2. Let $\mathcal{S} = \{P, f\}$, where P is a unary predicate and f is a binary function sign.

   For each of the following formulas in $\mathcal{F}_\mathcal{S}$ find an $\mathcal{S}$-model and a corresponding variable assignment relative to which the formula is true and find an $\mathcal{S}$-model and a corresponding variable assignment relative to which the formula is false:

   (a) $\forall v_1 f(v_2, v_1) \equiv v_2$

   (b) $\exists v_2 \forall v_1 f(v_2, v_1) \equiv v_2$

   (c) $\exists v_2(P(v_2) \wedge \forall v_1 P(f(v_2, v_1)))$

3. Let D be finite and non-empty, let $\mathcal{S}$ be finite. Show that there are only finitely many $\mathcal{S}$-models with domain D.

4. A formula in which $\neg$, $\rightarrow$, $\leftrightarrow$ do not occur is called *positive*.

   Prove: For every positive formula there is a model and a variable assignment which taken together satisfy the formula (independent of what $\mathcal{S}$ is like).

   Hint: You might consider "trivial" models the domains of which only have one member.

5. Prove the *coincidence lemma* by induction over terms and formulas (see lemma 4).

3.3 Some Important Semantic Concepts

For everything that follows we fix a symbol set $\mathcal{S}$.

Definition 8 *For all* $\varphi \in \mathcal{F}_{\mathcal{S}}, \Phi \subseteq \mathcal{F}_{\mathcal{S}}$*:*
φ *follows logically from* Φ*, briefly:* $\Phi \vDash \varphi$ *iff*
for all $\mathcal{S}$*-models* $\mathfrak{M}$*, for all variable assignments* s *over* $\mathfrak{M}$*:*

$$\textit{if } \mathfrak{M}, s \vDash \Phi, \textit{ then } \mathfrak{M}, s \vDash \varphi$$

We also say equivalently:
Φ logically implies φ; φ is a logical consequence of Φ.

Careful: "$\vDash$" has different meanings in different contexts!

Example 15 *Let* $\mathcal{S} = \mathcal{S}_{Gr} = \{e, \circ\}$*:*
Let Φ *be the set that has the group axioms (G1), (G2), (G3) of p. 4 (formalised by means of* $\mathcal{S}_{Gr}$*) as its only members.*
It follows that $\Phi \vDash \forall x \; \exists y \; y \circ x \equiv e$*.*

This is because if $\mathfrak{M}$ *is a model of* Φ*, i.e.,* $\mathfrak{M} \vDash \Phi$*, then* $(D, \mathfrak{I}(\circ), \mathfrak{I}(e))$ *is a group (for the rest of the argument recall p.4f).*

(Note that "$\circ$*" on p.4f denotes the group multiplication function in a group, whereas here "*$\circ$*" denotes a function* sign. *In the present context, it is "*$\mathfrak{I}(\circ)$*" which denotes a group multiplication function.)*

Furthermore:
$\Phi \nvDash \forall x \forall y \; x \circ y \equiv y \circ x$
Counterexample: any non-Abelian group, e.g. S_3 *(the permutation group for a set with three elements).*

$\Phi \nvDash \neg \forall x \forall y \; x \circ y \equiv y \circ x$
Counterexample: any Abelian group, e.g. $(\mathbb{Z}, +)$*.*

We see that it is *not* generally the case that: $\Phi \nvDash \varphi \implies \Phi \vDash \neg\varphi$

$$\textit{BUT: } \mathfrak{M}, s \nvDash \varphi \implies \mathfrak{M}, s \vDash \neg\varphi$$

Now we single out important semantic concepts that apply to formulas of a particular type.

Some formulas have the property of being *true under all interpretations*:

Definition 9 *For all* $\varphi \in \mathcal{F}_{\mathcal{S}}$*:*
φ *is logically true iff*
for all $\mathcal{S}$*-models* $\mathfrak{M}$*, for all variable assignments* s *over* $\mathfrak{M}$*:*

$$\mathfrak{M}, s \vDash \varphi$$

Example 16
(i) $\varphi \vee \neg\varphi$*, (ii)* $\forall x \exists y\, x \equiv y$ *are logically true.*
(iii) $P(c)$*, (iv)* $\exists x P(x)$ *are not logically true.*

Some formulas are *true under some interpretation*:

Definition 10 *For all* $\varphi \in \mathcal{F}_{\mathcal{S}}, \Phi \subseteq \mathcal{F}_{\mathcal{S}}$*:*
φ *is satisfiable iff*
there is an $\mathcal{S}$*-model* $\mathfrak{M}$ *and a variable assignment* s *over* $\mathfrak{M}$*, such that:*
$\mathfrak{M}, s \vDash \varphi$*.*

$\Phi \subseteq \mathcal{F}_{\mathcal{S}}$ *is (simultaneously) satisfiable iff there are* $\mathfrak{M}, s$ *such that* $\mathfrak{M}, s \vDash \Phi$*.*

Example 17
(i) $\varphi \vee \neg\varphi$*, (ii)* $P(c)$*, (iii)* $\neg P(c)$ *are satisfiable.*
(iv) $\varphi \wedge \neg\varphi$*, (v)* $\neg\forall x\, x \equiv x$ *are not satisfiable.*
$\{P(c), \exists x\, Q(x, x)\}$ *is satisfiable.*
$\{P(c), \exists x\, Q(x, x), P(c) \rightarrow \forall x\, \neg Q(x, x)\}$ *is not satisfiable (i.e., not simultaneously satisfiable).*

Logical consequence, logical truth, and satisfiability are themselves logically related to each other:

Lemma 5 *For all* $\varphi \in \mathcal{F}_{\mathcal{S}}, \Phi \subseteq \mathcal{F}_{\mathcal{S}}$*:*

1. φ *is logically true iff* $\varnothing \models \varphi$*.*
2. $\Phi \models \varphi$ *iff* $\Phi \cup \{\neg\varphi\}$ *is not satisfiable.*
3. φ *is logically true iff* $\neg\varphi$ *is not satisfiable.*

Proof.

1. $\varnothing \models \varphi \iff$ for all $\mathfrak{M}, s$: if $\mathfrak{M}, s \models \varnothing$, then $\mathfrak{M}, s \models \varphi$
   (but $\mathfrak{M}, s \models \varnothing$ is true for trivial reasons, because what it means is

   for all ψ: if $\psi \in \varnothing$, then $\mathfrak{M}, s \models \psi$

   and the "if"-part of this sentence is false for all ψ)
   $\iff$ for all $\mathfrak{M}, s$: $\mathfrak{M}, s \models \varphi$
   $\iff$ φ is logically true (by def.) ✓
2. $\Phi \models \varphi \iff$ for all $\mathfrak{M}, s$: if $\mathfrak{M}, s \models \Phi$, then $\mathfrak{M}, s \models \varphi$
   $\iff$ not there are $\mathfrak{M}, s$, such that: $\mathfrak{M}, s \models \Phi, \mathfrak{M}, s \not\models \varphi$
   $\iff$ not there are $\mathfrak{M}, s$, such that: $\mathfrak{M}, s \models \Phi, \mathfrak{M}, s \models \neg\varphi$
   $\iff$ not there are $\mathfrak{M}, s$, such that: $\mathfrak{M}, s \models \Phi \cup \{\neg\varphi\}$
   $\iff$ $\Phi \cup \{\neg\varphi\}$ not satisfiable (by def.) ✓
3. φ is logically true
   $\iff \varnothing \models \varphi$ (by 1.)
   $\iff \{\neg\varphi\}$ is not satisfiable (by 2.)
   $\iff \neg\varphi$ is not satisfiable ✓

■

Sometimes two formulas "*say the same*":

Definition 11 *For all* $\varphi, \psi \in \mathcal{F}_{\mathcal{S}}$*:*
φ *is logically equivalent to* ψ *iff* $\varphi \models \psi$ and $\psi \models \varphi$*.*

Example 18

- *$\varphi \wedge \psi$ is logically equivalent to $\neg(\neg\varphi \vee \neg\psi)$*
- *$\varphi \to \psi$ is logically equivalent to $\neg\varphi \vee \psi$*
- *$\varphi \leftrightarrow \psi$ is logically equivalent to $\neg(\varphi \vee \psi) \vee \neg(\neg\varphi \vee \neg\psi)$*
- *$\forall x\varphi$ is logically equivalent to $\neg\exists x\neg\varphi$*

This allows us to focus just on $\neg$, $\vee$, $\exists$ in all that follows!

E.g.:

$$\begin{aligned} \forall x\,(P(x) \wedge Q(x)) &\text{ is logically equivalent to} \\ \neg\exists x\neg\neg\,(\neg P(x) \vee \neg Q(x)) &\text{ is logically equivalent to} \\ \neg\exists x\,(\neg P(x) \vee \neg Q(x)) & \end{aligned}$$

Remark 11 *We said that we "fixed" a symbol set $\mathcal{S}$ at the beginning of this subsection. But strictly we have defined "$\mathcal{S}$-logically implies", "$\mathcal{S}$-logically true", "$\mathcal{S}$-satisfiable", "$\mathcal{S}$-logically equivalent" for* arbitrary *symbol set $\mathcal{S}$ (it is just so awkward to keep the reference to $\mathcal{S}$ while using all these notions)!*

Fortunately, the particular choice of a symbol set often does not really matter so much. E.g.:

Let $\mathcal{S}, \mathcal{S}'$ be symbol sets, such that $\mathcal{S} \subseteq \mathcal{S}'$.
Let φ be an $\mathcal{S}$-formula ($\Longrightarrow$ φ is also an $\mathcal{S}'$-formula).
Then:
φ is $\mathcal{S}$-satisfiable iff φ is $\mathcal{S}'$-satisfiable.

Proof.
($\Rightarrow$) Assume that φ is $\mathcal{S}$-satisfiable.
By definition, there is an $\mathcal{S}$-model $\mathfrak{M}$ and there is a variable assignment s over $\mathfrak{M}$, such that:

$$\mathfrak{M}, s \vDash \varphi \qquad \text{(i.e., } Val_{\mathfrak{M},s}(\varphi) = 1\text{)}$$

Now we define an $\mathcal{S}'$-model $\mathfrak{M}'$: let $D' := D$, $\mathfrak{I}' \mid_{\mathcal{S}} \equiv \mathfrak{I}$ (i.e., $\mathfrak{I}'$ and $\mathfrak{I}$ are identical on $\mathcal{S}$), $\mathfrak{I}'$ on $\mathcal{S}' \backslash \mathcal{S}$ is chosen arbitrarily.

Furthermore, let $s' := s$.

By the coincidence lemma (lemma 4) it follows that $\mathfrak{M}', s' \vDash \varphi$ (since φ is an $\mathcal{S}$-formula, the symbols in $\mathcal{S}$ are interpreted in the same way by $\mathfrak{I}'$ and $\mathfrak{I}$, and the two models $\mathfrak{M}'$ and $\mathfrak{M}$ have the same domain).
Hence, φ is $\mathcal{S}'$-satisfiable. ($\mathfrak{M}'$ is called an *expansion* of $\mathfrak{M}$.) ✓

($\Leftarrow$) Analogously (in this case one simply "forgets" about the interpretation of symbols in $\mathcal{S}'\backslash\mathcal{S}$: this yields a so-called *reduct* of $\mathfrak{M}'$). ■

By lemma 5: analogously for logical consequence, logical truth, and so forth.

3.4 Problem Set 3

1. The convergence of a real-valued sequence (x_n) to a limit x is usually defined as follows:

   (Conv) For all $\epsilon > 0$ there is a natural number n, such that for all natural numbers $m > n$ it holds that: $|x_m - x| < \epsilon$

   Represent (Conv) in a first-order language by choosing an appropriate symbol set $\mathcal{S}$ and define the corresponding $\mathcal{S}$-model.

   Hint: (i) Real sequences are functions from $\mathbb{N}$ to $\mathbb{R}$, i.e., you may consider x_m as being of the form $f(m)$; f can be regarded as being defined on $\mathbb{R}$ even though only its values for members of $\mathbb{N}$ are "relevant". (ii) $|x_m - x|$ may either be considered as the result of applying a binary "distance" function to the arguments x_m and x or as the result of applying two functions – subtraction and absolute value – to these arguments.

2. (This problem counts for award of CREDIT POINTS.)
   Show that for arbitrary $\mathcal{S}$-formulas φ, ψ, ρ, and arbitrary sets Φ of $\mathcal{S}$-formulas the following is the case:

   (a) $(\varphi \vee \psi) \models \rho$ iff $\varphi \models \rho$ and $\psi \models \rho$.

   (b) $\Phi \cup \{\varphi\} \models \psi$ iff $\Phi \models (\varphi \rightarrow \psi)$.

   (c) $\varphi \models \psi$ (i.e., $\{\varphi\} \models \psi$) iff $(\varphi \rightarrow \psi)$ is logically true.

3. (a) Prove for arbitrary $\mathcal{S}$-formulas φ, ψ:
   $\exists x \forall y \varphi \models \forall y \exists x \varphi$

   (b) Show that the following is *not* the case for all $\mathcal{S}$-formulas φ, ψ:
   $\forall y \exists x \varphi \models \exists x \forall y \varphi$

4. (a) Prove for all $\mathcal{S}$-formulas φ, ψ:
   $\exists x(\varphi \vee \psi)$ is logically equivalent to $\exists x \varphi \vee \exists x \psi$.

   (b) Show that the following is *not* the case for all $\mathcal{S}$-formulas φ, ψ:
   $\exists x(\varphi \wedge \psi)$ is logically equivalent to $\exists x \varphi \wedge \exists x \psi$.

5. Let Φ be an $\mathcal{S}$-formula set, let φ und ψ be $\mathcal{S}$-formulas. Show:

   If $\Phi \cup \{\varphi\} \models \psi$ and $\Phi \models \varphi$, then $\Phi \models \psi$.

6. A set Φ of $\mathcal{S}$-sentences is called "independent if and only if there is no $\varphi \in \Phi$ such that: $\Phi \setminus \{\varphi\} \vDash \varphi$ (i.e., no φ is "redundant", because it is impossible to conclude φ from $\Phi \setminus \{\varphi\}$).

   Prove: (a) the set of the three group axioms and (b) the set of the three axioms for equivalence structures are both independent (see chapter one for these axioms).

3.5 Substitution and the Substitution Lemma

In the next chapter on the sequence calculus we will consider derivation rules by which terms t can be *substituted* for variables x. E.g.:

Example 19

- *From*

  $$\forall x \; P(x)$$

  one can conclude

  $$P(t) \; \} \; \textit{formula that results from } P(x) \textit{ by substituting } t \textit{ for } x$$

- *From*

  $$P(t) \; \} \; \textit{formula that results from } P(x) \textit{ by substituting } t \textit{ for } x$$

  one can conclude

  $$\exists x \; P(x)$$

- *From*

  $$P(t_1)\} \; \textit{formula that results from } P(x) \textit{ by substituting } t_1 \textit{ for } x$$
  $$t_1 \equiv t_2$$

  one can conclude

  $$P(t_2) \; \} \; \textit{formula that results from } P(x) \textit{ by substituting } t_2 \textit{ for } x$$

But one has to be *careful*:

Should we be allowed to draw an inference from

$$\forall x \; \exists y \quad y < x \; \} \text{ true in the model of the real ordered field}$$

to

$$\exists y \quad y < y \; \} \text{ false in the model of the real ordered field}$$

by a substitution of y for x?

No!

Why does this last substitution go wrong?
In

$$\exists y \; y < x$$

the variable x occurs freely, but in

$$\exists y \; y < y$$

the second occurrence of y, which was substituted for x, is bound!
$\Longrightarrow$ This corresponds to a change of meaning.

Problems like these can be avoided in the following manner:
Draw an inference from $\underbrace{\forall x \exists y \; y < x}_{\downarrow}$

convert into: $\forall x \exists u \; u < x$
then: substitute y for x

to $\exists u \; u < y$
(this last formula is true in the model of the real ordered field – independent of which real number is chosen to be $s(y)$).

Wanted:

1. An "intelligent" substitution function $\frac{t}{x}$ by which a term t is substituted for free occurrences of a variable x (substitutions for bound occurrences should be prohibited), such that problematic cases as the above one are avoided by automatic renaming of bound variables.

2. It should be possible to substitute terms $t_0, \ldots, t_n$ "simultaneously" for pairwise distinct variables $x_0, \ldots, x_n$ (t_0 for x_0, t_1 for $x_1, \ldots$).

   E.g.: $[P(x,y)]\frac{y,x}{x,y} = P(y,x)$

   (y is substituted for x and simultaneously x is substituted for y.)

   BUT: $[[P(x,y)\frac{y}{x}]\frac{x}{y}] = [P(y,y)]\frac{x}{y} = P(x,x)$!

   (So simultaneous substitution cannot be replaced by the iteration of simple substitutions.)

3. We want:
   $Val_{\mathfrak{M},s}\left(t\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}\right) = Val_{\mathfrak{M},s\frac{Val_{\mathfrak{M},s}(t_0),\ldots,Val_{\mathfrak{M},s}(t_n)}{x_0,\ldots,x_n}}(t)$

4. We want:

$$\mathfrak{M}, s \models \varphi\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} \iff \mathfrak{M}, s\frac{Val_{\mathfrak{M},s}(t_0),\ldots,Val_{\mathfrak{M},s}(t_n)}{x_0,\ldots,x_n} \models \varphi$$

So we also need a generalized way of manipulating variable assignments; let $x_0, \ldots, x_n$ be pairwise distinct variables:

$$s\frac{d_0,\ldots,d_n}{x_0,\ldots,x_n}(y) = \begin{cases} s(y) & \text{if } y \neq x_0, \ldots, x_n \\ d_i & \text{if } y = x_i \ \ (\text{for } 0 \leq i \leq n) \end{cases}$$

This is how a substitution function that satisfies 1.–4. can be defined:

Definition 12 *Let $\mathcal{S}$ be an arbitrary symbol set.*
Let $t_0, \ldots, t_n \in \mathcal{T}_\mathcal{S}$, let $x_0, \ldots, x_n$ be pairwise distinct variables.
We define the substitution function $\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}$ on $\mathcal{T}_\mathcal{S} \cup \mathcal{F}_\mathcal{S}$ as follows:

- $[x]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} := \begin{cases} t_i & \text{for } x = x_i \ (0 \leq i \leq n) \\ x & \text{else} \end{cases}$
- $[c]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} := c$
- $[f(t'_1, \ldots, t'_m)]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} := f\left([t'_1]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}, \ldots, [t'_m]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}\right)$
- $[t'_1 \equiv t'_2]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} := [t'_1]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} \equiv [t'_2]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}$
- $[P(t'_1, \ldots, t'_m)]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}$ *is defined analogously to the case of* $\equiv$
- $[\neg\varphi]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} := \neg[\varphi]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}$
- $[\varphi \vee \psi]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} := ([\varphi]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} \vee [\psi]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n})$

  (accordingly for $\wedge$, $\rightarrow$, $\leftrightarrow$)
- $\exists x\varphi$*:*

  Let $x_{i_1}, \ldots, x_{i_k}$ be those variables x_i among $x_0, \ldots, x_n$ for which it holds that:

  - $x_i \in free(\exists x\varphi)$
  - $x_i \neq t_i$

  (Call these variables the relevant *variables of the substitution.)*

$[\exists x\varphi]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} := \ \exists u\ [\varphi]\frac{t_{i_1},\ldots,t_{i_k},u}{x_{i_1},\ldots,x_{i_k},x}$

where $u := x$*, if* x *does not occur in* $t_{i_1},\ldots,t_{i_k}$

else: let u *be the first variable in* $v_0, v_1, v_2, \ldots$ *that does not occur in* $\varphi, t_{i_1},\ldots,t_{i_k}$

Note that we use '[.]' in order to make clear to what term or formula we apply the substitution function $\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}$.

Remark 12 *Consider the substitution case for* $\exists x\varphi$*:*

- x *is certainly distinct from any of* $x_{i_1},\ldots,x_{i_k}$*, because* $x \notin free(\exists x\varphi)$*.*
- *Assume there are no variables* x_i *with* $x_i \in free(\exists x\varphi)$ *and* $x_i \neq t_i$ *(so there are no relevant variables)*

  $\Longrightarrow\ k = 0$

  $\Longrightarrow$ *there are no* $t_{i_1},\ldots,t_{i_k}$ *to consider*

  $\Longrightarrow$ x *does not occur within* $t_{i_1},\ldots,t_{i_k}$

  $\Longrightarrow\ u = x$ *(there is nothing to rename)*

  $\Longrightarrow [\exists x\varphi]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} = \exists u[\varphi]\frac{u}{x} = \exists x[\varphi]\frac{x}{x} = \exists x\varphi$
- *It follows from the definition of our substitution function that* u *does not occur within* $t_{i_1},\ldots,t_{i_k}$*.*

Example 20 *(For two variables* x, y *with* $x \neq y$*:)*

- $[\exists y\ y < x]\frac{y}{x}$

  $= \exists u\ [y < x]\frac{y,u}{x,y}$ *(since* y *occurs within* t_{i_1}*, i.e., within* y*, it follows from our definition that* u *must be distinct from* x, y*)*

  $= \exists u\ [y]\frac{y,u}{x,y} < [x]\frac{y,u}{x,y}$

  $= \exists u\ u < y$ ✓

  The renaming works!

  (Later we will use this in order to draw inferences such as the one from $\forall x\exists y\ y < x$ *to* $\exists u\ u < y$*.)*

- $[\exists y\ y < x]\frac{x}{y}$ *(since y is not free in $\exists y\ y < x$, only the substitution for u is going to remain)*

  $= \exists u[y < x]\frac{u}{y}$ *(since the number k of relevant variables is in this case 0, there are no $t_{i_1}, \ldots, t_{i_k}$ in which y could occur, thus it follows that $u = y$)*

  $= \exists y[y < x]\frac{y}{y}$

  $= \exists y\ y < x$

  We see that nothing can be substituted for bound variables.

- $[\exists v_0\ P(v_0, f(v_1, v_2))]\frac{v_0, v_2, v_4}{v_1, v_2, v_0}$

  (in the second substitution we have $x_1 = t_1 = v_2$ and in the third substitution v_0 is bound in $\exists v_0\ P(v_0, f(v_1, v_2))$, so these two substitutions are omitted; furthermore, since v_0 occurs within $t_{i_1} = v_0$, it follows that $u = v_3$ because v_3 is the variable with least index that does not occur in $\varphi, t_{i_1}, \ldots, t_{i_k}$, i.e., in $\exists v_0 P(v_0, f(v_1, v_2)), v_0$)

  $= \exists v_3[P(v_0, f(v_1, v_2))]\frac{v_0, v_3}{v_1, v_0}$

  $= \exists v_3 P(v_3, f(v_0, v_2))$

Lemma 6 *(Substitution Lemma)*
Let $\mathfrak{M}$ be an $\mathcal{S}$-model:

1. *For all terms $t \in \mathcal{T}_\mathcal{S}$:*

   For all variable assignments s over $\mathfrak{M}$, for all terms $t_0, \ldots, t_n \in \mathcal{T}_\mathcal{S}$, for all pairwise distinct variables $x_0, \ldots, x_n$:

   $$Val_{\mathfrak{M},s}(t\frac{t_0, \ldots, t_n}{x_0, \ldots, x_n}) = Val_{\mathfrak{M},s\frac{Val_{\mathfrak{M},s}(t_0), \ldots, Val_{\mathfrak{M},s}(t_n)}{x_0, \ldots, x_n}}(t)$$

2. *For all formulas $\varphi \in \mathcal{F}_\mathcal{S}$:*

   For all variable assignments s over $\mathfrak{M}$, for all terms $t_0, \ldots, t_n \in \mathcal{T}_\mathcal{S}$, for all pairwise distinct variables $x_0, \ldots, x_n$:

   $$\mathfrak{M}, s \models \varphi\frac{t_0, \ldots, t_n}{x_0, \ldots, x_n} \text{ iff } \mathfrak{M}, s\frac{Val_{\mathfrak{M},s}(t_0), \ldots, Val_{\mathfrak{M},s}(t_n)}{x_0, \ldots, x_n} \models \varphi$$

Proof. By induction over terms and formulas:
Concerning 1:

- $t = c:$ ✓

- $t = x:$

  a) assume $x \neq x_0, \ldots, x_n$

  $\Longrightarrow [x]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} = x$

  $\Longrightarrow Val_{\mathfrak{M},s}(x\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}) = Val_{\mathfrak{M},s}(x) = s(x)$

  $= s\frac{Val_{\mathfrak{M},s}(t_0),\ldots,Val_{\mathfrak{M},s}(t_n)}{x_0,\ldots,x_n}(x)$

  $= Val_{\mathfrak{M},s\frac{Val_{\mathfrak{M},s}(t_0),\ldots,Val_{\mathfrak{M},s}(t_n)}{x_0,\ldots,x_n}}(x)$ ✓

  b) assume $x = x_i$ for $0 \leq i \leq n$

  $\Longrightarrow [x]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} = t_i$

  $\Longrightarrow Val_{\mathfrak{M},s}(x\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}) = Val_{\mathfrak{M},s}(t_i)$

  $= s\frac{Val_{\mathfrak{M},s}(t_0),\ldots,Val_{\mathfrak{M},s}(t_n)}{x_0,\ldots,x_n}(x_i)$

  $= Val_{\mathfrak{M},s\frac{Val_{\mathfrak{M},s}(t_0),\ldots,Val_{\mathfrak{M},s}(t_n)}{x_0,\ldots,x_n}}\underbrace{(x_i)}_{=\,x\text{ by ass.}}$ ✓

- $t = f(t'_1, \ldots, t'_m)$: by inductive assumption ✓

Concerning 2:

- $\varphi = P(t'_1, \ldots, t'_m)$:

  $\mathfrak{M}, s \models [P(t'_1, \ldots, t'_m)]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}$

  $\Longleftrightarrow \mathfrak{M}, s \models P\left([t'_1]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}, \ldots, [t'_m]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}\right)$

  $\Longleftrightarrow \left(Val_{\mathfrak{M},s}\left([t'_1]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}\right), \ldots, Val_{\mathfrak{M},s}\left([t'_m]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}\right)\right) \in \mathfrak{I}(P)$
  (def. of Val)

  $\Longleftrightarrow \left(Val_{\mathfrak{M},s\frac{Val_{\mathfrak{M},s}(t_0),\ldots,Val_{\mathfrak{M},s}(t_n)}{x_0,\ldots,x_n}}(t'_1), \ldots, Val_{\mathfrak{M},s\frac{Val_{\mathfrak{M},s}(t_0),\ldots,Val_{\mathfrak{M},s}(t_n)}{x_0,\ldots,x_n}}(t'_m)\right) \in$
  $\mathfrak{I}(P)$ (by 1.)

  $\Longleftrightarrow \mathfrak{M}, s\frac{Val_{\mathfrak{M},s}(t_0),\ldots,Val_{\mathfrak{M},s}(t_n)}{x_0,\ldots,x_n} \models P(t'_1, \ldots, t'_m)$ ✓

- $\varphi = (t'_1 \equiv t'_m)$: analogously ✓

- $\neg$, $\vee$: obvious by inductive assumption ✓

- $\wedge,\ \rightarrow,\ \leftrightarrow,\ \forall$: need not be treated as extra cases (express them in terms of $\vee,\ \neg,\ \exists$) ✓

- $\varphi = \exists x \psi$: we assume inductively that 2. from above is true of ψ

$\mathfrak{M}, s \models [\exists x\ \psi]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}$

$\iff \mathfrak{M}, s \models \exists u\ [\psi]\frac{t_{i_1},\ldots,t_{i_k},u}{x_{i_1},\ldots,x_{i_k},x}$ (def. of substitution)

$\iff$ there is a $d \in D$, such that:

$\mathfrak{M}, s\frac{d}{u} \models [\psi]\frac{t_{i_1},\ldots,t_{i_k},u}{x_{i_1},\ldots,x_{i_k},x}$ (def. of Val)

$\iff$ there is a $d \in D$, such that:

$\mathfrak{M}, \left(s\frac{d}{u}\right)\frac{Val_{\mathfrak{M},s\frac{d}{u}}(t_{i_1}),\ldots,Val_{\mathfrak{M},s\frac{d}{u}}(t_{i_k}),Val_{\mathfrak{M},s\frac{d}{u}}(u)}{x_{i_1},\ldots,x_{i_k},x} \models \psi$ (by ind. ass.)

$\iff$ there is a $d \in D$, such that:

$\mathfrak{M}, \left(s\frac{d}{u}\right)\frac{Val_{\mathfrak{M},s}(t_{i_1}),\ldots,Val_{\mathfrak{M},s}(t_{i_k}),d}{x_{i_1},\ldots,x_{i_k},x} \models \psi$

(by the coincidence lemma, which we can apply because u not in $t_{i_1},\ldots,t_{i_k}$ – see p. 44)

$\iff$ there is a $d \in D$, such that:

$\mathfrak{M}, s\frac{Val_{\mathfrak{M},s}(t_{i_1}),\ldots,Val_{\mathfrak{M},s}(t_{i_k}),d}{x_{i_1},\ldots,x_{i_k},x} \models \psi$

(Because:

in case $u = x$: one of the $\frac{d}{u}$ is superfluous and can be omitted

in case $u \neq x$: $u \notin \psi$ $\Rightarrow$ apply coincidence lemma)

$\iff$ there is a $d \in D$, such that:

$\mathfrak{M}, \left(s\frac{Val_{\mathfrak{M},s}(t_{i_1}),\ldots,Val_{\mathfrak{M},s}(t_{i_k})}{x_{i_1},\ldots,x_{i_k}}\right)\frac{d}{x} \models \psi$ ($x \neq x_{i_1},\ldots,x_{i_k}$ – see p. 44)

$\iff \mathfrak{M}, s\frac{Val_{\mathfrak{M},s}(t_{i_1}),\ldots,Val_{\mathfrak{M},s}(t_{i_k})}{x_{i_1},\ldots,x_{i_k}} \models \exists x\ \psi$ (def. of $Val(\exists x\psi)$)

$\iff \mathfrak{M}, s\frac{Val_{\mathfrak{M},s}(t_0),\ldots,Val_{\mathfrak{M},s}(t_n)}{x_0,\ldots,x_n} \models \exists x\ \psi$

Since: for $i \notin \{i_1,\ldots,i_k\}$ (i.e., for the index of an *irrelevant* variable) either of the following must be the case:

i) $x_i \notin free(\exists x\psi) \Rightarrow$ apply coincidence lemma (Val only depends on free variables)! ✓

ii) $x_i = t_i$: $Val_{\mathfrak{M},s}(t_i) = Val_{\mathfrak{M},s}(x_i) = s(x_i)$

before the equivalence sign: $x_i \mapsto s(x_i)$
after the equivalence sign: $x_i \mapsto Val_{\mathfrak{M},s}(t_i) = s(x_i)$ ✓

■

Remark 13 *It is easy to express* unique existence *by means of substitution:*
$\exists! x\varphi \ := \ \exists x(\varphi \wedge \forall y(\varphi\frac{y}{x} \rightarrow y \equiv x))$

By the substitution lemma:
$\mathfrak{M}, s \vDash \exists! x\varphi \iff$
there is one and only one $d \in D$, *such that:* $\mathfrak{M}, s\frac{d}{x} \vDash \varphi$

3.6 Problem Set 4

1. Show:

   (a) $[\exists v_0 \exists v_1 (P(v_0, v_2) \wedge P(v_1, v_3))] \frac{v_2 \quad v_2 \quad v_2}{v_0 \quad v_1 \quad v_3}$
   $= \exists v_0 \exists v_1 (P(v_0, v_2) \wedge P(v_1, v_2))$

   (b) $[\exists v_0 \exists v_1 (P(v_0, v_2) \wedge P(v_1, v_3))] \frac{v_3 \quad f(v_2, v_3)}{v_2 \quad v_3}$
   $= \exists v_0 \exists v_1 (P(v_0, v_3) \wedge P(v_1, f(v_2, v_3)))$

   (c) $[\exists v_0 \exists v_1 (P(v_0, v_2) \wedge P(v_1, v_3))] \frac{v_2 \quad v_0 \quad f(v_2, v_3)}{v_0 \quad v_2 \quad v_3}$
   $= \exists v_4 \exists v_1 (P(v_4, v_0) \wedge P(v_1, f(v_2, v_3)))$

   (d) $[\forall v_0 \exists v_1 (P(v_0, v_1) \wedge P(v_0, v_2)) \vee \exists v_2 f(v_2, v_2) \equiv v_0] \frac{v_0 \quad f(v_0, v_1)}{v_0 \quad v_2}$
   $= \forall v_3 \exists v_4 (P(v_3, v_4) \wedge P(v_3, f(v_0, v_1))) \vee \exists v_2 f(v_2, v_2) \equiv v_0$

2. Let $t_0, \ldots, t_n$ be $\mathcal{S}$-terms, $x_0, \ldots, x_n$ pairwise distinct variables, φ an $\mathcal{S}$-formula and y a variable.

   Prove:

   (a) If π is a permutation of the numbers $0, \ldots, n$, then:
   $\varphi \frac{t_0, \ldots, t_n}{x_0, \ldots, x_n} = \varphi \frac{t_{\pi(0)}, \ldots, t_{\pi(n)}}{x_{\pi(0)}, \ldots, x_{\pi(n)}}$

   (b) If $y \in var(t \frac{t_0, \ldots, t_n}{x_0, \ldots, x_n})$, then

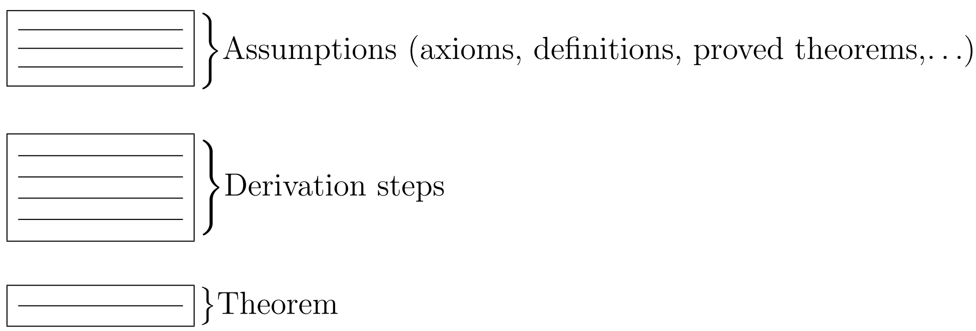
      i. $y \in var(t_0) \cup \ldots \cup var(t_n)$ or
      ii. $y \in var(t)$ and $y \neq x_0, \ldots, x_n$.

4 A Sequent Calculus for First-Order Predicate Logic

In the following we will make precise what we mean by a "mathematical proof".

4.1 Preliminary Remarks on the Sequent Calculus

(Informal) Mathematical proofs have roughly a structure like this:



We are going to call sequences of sentences like this "*sequents*". If such sequents correspond to actual proofs, we will call them *"correct"*.

More formally: Let $\mathcal{S}$ be an arbitrary symbol set.

An ($\mathcal{S}$-)*sequent* is a finite sequence

$$\varphi_1\varphi_2\ldots\varphi_{n-1}\varphi_n$$

of $\mathcal{S}$-formulas ($n \geq 1$).

We call

- $\varphi_1\varphi_2\ldots\varphi_{n-1}$ the *antecedent* of the sequent ("assumptions")
- φ_n the *consequent* of the sequent (i.e., the formula for which we want to claim that it follows from the assumptions)

While proving theorems informally, we implicitly "manipulate" sequents in various ways (we extend sequents, we drop auxiliary assumptions,...).

E.g., indirect proofs:
add the negation of what you want to prove;
derive a contradiction;
conclude the sentence you intended to prove.

Formally:

- Show that both

  $\varphi_1 \ldots \varphi_k \; \neg\psi \; \rho$

  and

  $\varphi_1 \ldots \varphi_k \; \neg\psi \; \neg\rho$

  are correct.

- Conclude that

  $\varphi_1 \ldots \varphi_k \; \psi$ (¬ψ is dropped because it led to a contradiction)

  is correct.

We have to make "show" and "conclude" precise. What we need is (i) rules which introduce sequents that are obviously correct, like

$$\varphi \, \varphi$$

and (ii) rules which lead from correct sequents to further correct sequents, e.g.,

$$\left.\begin{array}{l}\varphi_1 \ldots \varphi_k \; \neg\psi \; \rho \\ \varphi_1 \ldots \varphi_k \; \neg\psi \; \neg\rho\end{array}\right\} \text{Premises}$$
$$\overline{\varphi_1 \ldots \varphi_k \; \psi} \; \} \text{ Conclusion}$$

Rules such as the latter will often be abbreviated, e.g., by

$$\begin{array}{l}\Gamma \; \neg\psi \; \rho \\ \Gamma \; \neg\psi \; \neg\rho \\ \hline \Gamma \; \psi\end{array}$$

(so we use "Γ", "Δ",... as variables for sequents).

The sequent calculus is a specific set of such rules (some have premises, some have not). We will see that its premise-free rules only lead to correct sequents and that its rules with premises lead from correct sequents to other correct sequent (so these rules are "correctness-preserving").

But what is the *correctness* of a sequent?

Definition 13 *For all $\mathcal{S}$-sequents $\Gamma\ \varphi$:*
$\Gamma\ \varphi$ *is correct :iff* $\underbrace{\{\psi|\psi \text{ is sequence member of } \Gamma\}}_{\text{briefly: } \Gamma} \vDash \varphi$

(We will exploit the systematic ambiguity of "Γ" and other capital Greek letters: in some contexts they will denote *sequents* of formulas, in other contexts they will denote *sets* of formulas, in particular, sets of members of a sequent – never mind...).

Once we are given the rules of the sequent calculus, *derivability of sequents in the sequent calculus* can be defined analogously to *derivability of terms in the terms calculus.*

Finally, we can define the derivability of *formulas* from other *formulas* on basis of the derivability of sequents:

Definition 14 *Let Φ be a set of $\mathcal{S}$-formulas, let φ be an $\mathcal{S}$-formula:*
φ is derivable *from Φ, briefly: $\Phi \vdash \varphi$:iff*
there are $\varphi_1, \ldots, \varphi_n \in \Phi$, such that

$$\varphi_1 \ldots \varphi_n \varphi$$

is derivable in the sequent calculus.

Our goal is to prove:

$$\Phi \vdash \varphi \iff \Phi \vDash \varphi \qquad \textit{(Soundness and Completeness Theorem)}$$

$\Longrightarrow$: Soundness of the sequent calculus
$\Longleftarrow$: Completeness of the sequent calculus

Lemma 7 $\Phi \vdash \varphi \iff$ *there is a finite set $\Phi' \subseteq \Phi$ such that $\Phi' \vdash \varphi$*

Proof.
($\Rightarrow$:) follows from the definition of $\vdash$ ✓
($\Leftarrow$:) ✓ ■

Now we are going to introduce the rules of the sequent calculus. These rules are divided into the following groups: basic rules; rules for propositional connectives; rules for quantifiers; rules for equality.

4.2 Basic Rules

Antecedent rule: (Ant.)

$$\frac{\Gamma\varphi}{\Gamma'\varphi} \qquad \text{(for } \Gamma \subseteq \Gamma'\text{)}$$

("$\Gamma \subseteq \Gamma'$" means: every sequence member of Γ is a sequence member of Γ')

Explanation:

We are always allowed to add assumptions and we are always allowed to permute them.

Correctness:
Proof. Assume that $\Gamma\varphi$ is correct, i.e., $\Gamma \vDash \varphi$.
Let Γ' be a set of $\mathcal{S}$-formulas, such that $\Gamma' \supseteq \Gamma$.
Let $\mathfrak{M}, s$ be chosen arbitrarily, such that $\mathfrak{M}, s \vDash \Gamma'$.
$\Longrightarrow \mathfrak{M}, s \vDash \Gamma$
$\Longrightarrow \mathfrak{M}, s \vDash \varphi$ (by assumption)
It follows that $\Gamma' \vDash \varphi$, i.e., $\Gamma'\varphi$ is correct. ■

Assumption rule: (Ass.)

$$\frac{}{\Gamma\varphi} \qquad \text{(for } \varphi \text{ being a sequence member of } \Gamma\text{)}$$

Explanation:

We are always allowed to conclude assumptions from themselves.

Correctness:
Proof.
If $\varphi \in \Gamma$, then certainly $\Gamma \vDash \varphi$; hence, $\Gamma\varphi$ is correct. ■

4.3 Rules for Propositional Connectives

Proof by cases: (PC)

$$\begin{array}{l} \Gamma \;\; \psi \; \varphi \\ \underline{\Gamma \neg\psi \; \varphi} \\ \;\Gamma \; \varphi \end{array} \qquad \left(\text{e.g. } \begin{array}{r} x \geq 0 \Rightarrow \varphi \\ x < 0 \Rightarrow \varphi \end{array} \right\} \Longrightarrow \varphi \checkmark \right)$$

Explanation:

If we can show φ both under the assumption ψ and under the assumption $\neg\psi$ (and since one of these two assumptions must actually be the case), we are allowed to conclude φ without assuming anything about ψ or $\neg\psi$.

Note that this rule allows us to *shorten* sequents.

Correctness:
Proof. Assume that $\Gamma\psi\varphi$, $\Gamma\neg\psi\varphi$ are correct, i.e.,
$\Gamma \cup \{\psi\} \vDash \varphi$, $\Gamma \cup \{\neg\psi\} \vDash \varphi$

Let $\mathfrak{M}, s$ be chosen arbitrarily such that $\mathfrak{M}, s \vDash \Gamma$.

There are two possible cases:
Case 1: $\mathfrak{M}, s \vDash \psi$
$\Longrightarrow \mathfrak{M}, s \vDash \Gamma \cup \{\psi\}$
$\Longrightarrow \mathfrak{M}, s \vDash \varphi$ ✓ (by assumption)

Case 2: $\mathfrak{M}, s \nvDash \psi$
$\Longrightarrow \mathfrak{M}, s \vDash \neg\psi$
$\Longrightarrow \mathfrak{M}, s \vDash \Gamma \cup \{\neg\psi\}$
$\Longrightarrow \mathfrak{M}, s \vDash \varphi$ ✓ (by assumption)

$\Longrightarrow \Gamma \vDash \varphi \Longrightarrow \Gamma\varphi$ is correct. ■

<u>Contradiction:</u> (CD)

$$\frac{\begin{array}{l}\Gamma \neg\psi\ \rho \\ \Gamma \neg\psi \neg\rho\end{array}}{\Gamma\ \psi}$$

Explanation:

If assuming $\neg\psi$ leads to a contradiction, then we are allowed to infer ψ.

Correctness:

Proof. Assume that $\Gamma\neg\psi\rho,\ \Gamma\neg\psi\neg\rho$ are correct, i.e.,
$\Gamma \cup \{\neg\psi\} \vDash \rho,\ \Gamma \cup \{\neg\psi\} \vDash \neg\rho.$

So for all $\mathfrak{M}, s$ with $\mathfrak{M}, s \vDash \Gamma \cup \{\neg\psi\}$ it must hold that:

$$\mathfrak{M}, s \vDash \rho \text{ and } \underbrace{\mathfrak{M}, s \vDash \neg\rho}_{\Leftrightarrow \mathfrak{M}, s \nvDash \rho}$$

$\Longrightarrow$ there are no $\mathfrak{M}, s$ such that $\mathfrak{M}, s \vDash \Gamma \cup \{\neg\psi\}$ $\quad(\star)$

$\Longrightarrow$ for all $\mathfrak{M}, s$ with $\mathfrak{M}, s \vDash \Gamma$ holds: $\mathfrak{M}, s \vDash \psi$

Because: otherwise there would exist $\mathfrak{M}, s$ with $\mathfrak{M}, s \vDash \Gamma$ and $\underbrace{\mathfrak{M}, s \nvDash \psi}_{\mathfrak{M}, s \vDash \neg\psi}$

But this would contradict $(\star)$!

$\Longrightarrow \Gamma \vDash \psi \implies \Gamma\psi$ is correct. ∎

$\vee$-*Introduction in the antecedent:* ($\vee$-Ant.)

$$\frac{\begin{array}{c}\Gamma\varphi\ \rho\\ \Gamma\psi\ \rho\end{array}}{\underbrace{\Gamma(\varphi \vee \psi)}_{antecedent}\ \rho}$$

Explanation:

Disjunctions $\varphi \vee \psi$ in the antecedent allow for being treated in terms of two cases – case φ on the one hand and case ψ on the other.

(Here the sequents in question do *not* get shorter.)

Correctness: analogous to proof by cases!

$\vee$-*Introduction in the consequent:* ($\vee$-Con.)

(i) $\dfrac{\Gamma\varphi}{\Gamma(\varphi \vee \psi)}$ (ii) $\dfrac{\Gamma\psi}{\Gamma(\varphi \vee \psi)}$

Explanation:

We are always allowed to weaken consequents by introducing disjunctions.

Correctness:
Proof. (i) Assume that $\Gamma\varphi$ is correct, i.e. $\Gamma \vDash \varphi$.
$\Longrightarrow$ for all $\mathfrak{M}, s$ with $\mathfrak{M}, s \vDash \Gamma$ holds:

$$\underbrace{\mathfrak{M}, s \vDash \varphi}_{\Rightarrow \mathfrak{M}, s \vDash \varphi \vee \psi}$$

$\Longrightarrow\ \Gamma \vDash \varphi \vee \psi\ \Longrightarrow\ \Gamma(\varphi \vee \psi)$ is correct.
(ii) Analogously. ■

From the rules introduced so far we can "derive" further rules in the sense that these latter rules can be regarded as abbreviations of combined applications of the rules above:

– Excluded middle: (EM)

$$\frac{}{\varphi \vee \neg\varphi}$$

Derivation:

1. $\varphi\ \varphi$ (Ass.)
2. $\varphi\ \varphi \vee \neg\varphi$ ($\vee$-Con.(i)) with 1.
3. $\neg\varphi\ \neg\varphi$ (Ass.)
4. $\neg\varphi\ \varphi \vee \neg\varphi$ ($\vee$-Con.(ii)) with 3.
5. $\varphi \vee \neg\varphi$ (PC) with 2., 4.

■

– Triviality: (Triv.)

$$\begin{array}{l} \Gamma\varphi \\ \Gamma\neg\varphi \\ \hline \Gamma\psi \end{array}$$

Derivation:

1. $\Gamma\varphi$ (Premise)
2. $\Gamma\neg\varphi$ (Premise)
3. $\Gamma\neg\psi\varphi$ (Ant.) with 1.
4. $\Gamma\neg\psi\neg\varphi$ (Ant.) with 2.
5. $\Gamma\psi$ (CD) with 3.,4.

■

– Chain syllogism: (CS)

$$\begin{array}{l} \Gamma\varphi\psi \\ \Gamma\varphi \\ \hline \Gamma\psi \end{array}$$

Derivation:

1. $\Gamma\varphi\psi$ (Premise)
2. $\Gamma\varphi$ (Premise)
3. $\Gamma\neg\varphi\varphi$ (Ant.) with 2.
4. $\Gamma\neg\varphi\neg\varphi$ (Ass.)
5. $\Gamma\neg\varphi\psi$ (Triv.) with 3.,4.
6. $\Gamma\psi$ (PC) with 1.,5.

■

– <u>Contraposition:</u> (CP)

1. $\dfrac{\Gamma\varphi\psi}{\Gamma\neg\psi\neg\varphi}$ 2. $\dfrac{\Gamma\varphi\neg\psi}{\Gamma\psi\neg\varphi}$ 3. $\dfrac{\Gamma\neg\varphi\psi}{\Gamma\neg\psi\varphi}$ 4. $\dfrac{\Gamma\neg\varphi\neg\psi}{\Gamma\psi\varphi}$

<u>Derivation:</u>

1.
   1. $\Gamma\varphi\psi$ (Premise)
   2. $\Gamma\neg\psi\varphi\psi$ (Ant.) with 1.
   3. $\Gamma\neg\psi\varphi\neg\psi$ (Ass.)
   4. $\Gamma\neg\psi\varphi\neg\varphi$ (Triv.) with 2.,3.
   5. $\Gamma\neg\psi\neg\varphi\neg\varphi$ (Ass.)
   6. $\Gamma\neg\psi\neg\varphi$ (PC) with 4.,5.

2.-4. analogously ■

(Note that we could *not* have used CD in order to derive 6. directly from 2. and 3.: φ is not necessarily a negation formula as demanded by CD!)

– <u>Disjunctive syllogism:</u> (DS)

$$\begin{array}{c}\Gamma(\varphi \vee \psi)\\ \Gamma\neg\varphi\\ \hline \Gamma\psi\end{array}$$

Plan of derivation:
First derive $\Gamma \varphi \psi$ and $\Gamma \psi \psi$ on the basis of $\Gamma \neg \varphi$.
$\Longrightarrow \Gamma \varphi \vee \psi \;\; \psi$ will be derivable $\Longrightarrow$ using chain syllogism we will be done!

<u>Derivation:</u>

1. $\Gamma \varphi \vee \psi$ (Premise)
2. $\Gamma \neg \varphi$ (Premise)
3. $\Gamma \varphi \neg \varphi$ (Ant.) with 2.
4. $\Gamma \varphi \varphi$ (Ass.)
5. $\Gamma \varphi \psi$ (Triv.) with 4.,3.
6. $\Gamma \psi \psi$ (Ass.)
7. $\Gamma \varphi \vee \psi \; \psi$ ($\vee$-Ant.) with 5.,6.
8. $\Gamma \psi$ (CS) with 7.,1.

■

4.4 Rules for Quantifiers

(In the following, x and y are *arbitrary* variables again.)

<u>*$\exists$-Introduction in the consequent:*</u> ($\exists$-Con.)

$$\frac{\Gamma \; \varphi\frac{t}{x}}{\Gamma \; \exists x \varphi}$$

Explanation:

If we can conclude from Γ that t has the property expressed by the formula φ, then we are also allowed to conclude from Γ that there exists something which has the property expressed by φ.

Correctness:
Proof. Assume that $\Gamma \; \varphi\frac{t}{x}$ is correct, i.e., $\Gamma \vDash \varphi\frac{t}{x}$.
Let $\mathfrak{M}, s$ be arbitrary, such that $\mathfrak{M}, s \vDash \Gamma$.
$\Longrightarrow \mathfrak{M}, s \vDash \varphi\frac{t}{x}$ (by assumption)
$\Longrightarrow \mathfrak{M}, s\frac{Val_{\mathfrak{M},s}(t)}{x} \vDash \varphi$ (substitution lemma)
$\Longrightarrow$ there is a $d \in D$, such that $\mathfrak{M}, s\frac{d}{x} \vDash \varphi$ $(d = Val_{\mathfrak{M},s}(t))$

$\Longrightarrow \mathfrak{M}, s \models \exists x\varphi$
$\Longrightarrow \Gamma \models \exists x\varphi \Longrightarrow \Gamma\, \exists x\varphi$ is correct. ■

<u>*∃-Introduction in the antecedent:*</u> (∃-Ant.)

$$\frac{\Gamma\ \varphi\frac{y}{x}\ \psi}{\Gamma\ \exists x\varphi\ \psi} \qquad \text{(if } y \text{ is not free in the sequent } \Gamma\ \exists x\varphi\ \psi\text{)}$$

Explanation:

Assume that we can derive from Γ and from the fact that y has the property expressed by φ the conclusion that ψ is the case. Furthermore assume that Γ does not include any assumptions on y nor does ψ say anything about y. Then we are allowed to conclude ψ from Γ together with the assumption that there exists *some* x which has the property expressed by φ.
Compare the proof of "For all x there is a y, such that $y \circ x = e$" (theorem 1) in chapter 1: there we had

Proof.
• Let x be arbitrary (in G).
• By (G3) there is a y, such that $x \circ y = e$.
Now let y be some group element for which $x \circ y = e$:
• for this y it follows from (G3) that there is a z, such that ...
⋮
• $\Longrightarrow y \circ x = e$, which implies that there is a left-inverse element for x.
So we have shown for our arbitrarily chosen y for which $x \circ y = e$ holds that it is a left-inverse element of x and thus that there is a left-inverse element for x. But since y was chosen arbitrarily in the sense that we did not assume anything about y except that it satisfies $x \circ y = e$, the same conclusion can be drawn from the mere *existence* of a group element that satisfies $x \circ y = e$. ■

Now written in terms of sequents:

$$\frac{\overbrace{\text{Group axioms}}^{\Gamma}\quad \overbrace{x \circ y \equiv e}^{\varphi\frac{y}{y}}\quad \overbrace{\exists y\ y \circ x \equiv e}^{\psi}}{\underbrace{\text{Group axioms}}_{\Gamma}\quad \exists y\ \underbrace{x \circ y \equiv e}_{\varphi}\quad \underbrace{\exists y\ y \circ x \equiv e}_{\psi}}$$

The "if"-condition (y not free in $\Gamma\,\exists y\varphi\,\psi$) is satisfied and thus neither $\Gamma\,\exists y\varphi$ nor ψ contains any information about y.

Correctness:

Proof. Assume $\Gamma\,\varphi\frac{y}{x}\;\psi$ is correct,
i.e., $\Gamma \cup \{\varphi\frac{y}{x}\} \vDash \psi\;\;(\star)$ and y is not free in $\Gamma\,\exists x\varphi\;\psi$
Let $\mathfrak{M}, s$ be arbitrary with $\mathfrak{M}, s \vDash \Gamma \cup \{\exists x\varphi\}$ (to show: $\mathfrak{M}, s \vDash \psi$).
$\Longrightarrow \mathfrak{M}, s \vDash \exists x\varphi$
$\Longrightarrow$ there is a $d \in D$, such that $\mathfrak{M}, s\frac{d}{x} \vDash \varphi$
$\Longrightarrow$ there is a $d \in D$, such that $\mathfrak{M}, (s\frac{d}{y})\frac{d}{x} \vDash \varphi$

because:

for $x = y$: ✓
for $x \neq y$: by assumption, $y \notin free(\exists x\;\varphi)$
$\Longrightarrow y \notin free(\varphi)$ (since $y \neq x$)
Thus: $\mathfrak{M}, s\frac{d}{x} \vDash \varphi \iff \mathfrak{M}, (s\frac{d}{y})\frac{d}{x} \vDash \varphi$ ✓
(by the coincidence lemma, for $y \notin free(\varphi)$)

$\Longrightarrow$ there is a $d \in D$, such that $\mathfrak{M}, (s\frac{d}{y})\frac{\overbrace{Val_{\mathfrak{M}, s\frac{d}{y}}(y)}^{d}}{x} \vDash \varphi$
Hence, by the substitution lemma: $\mathfrak{M}, s\frac{d}{y} \vDash \varphi\frac{y}{x}$

Furthermore: by assumption $\mathfrak{M}, s \vDash \Gamma$
$\Longrightarrow \mathfrak{M}, s\frac{d}{y} \vDash \Gamma$ (from the coincidence lemma – $y \notin free(\Gamma)$)

Thus, for d as above:
$\mathfrak{M}, s\frac{d}{y} \vDash \Gamma \cup \{\varphi\frac{y}{x}\}$
$\Longrightarrow \mathfrak{M}, s\frac{d}{y} \vDash \psi$ (by $(\star)$)
$\Longrightarrow \mathfrak{M}, s \vDash \psi$ (coincidence lemma, for $y \notin free(\psi)$)
$\Longrightarrow \Gamma \cup \{\exists x\varphi\} \vDash \psi \Longrightarrow \Gamma\,\exists x\varphi\;\psi$ is correct. ■

Remark 14 *Postulating the "if" condition is* necessary*:*

$$\frac{[P(x,y)]\frac{y}{x} \qquad P(y,y)}{\exists x\,\underbrace{P(x,y)}_{\varphi} \qquad \underbrace{P(y,y)}_{\psi}}$$

(correct by Ass. rule)
(incorrect – e.g. $<$-relation on $\mathbb{R}$)

Not *an instance of our rule: y is free in $\exists x P(x,y)\;\; P(y,y)$!*

4.5 Rules for Equality

Reflexivity: (Ref.)

$$\overline{\quad t \equiv t \quad}$$

Explanation:

The equality relation on D is reflexive (independent of the model).

Correctness:
Proof.
For all $\mathfrak{M}, s$ holds:
$Val_{\mathfrak{M},s}(t) = Val_{\mathfrak{M},s}(t)$
$\Longrightarrow \mathfrak{M}, s \vDash t \equiv t$ ■

Substitution rule: (Sub.)

$$\frac{\Gamma \ \varphi\frac{t}{x}}{\Gamma \ t \equiv t' \ \varphi\frac{t'}{x}}$$

Explanation:

Substitution of identicals!

Correctness:
Proof. Assume $\Gamma \varphi\frac{t}{x}$ is correct, i.e., $\Gamma \vDash \varphi\frac{t}{x}$.
Let $\mathfrak{M}, s$ be arbitrary with $\mathfrak{M}, s \vDash \Gamma \cup \{t \equiv t'\}$ (to show: $\mathfrak{M}, s \vDash \varphi\frac{t'}{x}$)
$\Longrightarrow \mathfrak{M}, s \vDash \varphi\frac{t}{x}$ (from the assumption)
$\Longrightarrow \mathfrak{M}, s\frac{Val_{\mathfrak{M},s}(t)}{x} \vDash \varphi$ (substitution lemma)
Since $\mathfrak{M}, s \vDash t \equiv t' \implies Val_{\mathfrak{M},s}(t) = Val_{\mathfrak{M},s}(t')$
and therefore also: $\mathfrak{M}, s\frac{Val_{\mathfrak{M},s}(t')}{x} \vDash \varphi$
$\Longrightarrow \mathfrak{M}, s \vDash \varphi\frac{t'}{x}$ (substitution lemma)
$\Longrightarrow \Gamma \cup \{t \equiv t'\} \vDash \varphi\frac{t'}{x} \implies \Gamma \ t \equiv t' \ \varphi\frac{t'}{x}$ is correct ■

From the rules in sections 4.4 and 4.5 we can derive the following auxiliary rules:

– $$\frac{\Gamma\ \varphi}{\Gamma\ \exists x\varphi}$$

by (∃-Con.) $\quad (t := x;\ \varphi\frac{x}{x} = \varphi)$

– $$\frac{\Gamma\ \varphi\ \psi}{\Gamma\ \exists x\varphi\ \psi}$$ (if x is not free in $\Gamma\ \psi$)

by (∃-Ant.) $\quad$ ($y := x$; note that x is of course bound in $\exists x\varphi$)

– $$\frac{\Gamma\ \varphi}{\Gamma\ x \equiv t'\ \varphi\frac{t'}{x}}$$

by (Sub.) $\quad (t := x)$

– <u>Symmetry:</u> (Symm.)

$$\frac{\Gamma\ t_1 \equiv t_2}{\Gamma\ t_2 \equiv t_1}$$

Since:

1. $\Gamma\ t_1 \equiv t_2$ (Premise)
2. $t_1 \equiv t_1$ (Ref.)
3. $\Gamma\ \underbrace{t_1 \equiv t_1}_{=\varphi\frac{t_1}{x}}$ (Ant.) with 2.
4. $\Gamma\ t_1 \equiv t_2\ \underbrace{t_2 \equiv t_1}_{=\varphi\frac{t_2}{x}}$ (Sub.) with 3. for $\varphi =\ x \equiv t_1 \quad \Longrightarrow \varphi\frac{t_1}{x} =\ t_1 \equiv t_1,\ \varphi\frac{t_2}{x} =\ t_2 \equiv t_1$
5. $\Gamma\ t_2 \equiv t_1$ (CS) with 4.,1.

■

– Transitivity: (Trans.)

$$\frac{\begin{array}{l}\Gamma\ t_1 \equiv t_2 \\ \Gamma\ t_2 \equiv t_3\end{array}}{\Gamma\ t_1 \equiv t_3}$$

Since:

1. $\Gamma\ t_1 \equiv t_2$ (Premise)
2. $\Gamma\ t_2 \equiv t_3$ (Premise)
3. $\Gamma\ t_2 \equiv t_3\ t_1 \equiv t_3$ (Sub.) with 1. for $\varphi = \ t_1 \equiv x \quad \Longrightarrow \varphi\frac{t_2}{x} = \ t_1 \equiv t_2,\ \varphi\frac{t_3}{x} = \ t_1 \equiv t_3$
4. $\Gamma\ t_1 \equiv t_3$ (CS) with 3.,2.

■

–
$$\frac{\begin{array}{l}\Gamma\ P(t_1,\ldots,t_n) \\ \Gamma\ t_1 \equiv t'_1 \\ \quad\vdots \\ \Gamma\ t_n \equiv t'_n\end{array}}{\Gamma\ P(t'_1,\ldots,t'_n)}$$

Since: (e.g. for $n = 2$)

1. $\Gamma\ P(t_1, t_2)$ (Premise)
2. $\Gamma\ t_1 \equiv t'_1$ (Premise)
3. $\Gamma\ t_2 \equiv t'_2$ (Premise)
4. $\Gamma\ t_1 \equiv t'_1\ P(t'_1, t_2)$ (Sub.) with 1. for $\varphi = P(x, t_2)$
5. $\Gamma\ P(t'_1, t_2)$ (CS) with 4.,2.
6. $\Gamma\ t_2 \equiv t'_2\ P(t'_1, t'_2)$ (Sub.) with 5. for $\varphi = P(t'_1, x)$
7. $\Gamma\ P(t'_1, t'_2)$ (CS) with 6.,3.

■

Analogously:

$$\frac{\begin{array}{l}\Gamma\ t_1 \equiv t_1' \\ \quad\vdots \\ \Gamma\ t_n \equiv t_n'\end{array}}{\Gamma\ f(t_1, \ldots, t_n) \equiv f(t_1', \ldots, t_n')}$$

For the proof: by Reflexivity $\frac{}{f(t_1,\ldots,t_n)\equiv f(t_1,\ldots,t_n)}$, then add Γ by (Ant.) $\Rightarrow$ Subst. $\Rightarrow$ CS (repeat last two steps n times)

4.6 The Soundness Theorem for the Sequent Calculus

It follows:

Theorem 3 ***(Soundness Theorem)***
For all $\Phi \subseteq \mathcal{F}_\mathcal{S}$, for all $\varphi \in \mathcal{F}_\mathcal{S}$, it holds:

If $\Phi \vdash \varphi$, then $\Phi \vDash \varphi$.

Proof.
Assume $\Phi \vdash \varphi$, i.e., there are $\varphi_1, \ldots, \varphi_n \in \Phi$, such that the sequent $\varphi_1 \ldots \varphi_n \varphi$ is derivable in the sequent calculus.

I.e.
1. ...
2. ...
⋮
m. $\varphi_1 \ldots \varphi_n \varphi$

We have already shown: each rule of the sequent calculus is correct (premise-free rules lead to correct sequents, rules with premises preserve correctness). By induction over the length of derivations in the sequent calculus it follows: Every sequent that is derivable in the sequent calculus is correct.

So this must hold also for $\varphi_1 \ldots \varphi_n \varphi$, thus $\underbrace{\{\varphi_1, \ldots, \varphi_n\}}_{\subseteq \Phi} \vDash \varphi$

$\Rightarrow \Phi \vDash \varphi$. ■

Remark 15 *The sequent calculus does not only contain correct rules for* $\neg$, $\vee$, $\exists$, $\equiv$, *but also for* $\wedge$, $\rightarrow$, $\leftrightarrow$, $\forall$ *by means of the metalinguistic abbreviations that we considered in the chapter on semantics.*
E.g. $\varphi \rightarrow \psi := \neg\varphi \vee \psi$

Using such abbreviations we get:

Modus Ponens:

$$\frac{\begin{array}{l}\Gamma\ \overbrace{\varphi \rightarrow \psi}^{\neg\varphi\vee\psi} \\ \Gamma\ \varphi\end{array}}{\Gamma\ \psi} \qquad \text{proof analogous to:} \qquad \frac{\begin{array}{l}\Gamma\ \varphi \vee \psi \\ \Gamma\ \neg\varphi\end{array}}{\Gamma\ \psi}$$

4.7 Some Important Proof-Theoretic Concepts

(Compare section 3.3!)

We have already defined the notion of *derivability* for formulas ("$\Phi \vdash \varphi$").

Some formulas have the property of being *derivable without any premises*:

Definition 15 *For all* $\varphi \in \mathcal{F}_{\mathcal{S}}$*:*
φ *ist provable :iff the (one-element) sequent* φ *is derivable in the sequent calculus (briefly:* $\vdash \varphi$*).*

E.g., we know that $\varphi \vee \neg\varphi$ (for arbitrary $\varphi \in \mathcal{F}_{\mathcal{S}}$) is provable: see p. 57.

Some formulas have the property of *not including (explicitly or implicitly) a contradiction*:

Definition 16 *For all* $\varphi \in \mathcal{F}_{\mathcal{S}}$, $\Phi \subseteq \mathcal{F}_{\mathcal{S}}$*:*
φ *is consistent :iff there is no* $\psi \in \mathcal{F}_{\mathcal{S}}$ *with:* $\{\varphi\} \vdash \psi,\ \{\varphi\} \vdash \neg\psi$.
Φ *is consistent :iff there is no* $\psi \in \mathcal{F}_{\mathcal{S}}$ *with:* $\Phi \vdash \psi,\ \Phi \vdash \neg\psi$.

E.g. $P(c)$ is consistent, because:
assume $\{P(c)\} \vdash \psi,\ \{P(c)\} \vdash \neg\psi$
$\Longrightarrow\ \{P(c)\} \vDash \psi,\ \{P(c)\} \vDash \neg\psi$ (Soundness theorem)
$\Longrightarrow$ there are no $\mathfrak{M}, s$ with $\mathfrak{M}, s \vDash P(c)$
But this is false: take e.g. $D = \{1\},\ \mathfrak{I}(c) = 1,\ \mathfrak{I}(P) = D$
$\Rightarrow\ \mathfrak{M} \vDash P(c)$.

Lemma 8 *Φ is consistent iff there is a $\psi \in \mathcal{F}_\mathcal{S}$, such that $\Phi \nvdash \psi$.*

Proof. We show:

$$\begin{aligned}\Phi \text{ is not consistent} &\iff \text{there is no } \psi \in \mathcal{F}_\mathcal{S}, \text{ such that } \Phi \nvdash \psi \\ &\iff \text{for all } \psi \in \mathcal{F}_\mathcal{S}: \ \Phi \vdash \psi\end{aligned}$$

"$\Leftarrow$:" since $\Phi \vdash \psi,\ \Phi \vdash \neg\psi \Longrightarrow \Phi$ is not consistent ✓

"$\Rightarrow$:" assume Φ is not consistent,
i.e., $\Phi \vdash \varphi,\ \Phi \vdash \neg\varphi$ for some $\varphi \in \mathcal{F}_\mathcal{S}$:

Thus

$$\underbrace{\begin{array}{ll} 1. & \ldots \\ 2. & \ldots \\ \vdots & \vdots \\ \text{m.} & \Gamma\varphi \end{array}}_{\text{derivation 1 } (\Gamma \subseteq \Phi)} \qquad \underbrace{\begin{array}{ll} 1. & \ldots \\ 2. & \ldots \\ \vdots & \vdots \\ \text{n.} & \Gamma'\neg\varphi \end{array}}_{\text{derivation 2 } (\Gamma' \subseteq \Phi)}$$

Now:

$$\begin{array}{lll} 1. & \ldots & \\ \vdots & \vdots & \Big\} \text{ derivation 1} \\ m. & \Gamma\varphi & \\ m+1. & \ldots & \\ \vdots & \vdots & \Big\} \text{ derivation 2} \\ m+n. & \Gamma'\neg\varphi & \\ m+n+1. & \Gamma\Gamma'\varphi & \text{(Ant.) with m.} \\ m+n+2. & \Gamma\Gamma'\neg\varphi & \text{(Ant.) with m+n.} \\ m+n+3. & \Gamma\Gamma'\psi & \text{(Triv.) with m+n+1., m+n+2.} \end{array}$$

Since $\Gamma \cup \Gamma' \subseteq \Phi \Longrightarrow \ \Phi \vdash \psi$ for arbitrary $\psi \in \mathcal{F}_\mathcal{S}$ ✓ ■

Moreover:

Lemma 9 *Φ is consistent iff every finite subset $\Phi' \subseteq \Phi$ is consistent.*

Proof. Immediate from our definitions. ■

The soundness theorem above was formulated for $\vdash$ und $\vDash$. But there is a corresponding version for *consistency* and *satisfiabillity* (in fact we already sketched the proof of this second version when we showed that $\{P(c)\}$ is consistent on p. 66):

Corollary 2 ***(Soundness Theorem: Second Version)***
For all $\Phi \subseteq \mathcal{F}_{\mathcal{S}}$: if Φ is satisfiable, then Φ is consistent.

Proof.
We show: Φ is not consistent $\Longrightarrow$ Φ is not satisfiable.

Assume that Φ is not consistent:
$\Longrightarrow$ $\Phi \vdash \psi$, $\Phi \vdash \neg\psi$ (by definition)
$\Longrightarrow$ $\Phi \vDash \psi$, $\Phi \vDash \neg\psi$ (Soundness theorem)
$\Longrightarrow$ there are no $\mathfrak{M}, s$ with $\mathfrak{M}, s \vDash \Phi$,
i.e., Φ is not satisfiable. ■

Furthermore, we can show the proof-theoretic counterpart of lemma 5:

Lemma 10 *For all $\Phi \subseteq \mathcal{F}_{\mathcal{S}}, \varphi \in \mathcal{F}_{\mathcal{S}}$:*

1. *φ is provable iff $\varnothing \vdash \varphi$.*
2. *$\Phi \vdash \varphi$ iff $\Phi \cup \{\neg\varphi\}$ is not consistent.*
3. *φ is provable iff $\neg\varphi$ is not consistent.*

Proof.

1. Follows directly from the definitions. ✓
2. ($\Rightarrow$) Assume that $\Phi \vdash \varphi$:
   Obviously, this implies: $\Phi \cup \{\neg\varphi\} \vdash \varphi$.

   Furthermore: $\Phi \cup \{\neg\varphi\} \vdash \neg\varphi$ (by the Ass. rule)
   $\Longrightarrow$ $\Phi \cup \{\neg\varphi\}$ is not consistent. ✓

   ($\Leftarrow$) Assume that $\Phi \cup \{\neg\varphi\}$ is not consistent:
   $\Longrightarrow$ every formula is derivable from $\Phi \cup \{\neg\varphi\}$ (Lemma 8)
   $\Longrightarrow$ $\Phi \cup \{\neg\varphi\} \vdash \varphi$,
   i.e., there is a derivation of the following form:

	1.	$\ldots$	
	$\vdots$	$\vdots$	
	m.	$\Gamma \neg\varphi\ \varphi$	$(\Gamma \subseteq \Phi)$
Extend:	m+1.	$\Gamma \varphi\ \varphi$	(Ass.)
	m+2.	$\Gamma \varphi$	(PC) with m+1., m.

$\Longrightarrow\ \Gamma \vdash \varphi$

$\Longrightarrow\ \Phi \vdash \varphi$ ✓

3. Consider $\Phi = \varnothing$ and apply 2. and 1. ✓

■

We did not prove the semantic counterpart of the following lemma (since it would be trivial):

Lemma 11 *For all $\Phi \subseteq \mathcal{F}_{\mathcal{S}}, \varphi \in \mathcal{F}_{\mathcal{S}}$:*
If Φ is consistent, then $\Phi \cup \{\varphi\}$ is consistent or $\Phi \cup \{\neg\varphi\}$ is consistent.

Proof.
Assume that $\Phi \cup \{\varphi\},\ \Phi \cup \{\neg\varphi\}$ are both not consistent.
$\Phi \cup \{\neg\varphi\}$ is not consistent $\Longrightarrow\ \Phi \vdash \varphi$ (by 2. of lemma 10)
$\Phi \cup \{\varphi\}$ is not consistent $\Longrightarrow\ \Phi \vdash \neg\varphi$ (proof analogous to 2. of lemma 10)
$\Longrightarrow\ \Phi$ is not consistent. ■

Remark 16

- *We do not consider a proof-theoretic counterpart of the semantic notion of* logical equivalence.
- *Our proof-theoretic concepts are actually again relativized to a symbol set $\mathcal{S}$ ($\mathcal{S}$-*sequent calculus*, $\mathcal{S}$-*derivability*,...), just as our semantic concepts.*

Here is a lemma where we make this implicit reference to symbol sets explicit (we need this lemma later in the proof of the completeness theorem):

Lemma 12 *Let $\mathcal{S}_0 \subseteq \mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \ldots$ be a chain of symbol sets.*
Let $\Phi_0 \subseteq \Phi_1 \subseteq \Phi_2 \subseteq \ldots$ be a chain of sets of formulas, such that:
For all $n \in \{0, 1, 2, \ldots\}$: Φ_n is a set of formulas over the symbol set $\mathcal{S}_n$ and Φ_n is $\mathcal{S}_n$-consistent (i.e., consistent in the sequent calculus for formulas in $\mathcal{F}_{\mathcal{S}_n}$).
Let finally $\mathcal{S} = \bigcup_{n \in \{0,1,2,\ldots\}} \mathcal{S}_n$, $\Phi = \bigcup_{n \in \{0,1,2,\ldots\}} \Phi_n$.
It follows: Φ is $\mathcal{S}$-consistent.

Proof.
Let the symbol sets and formula sets be given as indicated above.
Assume that Φ is not $\mathcal{S}$-consistent:
$\Longrightarrow$ there is a $\Psi \subseteq \Phi$ with Ψ finite, Ψ not $\mathcal{S}$-consistent (by lemma 9)
$\Longrightarrow$ there is a $k \in \{0, 1, 2, \ldots\}$, such that $\Psi \subseteq \Phi_k$ (since Ψ is finite).

Ψ is not $\mathcal{S}$-consistent, therefore for some $\psi \in \mathcal{F}_{\mathcal{S}}$: (i) there is an $\mathcal{S}$-derivation of ψ from Ψ, (ii) there is an $\mathcal{S}$-derivation of $\neg\psi$ from Ψ.

But in these two sequent calculus derivations only finitely many symbols in $\mathcal{S}$ can occur. Thus there is an $m \in \{0, 1, 2, \ldots\}$, such that $\mathcal{S}_m$ contains all the symbols in these two derivations.
Without restricting generality, we can assume that $m \geq k$:
$\Longrightarrow$ Ψ is not $\mathcal{S}_m$-consistent
$\Longrightarrow$ Φ_k is not $\mathcal{S}_m$-consistent
Since $m \geq k$ it also follows that $\Phi_k \subseteq \Phi_m$
$\Longrightarrow$ Φ_m is not $\mathcal{S}_m$-consistent: contradiction!

$\Longrightarrow$ Φ is $\mathcal{S}$-consistent. ■

4.8 Problem Set 5

1. (This problem counts for award of CREDIT POINTS.)

   Are the following rules correct?

   (a) $$\begin{array}{lll} \Gamma & \varphi_1 & \psi_1 \\ \Gamma & \varphi_2 & \psi_2 \\ \hline \Gamma & \varphi_1 \lor \varphi_2 & \psi_1 \lor \psi_2 \end{array}$$

   (b) $$\begin{array}{lll} \Gamma & \varphi_1 & \psi_1 \\ \Gamma & \varphi_2 & \psi_2 \\ \hline \Gamma & \varphi_1 \lor \varphi_2 & \psi_1 \land \psi_2 \end{array}$$

2. Derive the following (auxiliary) rules from the rules of the sequent calculus:

   (a) $$\begin{array}{ll} \Gamma & \varphi \\ \hline \Gamma & \neg\neg\varphi \end{array}$$

   (b) $$\begin{array}{ll} \Gamma & \neg\neg\varphi \\ \hline \Gamma & \varphi \end{array}$$

   (c) $$\begin{array}{ll} \Gamma & \varphi \\ \Gamma & \psi \\ \hline \Gamma & \varphi \land \psi \end{array}$$

   (d) $$\begin{array}{lll} \Gamma & \varphi & \psi \\ \hline \Gamma & & \varphi \to \psi \end{array}$$

   (e) $$\begin{array}{ll} \Gamma & \varphi \land \psi \\ \hline \Gamma & \varphi \end{array}$$

   (f) $$\begin{array}{ll} \Gamma & \varphi \land \psi \\ \hline \Gamma & \psi \end{array}$$

3. Are the following rules correct?

   (a) $$\begin{array}{ll} \varphi & \psi \\ \hline \exists x\varphi & \exists x\psi \end{array}$$

   (b) $$\begin{array}{lll} \Gamma & \varphi & \psi \\ \hline \Gamma & \forall x\varphi & \exists x\psi \end{array}$$

4. Derive the following (auxiliary) rules from the rules of the sequent calculus:

(a) $\dfrac{\Gamma \quad \forall x\varphi}{\Gamma \quad \varphi\frac{t}{x}}$

(b) $\dfrac{\Gamma \quad \forall x\varphi}{\Gamma \quad \varphi}$

(c) $\dfrac{\Gamma \quad \varphi\frac{t}{x} \quad \psi}{\Gamma \quad \forall x\varphi \quad \psi}$

(d) $\dfrac{\Gamma \quad \varphi\frac{y}{x}}{\Gamma \quad \forall x\varphi}$ if y is not free in the sequent $\Gamma\ \forall x\varphi$.

(e) $\dfrac{\Gamma \quad \varphi \quad \psi}{\Gamma \quad \forall x\varphi \quad \psi}$

(f) $\dfrac{\Gamma \quad \varphi}{\Gamma \quad \forall x\varphi}$ if x is not free in the sequent Γ.

(Remark: $\forall x\varphi$ is to be regarded as $\neg\exists x\neg\varphi$ in the problems above.)

5 The Completeness Theorem and Its Consequences

Here is the idea of the proof:

Completeness theorem: For all Φ, φ: if $\Phi \vDash \varphi$ then $\Phi \vdash \varphi$.

Let Φ, φ be such that $\Phi \vDash \varphi$. Assume $\Phi \nvdash \varphi$:
$\Longrightarrow \Phi \cup \{\neg\varphi\}$ is consistent (by 2. of lemma 10)
$\Longrightarrow \Phi \cup \{\neg\varphi\}$ satisfiable (from ($\star$) below)

But: $\Phi \cup \{\neg\varphi\}$ is not satisfiable by 2. of lemma 5 (since $\Phi \vDash \varphi$)
Contradiction!
$\Longrightarrow \Phi \vdash \varphi$

So we have to show: For all Φ: if Φ is consistent, then Φ is satisfiable. ($\star$)

This means that for consistent Φ we must prove that $\mathfrak{M}, s$ exist, such that: $\mathfrak{M}, s \vDash \Phi$.

We divide this existence proof into two parts:

5.1 The Satisfiability of Maximally Consistent Sets of Formulas with Instances

5.2 The Extension of Consistent Sets of Formulas to Maximally Consistent Sets of Formulas with Instances

By (i) applying 5.2, then (ii) applying 5.1 – which is going to give us ($\star$) – and finally (iii) applying the proof strategy from above, we will be done.

5.1 The Satisfiability of Maximally Consistent Sets of Formulas with Instances

Let $\mathcal{S}$ be a symbol set that we keep fixed.

Definition 17 *Let* $\Phi \subseteq \mathcal{F}_{\mathcal{S}}$*:*

- Φ *is maximally consistent iff*

  Φ *is consistent and for all* $\varphi \in \mathcal{F}_{\mathcal{S}}$: $\Phi \vdash \varphi$ *or* $\Phi \vdash \neg\varphi$

- Φ *contains instances iff*

  for every formula of the form $\exists x \varphi \in \mathcal{F}_{\mathcal{S}}$ *there is a* $t \in \mathcal{T}_{\mathcal{S}}$*, such that:* $\Phi \vdash (\exists x \varphi \to \varphi \frac{t}{x})$

Now let Φ be maximally consistent with instances (i.e., it is maximally consistent and contains instances).
To show: Φ is satisfiable.

Let us consider an example first: let $\Phi = \{P(c_1)\}$.
$\Longrightarrow \Phi$ is consistent (since it is satisfiable – apply corollary 2).
But: Φ is not *maximally* consistent.

E.g., $\Phi \nvdash P(c_2)$,

$\Phi \nvdash \neg P(c_2)$

because $\Phi \cup \{\neg P(c_2)\}$, $\Phi \cup \{P(c_2)\}$ are consistent since they are satisfiable.

Furthermore: Φ does not contain instances for all formulas (only for *some*).

E.g., $\Phi \nvdash (\exists x \neg P(x) \to \underbrace{\neg P(t)}_{\neg P(x)\frac{t}{x}})$ for arbitrary $t \in \mathcal{T}_{\mathcal{S}}$

Since: choose any model of $\Phi \cup \{\underbrace{\neg(\underbrace{\exists\, x\, \neg P(x)}_{\text{true}} \to \underbrace{\neg P(t)}_{\text{false}})}_{\text{false}}\}$; such a model exists and and thus this formula set is consistent.

But e.g. $\Phi \vdash (\exists x P(x) \to P(c_1))$, since $\Phi \vdash P(c_1)$ and thus by the $\vee$-introduction rule in the consequent: $\Phi \vdash \underbrace{\neg \exists x P(x) \vee P(c_1)}_{\exists x P(x) \to P(c_1)}$

Another example:

Let $\mathfrak{M}, s$ be such that for every $d \in D$ there is a $t \in \mathcal{T}_{\mathcal{S}}$ such that: $Val_{\mathfrak{M},s}(t) = d$ (which implies that D is countable).

Consider $\Phi = \{\varphi \in \mathcal{F}_{\mathcal{S}} | \mathfrak{M}, s \vDash \varphi\}$
$\implies$ Φ is consistent and $\Phi \vdash \varphi$ or $\Phi \vdash \neg\varphi$ for arbitrary $\varphi \in \mathcal{F}_{\mathcal{S}}$.
(Since: $\mathfrak{M}, s \vDash \varphi$ or $\mathfrak{M}, s \vDash \neg\varphi \implies \varphi \in \Phi$ or $\neg\varphi \in \Phi$)

$\implies \Phi$ is maximally consistent!

Furthermore:
For all formulas $\exists x \varphi$ there is a t, such that: $\Phi \vdash (\exists x \varphi \to \varphi \frac{t}{x})$

Because:

Case 1: $\mathfrak{M}, s \nvDash \exists x \varphi$

$$\Rightarrow \underbrace{\mathfrak{M}, s \vDash \exists x \varphi \to \varphi \frac{t}{x}}_{\exists x \varphi \to \varphi \frac{t}{x} \ \in \Phi \quad \checkmark} \quad \text{for all } t \in \mathcal{T}_{\mathcal{S}}$$

Case 2: $\mathfrak{M}, s \vDash \exists x \varphi$
$\implies$ there is a $d \in D$, such that: $\mathfrak{M}, s\frac{d}{x} \vDash \varphi$, and
$d = Val_{\mathfrak{M},s}(t)$ for some $t \in \mathcal{T}_s$ (by assumption)
$\implies$ there is a $t \in \mathcal{T}_s$, such that: $\mathfrak{M}, s\frac{Val_{\mathfrak{M},s}(t)}{x} \vDash \varphi$
$\implies$ there is a $t \in \mathcal{T}_s$, such that: $\mathfrak{M}, s \vDash \varphi\frac{t}{x}$ (substitution lemma)
$\implies$ there is a $t \in \mathcal{T}_s$, such that: $\mathfrak{M}, s \vDash \exists x \varphi \to \varphi \frac{t}{x}$
For such a t it follows: $\exists x \varphi \to \varphi \frac{t}{x} \ \in \Phi \quad \checkmark$

So Φ is actually maximally consistent with instances.

Now we that we have seen an example of a maximally consistent set of formulas that contains instances, let us consider such sets *in general.*
We will show that every such set is satisfiable.

So let Φ be a maximally consistent formula set with instances.

Wanted: $\mathfrak{M}, s$ with $\mathfrak{M}, s \models \Phi$
Thus, if e.g. $P(f(x)) \in \Phi$
$\Longrightarrow \mathfrak{M}, s \models P(f(x)) \Leftrightarrow \mathfrak{I}(f)(s(x)) \in \mathfrak{I}(P)$

Problems:
1. What should we choose to be the members of the domain D?
2. How shall we define $\mathfrak{I}, s$?

Suggestions:

- Let us define D to be $\mathcal{T}_{\mathcal{S}}$! (This is about problem 1.)
- Every term is assumed to "denote" itself! (This is about problem 2.)
- $\mathfrak{I}(P)$ is defined in accordance with Φ (This is also about problem 2.)

I.e., *first attempt:* let

$D := \mathcal{T}_{\mathcal{S}}$
$\mathfrak{I}(c) := c$
$\mathfrak{I}(f)(t_1, \ldots, t_n) := f(t_1, \ldots, t_n)$
$\mathfrak{I}(P) := \{(t_1, \ldots, t_n) | \Phi \vdash P(t_1, \ldots, t_n)\}$ (so $\mathfrak{I}(P) = \mathfrak{I}^{\Phi}(P)$, i.e., $\mathfrak{I}$ depends on Φ)
$s(x) := x$

$\Rightarrow$ Assume $P(f(x)) \in \Phi$
$\Rightarrow \Phi \vdash P(f(x))$
$\Rightarrow \underbrace{f(x)}_{=\mathfrak{I}(f)(s(x))} \in \mathfrak{I}(P)$ (by definition of $\mathfrak{I}(P)$, $\mathfrak{I}(f)$, and $s(x)$)
$\Rightarrow \mathfrak{M}, s \models P(f(x))$ ✓

But there is yet another problem:

assume $f(x) \equiv f(y) \in \Phi$ (for $x \neq y$)
$\Rightarrow Val_{\mathfrak{M},s}(f(x)) = \mathfrak{I}(f)(\underbrace{s(x)}_{=x}) = f(x)$ (for $\mathfrak{M}, s$ defined as above)
$Val_{\mathfrak{M},s}(f(y)) = \mathfrak{I}(f)(\underbrace{s(y)}_{=y}) = f(y)$
BUT: $f(x) \neq f(y)$!
$\Rightarrow \mathfrak{M}, s \not\models f(x) \equiv f(y)$, although $f(x) \equiv f(y) \in \Phi$!

Improved attempt:
We must "adapt" $\mathfrak{M}, s$ to $\equiv$.
I.e., let us consider *equivalence classes of terms* rather than terms themselves as members of our intended domain: for $t_1, t_2 \in \mathcal{T}_{\mathcal{S}}$, let

$t_1 \sim t_2 \Leftrightarrow \Phi \vdash t_1 \equiv t_2$ (i.e., $\sim\, = \,\sim^{\Phi}$)

It follows:

Lemma 13

1. *$\sim$ is an equivalence relation on $\mathcal{T}_{\mathcal{S}}$.*
2. *$\sim$ is compatible with functions signs and predicates in $\mathcal{S}$, i.e.,*
   - $t_1 \sim t_1', \ldots, t_n \sim t_n' \Rightarrow f(t_1, \ldots, t_n) \sim f(t_1', \ldots, t_n')$
   - $t_1 \sim t_1', \ldots, t_n \sim t_n' \Rightarrow \Phi \vdash P(t_1, \ldots, t_n) \Leftrightarrow \Phi \vdash P(t_1', \ldots, t_n')$

Proof.

1. - $t \sim t$, since $\Phi \vdash t \equiv t$ (Refl.) ✓
   - Assume $t_1 \sim t_2 \implies \Phi \vdash t_1 \equiv t_2$
     $\Rightarrow \Phi \vdash t_2 \equiv t_1$ (Symm.) $\Rightarrow t_2 \sim t_1$ ✓
   - Assume $t_1 \sim t_2,\ t_2 \sim t_3 \implies \Phi \vdash t_1 \equiv t_2,\ \Phi \vdash t_2 \equiv t_3$
     $\Rightarrow \Phi \vdash t_1 \equiv t_3$ (Trans.) $\Rightarrow t_1 \sim t_3$ ✓ ✓
2. - Assume $t_1 \sim t_1', \ldots, t_n \sim t_n' \implies \Phi \vdash t_1 \equiv t_1', \ldots, \Phi \vdash t_n \equiv t_n'$
     $\Rightarrow \Phi \vdash f(t_1, \ldots, t_n) \equiv f(t_1', \ldots, t_n')$ (see p.65)
     $\Rightarrow f(t_1, \ldots, t_n) \sim f(t_1', \ldots, t_n')$ ✓
   - Assume $t_1 \sim t_1', \ldots, t_n \sim t_n'$
     $\vdots$
     analogously for $P(t_1, \ldots, t_n)$ and $P(t_1', \ldots, t_n')$ (see p.64) ✓ ✓

■

So we define:

$D^\Phi := \{[t]_\sim | t \in \mathcal{T}_\mathcal{S}\}$
$\mathfrak{I}^\Phi(c) := [c]_\sim$
$\mathfrak{I}^\Phi(f)\left([t_1]_\sim, \ldots, [t_n]_\sim\right) := [f(t_1, \ldots, t_n)]_\sim$
$\mathfrak{I}^\Phi(P) := \{([t_1]_\sim, \ldots, [t_n]_\sim) \,|\, \Phi \vdash P(t_1, \ldots, t_n)\}$
$\mathfrak{M}^\Phi := (D^\Phi, \mathfrak{I}^\Phi)$

$s^\Phi(x) := [x]_\sim$

(this is *well-defined*, since the definitions are independent of the choice of the representatives $t_1, \ldots, t_n$ as can be seen from 2. of lemma 13).

One can show:

Lemma 14 *For all $\Phi \subseteq \mathcal{F}_\mathcal{S}$:*

1. *For all $t \in \mathcal{T}_\mathcal{S} : Val_{\mathfrak{M}^\Phi, s^\Phi}(t) = [t]_\sim$*
2. *For all atomic formulas $\varphi \in \mathcal{F}_\mathcal{S}$:*
   $\mathfrak{M}^\Phi, s^\Phi \vDash \varphi$ *iff* $\Phi \vdash \varphi$
3. *For all $\varphi \in \mathcal{F}_\mathcal{S}$, for all pairwise distinct variables $x_1, \ldots, x_n$:*
   $\mathfrak{M}^\Phi, s^\Phi \vDash \exists x_1 \exists x_2 \ldots \exists x_n \, \varphi$ *iff*
   there are terms $t_1, \ldots, t_n \in \mathcal{T}_\mathcal{S}$ such that $\mathfrak{M}^\Phi, s^\Phi \vDash \varphi\frac{t_1, \ldots, t_n}{x_1, \ldots, x_n}$.

(Note that our assumption that Φ is maximally consistent and contains instances is not yet needed in order to derive the claims of this lemma.)

Proof.

1. By induction over $\mathcal{S}$-terms:
   - $t = c:\ Val_{\mathfrak{M}^\Phi, s^\Phi}(c) = \mathfrak{I}^\Phi(c) = [c]_\sim$ ✓ (def.)
   - $t = x:\ Val_{\mathfrak{M}^\Phi, s^\Phi}(x) = s^\Phi(x) = [x]_\sim$ ✓ (def.)
   - $t = f(t_1, \ldots, t_n)$
     $\Rightarrow Val_{\mathfrak{M}^\Phi, s^\Phi}(f(t_1, \ldots, t_n)) = \mathfrak{I}^\Phi(f)\left(Val_{\mathfrak{M}^\Phi, s^\Phi}(t_1), \ldots, Val_{\mathfrak{M}^\Phi, s^\Phi}(t_n)\right)$
     $= \mathfrak{I}^\Phi(f)\left([t_1]_\sim, \ldots, [t_n]_\sim\right)$ (by inductive assumption)
     $= [f(t_1, \ldots, t_n)]_\sim$ (def. of $\mathfrak{I}^\Phi$) ✓ ✓

2. • $\mathfrak{M}^\Phi, s^\Phi \vDash t_1 \equiv t_2 \Longleftrightarrow Val_{\mathfrak{M}^\Phi,s^\Phi}(t_1) = Val_{\mathfrak{M}^\Phi,s^\Phi}(t_2)$
$\Longleftrightarrow [t_1]_\sim = [t_2]_\sim$ (by 1.)
$\Longleftrightarrow t_1 \sim t_2 \Longleftrightarrow \Phi \vdash t_1 \equiv t_2$ (def. of $\sim$) ✓

• $\mathcal{M}^\Phi, s^\Phi \vDash P(t_1, \ldots, t_n)$ (for P $\neq \equiv$)
$\Longleftrightarrow \big(Val_{\mathfrak{M}^\Phi,s^\Phi}(t_1), \ldots, Val_{\mathfrak{M}^\Phi,s^\Phi}(t_n)\big) \in \mathfrak{I}^\Phi(P)$
$\Longleftrightarrow ([t_1]_\sim, \ldots, [t_n]_\sim) \in \mathfrak{I}^\Phi(P)$ (by 1.)
$\Longleftrightarrow \Phi \vdash P(t_1, \ldots, t_n)$ (def. of $\mathfrak{I}^\Phi$) ✓ ✓

3. $\mathfrak{M}^\Phi, s^\Phi \vDash \exists x_1 \exists x_2 \ldots \exists x_n\, \varphi$

$\Longleftrightarrow$ there are $\underbrace{d_1}_{=[t_1]_\sim}, \ldots, \underbrace{d_n}_{=[t_n]_\sim} \in D^\Phi$ with

$\mathfrak{M}^\Phi, s^\Phi \frac{d_1,\ldots,d_n}{x_1,\ldots,x_n} \vDash \varphi$

(the order in which s is manipulated is irrelevant, since $x_i \neq x_j$ for $i \neq j$)

$\Longleftrightarrow$ there are $t_1, \ldots, t_n \in \mathcal{T}_\mathcal{S}$ with

$\mathfrak{M}^\Phi, s^\Phi \frac{[t_1]_\sim,\ldots,[t_n]_\sim}{x_1,\ldots,x_n} \vDash \varphi$

$\Longleftrightarrow$ there are $t_1, \ldots, t_n \in \mathcal{T}_\mathcal{S}$ with

$\mathfrak{M}^\Phi, s^\Phi \frac{Val_{\mathfrak{M}^\Phi,s^\Phi}(t_1),\ldots,Val_{\mathfrak{M}^\Phi,s^\Phi}(t_n)}{x_1,\ldots,x_n} \vDash \varphi$ (by 1.)

$\Longleftrightarrow$ there are $t_1, \ldots, t_n \in \mathcal{T}_\mathcal{S}$ with

$\mathfrak{M}^\Phi, s^\Phi \vDash \varphi \frac{t_1,\ldots,t_n}{x_1,\ldots,x_n}$ (substitution lemma) ✓

■

So now we know:

$\mathfrak{M}^\Phi, s^\Phi$ satisfies all *atomic* formulas that are derivable from Φ (by 2. of the last lemma). Let us now extend this to *all* formulas in $\mathcal{F}_\mathcal{S}$:

Here we use:
- maximal consistency of Φ in order to derive this for $\neg, \vee$-formulas
- Φ's having instances in order to derive this for $\exists$-formulas

What we need to show:

Lemma 15 *For all maximally consistent $\Phi \subseteq \mathcal{F}_\mathcal{S}$ with instances, for all $\varphi, \psi \in \mathcal{F}_\mathcal{S}$:*

1. $\Phi \vdash \neg\varphi$ *iff* $\Phi \nvdash \varphi$

2. $\Phi \vdash \varphi \vee \psi$ *iff* $\Phi \vdash \varphi$ *or* $\Phi \vdash \psi$

3. $\Phi \vdash \exists x\varphi$ *iff there is a* $t \in \mathcal{T}_\mathcal{S}$ *such that:* $\Phi \vdash \varphi\frac{t}{x}$

Proof.

1. From Φ being *maximally* consistent follows: $\Phi \vdash \varphi$ or $\Phi \vdash \neg\varphi$

   From Φ being maximally *consistent* we have: it is not the case that both $\Phi \vdash \varphi$ and $\Phi \vdash \neg\varphi$

   $\Rightarrow$ So we are done. ✓

2. ($\Rightarrow$) Assume $\Phi \vdash \varphi \vee \psi$

   1.) If $\Phi \vdash \varphi$ then we are done.

   2.) If $\Phi \nvdash \varphi$, then $\Phi \vdash \neg\varphi$ (by maximal consistency)

   $\Rightarrow \Phi \vdash \psi$ (by disjunctive syllogism) ✓

   ($\Leftarrow$) Assume $\Phi \vdash \varphi$ or $\Phi \vdash \psi$

   $\Rightarrow \Phi \vdash \varphi \vee \psi$ ($\vee$-Con.) ✓ ✓

3. ($\Rightarrow$) Assume $\Phi \vdash \exists x\varphi$

   Since Φ contains instances, there is a t with: $\Phi \vdash \exists x\varphi \rightarrow \varphi\frac{t}{x}$

   By modus ponens: $\Phi \vdash \varphi\frac{t}{x}$ ✓

   ($\Leftarrow$) Assume $\Phi \vdash \varphi\frac{t}{x}$ for some $t \in \mathcal{T}_\mathcal{S}$

   $\Rightarrow \Phi \vdash \exists x\varphi$ ($\exists$-Con.) ✓ ✓

■

So finally we have:

Theorem 4 *(Henkin's Theorem)*
For all maximally consistent $\Phi \subseteq \mathcal{F}_\mathcal{S}$ *with instances, for all* $\varphi \in \mathcal{F}_\mathcal{S}$*:*
$\mathfrak{M}^\Phi, s^\Phi \vDash \varphi$ *iff* $\Phi \vdash \varphi$

Proof. By induction over $\mathcal{S}$-formulas:

- Atomic formulas: ✓ (2. of lemma 14)

- $\varphi = \neg\psi$:
  $\mathfrak{M}^\Phi, s^\Phi \models \neg\psi$
  $\iff \mathfrak{M}^\Phi, s^\Phi \not\models \psi$
  $\iff \Phi \nvdash \psi$ (by inductive assumption)
  $\iff \Phi \vdash \neg\psi$ ✓ (1. of lemma 15)
- $\varphi = \psi \vee \rho$:
  $\mathfrak{M}^\Phi, s^\Phi \models \psi \vee \rho$
  $\iff \mathfrak{M}^\Phi, s^\Phi \models \psi$ or $\mathfrak{M}^\Phi, s^\Phi \models \rho$
  $\iff \Phi \vdash \psi$ or $\Phi \vdash \rho$ (by inductive assumption)
  $\iff \Phi \vdash \psi \vee \rho$ ✓ (2. of lemma 15)
- $\varphi = \exists x\psi$:
  $\mathfrak{M}^\Phi, s^\Phi \models \exists x\psi$
  $\iff$ there is a $t \in \mathcal{T}_\mathcal{S}$ with: $\mathfrak{M}^\Phi, s^\Phi \models \psi\frac{t}{x}$ (3. of lemma 14)
  $\iff$ there is a $t \in \mathcal{T}_\mathcal{S}$ with: $\Phi \vdash \psi\frac{t}{x}$ (by inductive assumption)
  $\iff \Phi \vdash \exists x\psi$ ✓ (3. of lemma 15)

■

5.2 Extending Consistent Sets of Formulas to Maximally Consistent Sets of Formulas with Instances

Idea of the proof:

	Φ consistent (and only finitely many free variables)
extend (1st lemma) $\downarrow$	
	Ψ consistent with instances ($\Phi \subseteq \Psi$)
extend (2nd lemma) $\downarrow$	
	Θ maximally consistent with instances ($\Psi \subseteq \Theta$)
conclude (from 5.1) $\downarrow$	
	Θ satisfiable ($\Rightarrow \Phi \subseteq \Theta$ satisfiable)

Afterwards: get rid of restriction to finitely many free variables!

Lemma 16 *Let $\Phi \subseteq \mathcal{F}_{\mathcal{S}}$, such that Φ is consistent and the set of variables that occur freely in some formula in Φ is finite:*
there is a $\Psi \subseteq \mathcal{F}_{\mathcal{S}}$ with $\Phi \subseteq \Psi$, such that Ψ constains instances and Ψ is consistent.

Proof. We know: $\mathcal{F}_{\mathcal{S}}$ is countable (lemma 2)
$\Rightarrow$ the set of formulas in $\mathcal{F}_{\mathcal{S}}$ that begin with $\exists$ is countable
So let $\exists x_0\varphi_0,\ \exists x_1\varphi_1,\ \exists x_2\varphi_2, \ldots$ be an enumeration of existentially quantified formulas in $\mathcal{F}_{\mathcal{S}}$
(note that each "x_i" is a metavariable that stands for *some* of our first-order variables v_j – in particular, x_i is not necessarily identical to v_i!)
Now we define "instances" for this sequence of formulas:

$$\psi_n := \exists x_n\varphi_n \rightarrow \varphi_n \frac{y_n}{x_n}$$

where y_n is the variable v_k with the least index k such that

v_k does not occur freely in $\underbrace{\Phi \cup \{\psi_0, \ldots, \psi_{n-1}\}}_{\Phi_n} \cup \{\exists x_n\ \varphi_n\}$

Such a variable y_n – i.e., v_k – *does* exist:

there are only finitely many free variables in $\Phi_n \cup \{\exists x_n\ \varphi_n\}$
(since there are only finitely many variables in Φ)

Now let Ψ be defined as follows:

$$\Psi := \bigcup_{n \in \mathbb{N}_0} \Phi_n$$

$\Rightarrow$ • $\Phi \subseteq \Psi \subseteq \mathcal{F}_{\mathcal{S}}$ ✓

• Ψ contains instances ✓

• Ψ is consistent

Because:

$(\Phi =)\Phi_0 \subseteq \Phi_1 \subseteq \Phi_2 \subseteq \cdots$ and $\Psi = \bigcup_{n \in \mathbb{N}_0} \Phi_n$

(Corresponding symbol sets: $\mathcal{S}_0 \subseteq \mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \cdots$ with $\mathcal{S}_i := \mathcal{S}$ for all $i \geq 0$)

$\Longrightarrow$ So we can apply lemma 12:

If all formula sets Φ_n are consistent with respect to $\mathcal{S}_n = \mathcal{S}$, then the same holds for $\Psi = \bigcup_{n \in \mathbb{N}_0} \Phi_n$.

But the sets Φ_n are indeed consistent as induction over n shows:

induction basis: $\Phi_0 = \Phi$ is consistent (by assumption)✓

inductive assumption: Φ_n is consistent

inductive step ($n \to n+1$):

assume for contradiction that Φ_{n+1} is *not* consistent (where $\Phi_{n+1} = \Phi_n \cup \{\psi_n\}$):

$\Longrightarrow$ every formula in $\mathcal{F}_{\mathcal{S}}$ is derivable from Φ_{n+1} (by lemma 8)

In particular: let φ be a *sentence* in $\mathcal{F}_{\mathcal{S}}$

$\Longrightarrow \Phi_{n+1} \vdash \varphi$

Thus: there is a $\Gamma \subseteq \Phi_n$, such that the sequent $\Gamma\psi_n\varphi$ is derivable in the sequent calculus.

Consider such a derivation:

1.	$\ldots$	
$\vdots$	$\vdots$	
m.	$\Gamma \underbrace{\psi_n}_{\exists x_n \varphi_n \to \varphi_n \frac{y_n}{x_n} = (\neg \exists x_n \varphi_n \vee \varphi_n \frac{y_n}{x_n})} \varphi$	by def.

Extend derivation:

m+1.	$\Gamma \neg \exists x_n \varphi_n$	$\neg \exists x_n \varphi_n$	(Ass.)
m+2.	$\Gamma \neg \exists x_n \varphi_n$	$(\neg \exists x_n \varphi_n \vee \varphi_n \frac{y_n}{x_n})$	($\vee$-Con.) with m+1.
m+3.	$\Gamma \neg \exists x_n \varphi_n$	$(\neg \exists x_n \varphi_n \vee \varphi_n \frac{y_n}{x_n})$ φ	(Ant.) with m.
m+4.	$\Gamma \neg \exists x_n \varphi_n$	φ	(CS) with m+3., m+2.

$\vdots$	(analogously)		
m+8.	$\Gamma\ \varphi_n\frac{y_n}{x_n}$	φ	$\ldots$
m+9.	$\Gamma\ \exists x_n\varphi_n$	φ	($\exists$-Ant.) with m+8.
	since y_n is not in $free(\Gamma\ \exists x_n\varphi_n\ \ \varphi)$: $y_n \notin free(\Gamma\ \exists x_n\varphi_n)$ by choice of y_n $(\Gamma \subseteq \Phi_n)$ and $y_n \notin free(\varphi)$, since φ is a sentence		
m+10.	$\Gamma\varphi$		(PC) with m+9., m+4.

so: for both, say, $\varphi = \exists v_0 v_0 \equiv v_0$ and for $\varphi = \neg\exists v_0 v_0 \equiv v_0$
we can show: φ is derivable from Φ_n
$\Longrightarrow$ Φ_n is not consistent, which contradicts the inductive assumption

$\Longrightarrow$ Φ_{n+1} is consistent

$\Longrightarrow$ By induction: every set Φ_n is consistent $\Longrightarrow$ Ψ is consistent✓

■

Lemma 17 *Let $\Psi \subseteq \mathcal{F}_\mathcal{S}$ with Ψ consistent:*
there is a $\Theta \subseteq \mathcal{F}_\mathcal{S}$ such that $\Psi \subseteq \Theta$ and Θ is maximally consistent.

(Note that there is no assumption on instances.)

Proof.
Let $\varphi_0, \varphi_1, \varphi_2, \ldots$ be an enumeration of the formulas in $\mathcal{F}_\mathcal{S}$ (such an enumeration exists since $\mathcal{F}_\mathcal{S}$ is countable by lemma 2).
Now define:

$$\Theta_0 := \Psi$$

and for $n \geq 0$: $\quad \Theta_{n+1} := \begin{cases} \Theta_n \cup \{\varphi_n\} & \text{if } \Theta_n \cup \{\varphi_n\} \text{ consistent} \\ \Theta_n & \text{else} \end{cases}$

Let $\Theta := \bigcup_{n \in \mathbb{N}_0} \Theta_n$: so

- $\Psi \subseteq \Theta \subseteq \mathcal{F}_\mathcal{S}$ ✓
- Θ consistent: analogous to before ✓

  $(\Theta_0 \subseteq \Theta_1 \subseteq \Theta_2 \subseteq \cdots,\ \Theta = \bigcup_{n \in \mathbb{N}_0}$

Θ_n are consistent by definition
$\Longrightarrow$ as before: Θ is consistent by lemma 12)

- Θ is maximal:

  Since:

  > Let $\varphi \in \mathcal{F}_\mathcal{S}$ be chosen arbitrarily
  > Case 1: $\Theta \vdash \neg\varphi$ ✓
  > Case 2: $\Theta \nvdash \neg\varphi$
  > $\Longrightarrow$ $\Theta \cup \{\varphi\}$ is consistent (proof analogous to 2. of lemma 10)
  > By enumeration there is an $n \in \mathbb{N}_0$ with $\varphi = \varphi_n$
  > $\Longrightarrow$ $\Theta \cup \{\varphi_n\}$ consistent
  > $\Longrightarrow$ $\Theta_n \cup \{\varphi_n\}$ consistent (because $\Theta_n \subseteq \Theta$)
  > $\Longrightarrow$ $\Theta_{n+1} = \Theta_n \cup \{\varphi_n\}$ (since consistent)
  > $\Longrightarrow$ $\varphi = \varphi_n \in \Theta_{n+1} \subseteq \Theta$
  > $\Longrightarrow$ $\varphi \in \Theta$
  > $\Longrightarrow$ $\Theta \vdash \varphi$ ✓
  >
  > In both cases: $\Theta \vdash \varphi$ or $\Theta \vdash \neg\varphi$
  > $\Longrightarrow$ Θ is maximal ✓

■

Corollary 3 *Let $\Phi \subseteq \mathcal{F}_\mathcal{S}$, Φ consistent with only finitely many free variables in Φ:*
It follows that Φ is satisfiable.

Proof.
Extend Φ to consistent Ψ with instances according to lemma 16,
extend Ψ to maximally consistent Θ according to lemma 17
(since $\Psi \subseteq \Theta \Longrightarrow \Theta$ contains instances):
but this implies that $\Phi \subseteq \Theta$ is satisfiable by Henkin's theorem. ■

Now we are finally in the position to prove (one version of) the *completeness theorem*:

Theorem 5 ***(Completeness Theorem)***
For all $\Phi \subseteq \mathcal{F}_{\mathcal{S}}$ *it holds:*

If Φ *is consistent, then* Φ *is satisfiable.*

Proof. Since the set of free variables in Φ is perhaps infinite, we cannot apply corollary 3 directly in order to prove the completeness theorem.

So we have to make a small "detour":

Let $\mathcal{S}' := \mathcal{S} \cup \{c_0, c_1, c_2, \ldots\}$, where $c_0, c_1, c_2, \ldots$ are pairwise distinct "new" constants that are not yet contained in $\mathcal{S}$.

Let furthermore $\varphi' := \varphi\frac{c_0, c_1, \ldots, c_{n_\varphi}}{v_0, v_1, \ldots, v_{n_\varphi}}$ (for $\varphi \in \mathcal{F}_{\mathcal{S}}$)
where n_φ is minimal such that $free(\varphi) \subseteq \{v_0, \ldots, v_{n_\varphi}\}$
(this substitution obviously has no effect on variables that only occur in φ as bound variables).

Finally, let $\Phi' := \{\varphi' | \varphi \in \Phi\}$.

Note: Φ' is a set of *sentences* (over the symbol set $\mathcal{S}'$).

Subsequent proof structure:

Φ is $\mathcal{S}$-consistent by assumption,
$\Longrightarrow \Phi'$ is $\mathcal{S}'$-consistent
$\Longrightarrow \Phi'$ is satisfiable (there is an $\mathcal{S}'$-model)
$\Longrightarrow \Phi$ is satisfiable (there is an $\mathcal{S}$-model)

Now we will show these claims in a step-by-step manner:

- Φ is by assumption $\mathcal{S}$-consistent, therefore Φ' is $\mathcal{S}'$-consistent,

  because:

  1. Let $\Psi \subseteq \Phi'$, Ψ finite

  $\Longrightarrow \Psi = \{\varphi_1', \ldots, \varphi_n'\}$ for $\varphi_1, \ldots, \varphi_n \in \Phi$

  Since $\{\varphi_1, \ldots, \varphi_n\} \subseteq \Phi$

  $\Longrightarrow \{\varphi_1, \ldots, \varphi_n\}$ $\mathcal{S}$-consistent

  $\{\varphi_1, \ldots, \varphi_n\}$ can only contain finitely many variables
  (because it is a finite set)

$\Longrightarrow \{\varphi_1, \ldots, \varphi_n\}$ satisfiable (by corollary 3)

i.e., there is an $\mathcal{S}$-model $\mathfrak{M}$ and a variable assignment s over $\mathfrak{M}$, such that $\mathfrak{M}, s \models \{\varphi_1, \ldots, \varphi_n\}$

Now we extend $\mathfrak{M}$ to an $\mathcal{S}'$-model $\mathfrak{M}' = (D', \mathfrak{I}')$ where:

$D' := D$

$\mathfrak{I}' \mid_{\mathcal{S}} \equiv \mathfrak{I}$

$\mathfrak{I}'(c_i) := s(v_i)$ (for "new" c_i)

It follows that for all $\varphi \in \mathcal{F}_{\mathcal{S}}$:

$$\begin{aligned}
\mathfrak{M}, s \models \varphi &\iff \mathfrak{M}, s\frac{s(v_0),\ldots,s(v_{n_\varphi})}{v_0,\ldots,v_{n_\varphi}} \models \varphi \\
&\iff \mathfrak{M}', s\frac{s(v_0),\ldots,s(v_{n_\varphi})}{v_0,\ldots,v_{n_\varphi}} \models \varphi \quad \text{(coincidence lemma)} \\
&\iff \mathfrak{M}', s\frac{\mathfrak{I}'(c_0),\ldots,\mathfrak{I}'(c_{n_\varphi})}{v_0,\ldots,v_{n_\varphi}} \models \varphi \quad \text{(def. } \mathfrak{I}'\text{)} \\
&\iff \mathfrak{M}', s\frac{Val_{\mathfrak{M}',s}(c_0),\ldots,Val_{\mathfrak{M}',s}(c_{n_\varphi})}{v_0,\ldots,v_{n_\varphi}} \models \varphi \\
&\iff \mathfrak{M}', s \models \varphi\frac{c_0,\ldots,c_{n_\varphi}}{v_0,\ldots,v_{n_\varphi}} \quad \text{(substitution lemma)} \\
&\iff \mathfrak{M}', s \models \varphi' \quad \text{(def. } \varphi'\text{)}
\end{aligned}$$

Since $\mathfrak{M}, s \models \{\varphi_1, \ldots, \varphi_n\}$

$\Longrightarrow \mathfrak{M}', s \models \underbrace{\{\varphi'_1, \ldots, \varphi'_n\}}_{\Psi}$

$\Longrightarrow \Psi$ is satisfiable (it has an $\mathcal{S}'$-model)

$\Longrightarrow \Psi$ is $\mathcal{S}'$-consistent (by corollary 2)

2. We found that every finite subset of Φ' is $\mathcal{S}'$-consistent

$\Longrightarrow \Phi'$ is $\mathcal{S}'$-consistent (by lemma 9)

- Hence, Φ' is $\mathcal{S}'$-consistent and contains only finitely many free variables (namely 0)

  $\Longrightarrow \Phi'$ is satisfiable, i.e., there is an $\mathcal{S}'$-model $\mathfrak{M}'$, such that $\mathfrak{M}' \models \Phi'$ (as follows from corollary 3).

  We do not need to refer to a variable assignment, since Φ' only contains sentences.

- Now we restrict $\mathfrak{M}'$ to a model $\mathfrak{M}$ over $\mathcal{S}$ again:

  $D := D'$

  $\mathfrak{I} \equiv \mathfrak{I}' \mid_{\mathcal{S}}$

  and we set $s(v_i) := \mathfrak{I}'(c_i)$ for all variables v_i

  As before it follows that: $\mathfrak{M}, s \vDash \varphi \Longleftrightarrow \mathfrak{M}', s \vDash \varphi'$ for all $\varphi \in \mathcal{F}_{\mathcal{S}}$

  Because $\mathfrak{M}' \vDash \Phi'$

  $\Longrightarrow \mathfrak{M}, s \vDash \Phi$, i.e., Φ is satisfiable.

■

So we can finalise the plan of our proof and conclude the originally intended version of the completeness theorem:

Theorem 6 ***(Completeness Theorem)***
For all $\Phi \subseteq \mathcal{F}_{\mathcal{S}}$, for all $\varphi \in \mathcal{F}_{\mathcal{S}}$, it holds:

If $\Phi \vDash \varphi$, then $\Phi \vdash \varphi$.

I.e.: if φ follows logically (semantically) from Φ, then φ is derivable from Φ on the basis of the sequent calculus; thus the sequent calculus is *complete.*

This implies immediately:

Remark 17

- *φ is provable iff φ is logically true.*
- *Φ is consistent iff Φ is satisfiable.*
- *Since logical consequence and satisfiability are independent of the particular choice of $\mathcal{S}$, the same must hold for derivability and consistency.*

Isn't that a *great* result?

5.3 Consequences and Applications

We will now turn to two important consequences of the completeness theorem (and of the methods by which we proved it): the *theorem of Loewenheim-Skolem* and the *compactness theorem*.

Theorem 7 *(Loewenheim-Skolem)*
For all $\Phi \subseteq \mathcal{F}_{\mathcal{S}}$*:*
If Φ *is satisfiable, then there are* $\mathfrak{M}, s$ *such that* $\mathfrak{M}, s \vDash \Phi$ *and:*
the domain D *of* $\mathfrak{M}$ *is countable.*

Proof.
The proof can be extracted from the proof of the completeness theorem:
Φ is satisfiable
$\Longrightarrow \Phi$ consistent
$\Longrightarrow \Phi'$ is consistent, where Φ' is as in the proof of theorem 5 (consistency was shown there)
$\Longrightarrow \Phi'$ has a model $\mathfrak{M}'$ with a countable domain the members of which are equivalence classes of the terms of $\mathcal{T}_{\mathcal{S}'}$ (by corollary 3 and Henkin's theorem)
$\Longrightarrow \Phi$ is satisfied by $\mathfrak{M}, s$ where $\mathfrak{M}$ has a countable domain (as shown in the proof of theorem 5) ■

Theorem 8 *(Compactness)*
For all $\Phi \subseteq \mathcal{F}_{\mathcal{S}}$*, for all* $\varphi \in \mathcal{F}_{\mathcal{S}}$*:*

1. $\Phi \vDash \varphi$ *if and only if there is a* $\Psi \subseteq \Phi$ *with* Ψ *finite and* $\Psi \vDash \varphi$*.*

2. Φ *is satisfiable if and only if for all* $\Psi \subseteq \Phi$ *with* Ψ *finite:* Ψ *is satisfiable.*

Proof. We already know that the proof-theoretic analogues to these claims hold (by lemma 7 and lemma 9). But this means we are done by the soundness and the completeness theorem. ■

The theorem of Loewenheim-Skolem and the compactness theorem are important tools in model theory and have several surprising implications and applications.

Example:

Consider the first-order theory of set theory:
Let $\mathcal{S}_{\text{set}} = \{ \underbrace{\in}_{\text{binary predicate}} \}$

(+ optional: constant $\varnothing$
binary predicate $\subseteq$
binary functions signs $\cap$, $\cup$, $\{\_\,,\ \_\}, \ldots$)

Set of set-theoretic definitions and axioms:

1. Definitions:

   (a) Definition of $\varnothing$:
   $\forall y(\varnothing = y \leftrightarrow \forall z\, \neg z \in y)$

   (b) Definition of $\subseteq$:
   $\forall x \forall y(x \subseteq y \leftrightarrow \forall z(z \in x \rightarrow z \in y))$

   (c) Definition of $\{\ ,\ \}$:
   $\forall x \forall y \forall z(\{x, y\} = z \leftrightarrow$
   $\forall w(w \in z \leftrightarrow w = x \vee w = y))$
   (and let us abbreviate $\{y, y\}$ by means of $\{y\}$)

   (d) Definition of $\cup$:
   $\forall x \forall y \forall z(x \cup y = z \leftrightarrow$
   $\forall w(w \in z \leftrightarrow (w \in x \vee w \in y)))$

   (e) Definition of $\cap$:
   $\forall x \forall y \forall z(x \cap y = z \leftrightarrow$
   $\forall w(w \in z \leftrightarrow (w \in x \wedge w \in y)))$

2. Axioms:

   (a) Axiom of Extensionality:
   $\forall x \forall y(\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y)$
   ("Two sets that have the same members are equal")

(b) Axiom Schema of Separation:
$\forall x_1 \ldots \forall x_n \forall x \exists y \forall z (z \in y \leftrightarrow$
$z \in x \wedge \varphi[z, x_1, \ldots, x_n])$
("For every set x and for every property E that is expressed by a formula φ with free variables $z, x_1, \ldots, x_n$ there is a set $\{z \in x \,|\, z$ has the property $E\}$")

(c) Axiom of Pairs:
$\forall x \forall y \exists z \forall w (w \in z \leftrightarrow w = x \vee w = y)$
("For every two sets x, y there is the pair set $\{x, y\}$")

(d) Axiom of Unions:
$\forall x \exists y \forall z (z \in y \leftrightarrow \exists w (w \in x \wedge z \in w))$
("For every set x there is a set y, which contains precisely the members of the members of x")

(e) Powerset Axiom:
$\forall x \exists y \forall z (z \in y \leftrightarrow z \subseteq x)$
("For every set x there is the power set y of x")

(f) Axiom of Infinity:
$\exists x (\varnothing \in x \wedge \forall y (y \in x \rightarrow y \cup \{y\} \in x))$
("There is an infinite set, namely the set that contains $\varnothing$, $\{\varnothing\}$, $\{\varnothing, \{\varnothing\}\}$,... as members")

(g) Axiom of Choice:
$\forall x (\neg \varnothing \in x \wedge \forall u \forall v (u \in x \wedge v \in x \wedge \neg u \equiv v \rightarrow u \cap v \equiv \varnothing) \rightarrow$
$\exists y \forall w (w \in x \rightarrow \exists ! z \; z \in w \cap y))$
("For every set x that has non-empty and pairwise disjoint sets as its members there is a (choice) set y that contains for each set in x precisely one member")

Remark: For some purposes the additional so-called Axiom of Replacement is needed as well.

Practically all theorems of standard mathematics can be *derived* from this set of definitions and axioms.

At the same time, Loewenheim-Skolem tells that if this set of definitions and axioms is consistent, then it has a model with a *countable* domain!!!!

Another example – now we consider *arithmetic*:

Let $\mathcal{S}_{\text{arith}} = \{\overline{0},\ \overline{1},\ \overline{+},\ \overline{\cdot}\}$;

standard model of arithmetic: $(\mathbb{N}_0, \mathfrak{I})$ with $\mathfrak{I}$ as expected
(so $\mathfrak{I}(\overline{0}) = 0, \mathfrak{I}(\overline{+}) = +$ on $\mathbb{N}_0, \ldots$).

Let Φ_{arith} be the set of $\mathcal{S}_{\text{arith}}$-sentences that are satisfied by this model, i.e.:
$\underbrace{\Phi_{\text{arith}}}_{\text{"arithmetic"}} = \{\varphi \in \mathcal{F}_{\mathcal{S}_{\text{arith}}} | \varphi \text{ sentence, } (\mathbb{N}_0, \mathfrak{I}) \vDash \varphi\}$

Now consider

$$\Psi = \Phi_{\text{arith}} \cup \{\neg\, x \equiv \overline{0},\ \neg\, x \equiv \overline{1},\ \neg\, x \equiv \overline{1} \mathbin{\overline{+}} \overline{1},\ \neg\, x \equiv (\overline{1} \mathbin{\overline{+}} \overline{1}) \mathbin{\overline{+}} \overline{1}, \ldots\}$$

It holds that every finite subset of Ψ is satisfiable:
just take the standard model of artithmetic and choose s in the way that $s(x)$ is a sufficiently large natural number (for a given finite subset of Ψ, $s(x)$ has to be large enough to be greater than any number denoted by any of the right-hand sides of the negated equations in the subset).

By the compactness theorem, this implies: Ψ is satisfiable, i.e., there is a model $\mathfrak{M}'$ and a variable assignment s', such that $\mathfrak{M}', s' \vDash \Psi$.

It follows:

$s'(x) \neq Val_{\mathfrak{M}',s'}(\overline{0})$, since $\neg\, x \equiv \overline{0} \in \Psi$
$s'(x) \neq Val_{\mathfrak{M}',s'}(\overline{1})$, since $\neg\, x \equiv \overline{1} \in \Psi$
$s'(x) \neq Val_{\mathfrak{M}',s'}(\overline{1 + 1})$, since $\neg\, x \equiv \overline{1 + 1} \in \Psi$
$\vdots$

If we finally identify the objects $Val_{\mathfrak{M}',s'}(\overline{1 + \ldots + 1})$ with our standard natural numbers, we get:
There exists a model of the set of true arithmetical sentences, such that the domain of this model contains a "new number" $s'(x)$ that is different from any of the "old" natural numbers $0, 1, 2, 3, \ldots$!!!!

5.4 Problem Set 6

1. Let $\mathcal{S}$ be an arbitrary symbol set.

   Let $\Phi = \{v_0 \equiv t \,|t \in \mathcal{T}_{\mathcal{S}}\} \cup \{\exists v_1 \exists v_2 \neg\, v_1 \equiv v_2\}$.

   Show:

   - Φ is consistent
   - there is no formula set $\Psi \subseteq \mathcal{F}_{\mathcal{S}}$ with $\Psi \supseteq \Phi$, such that Ψ is consistent and contains instances.

2. (This problem counts for award of CREDIT POINTS.)
   Explain why the following logical implication holds:

$$\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y) \vDash$$

$$\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \leftrightarrow x = y)$$

3. Prove: A map with countably many countries can be coloured by using at most four colours if and only if each of its finite submaps can be coloured by using at most four colours.

   (Hint: choose a symbol set $\mathcal{S}$ in which ever constant represents a country, in which there are four unary predicates that represent four colours, and in which there is a binary predicate that stands for the neighbourhood relation between countries; represent maps as sets of formulas for this symbol set; apply the compactness theorem.)

4. Let P be a binary predicate in $\mathcal{S}$.

   Prove that the formula

   $\forall x \neg P(x, x) \wedge \forall x \forall y \forall z (P(x, y) \wedge P(y, z) \rightarrow P(x, z)) \wedge \forall x \exists y P(x, y)$

   can only be satisfied by *infinite* models.

5. Prove: there is no formula φ, such that for all models $\mathfrak{M} = (D, \mathfrak{I})$ and for all variable assignments s holds:

   $\mathfrak{M}, s \vDash \varphi$ if and only if D is infinite.

   (Hint: use the compactness theorem.)

6 The Incompleteness Theorems for Arithmetic

6.1 Preliminary Remarks

Here is a review of what we have achieved so far:

- we introduced and studied first-order languages;
- then we considered models of such languages and defined semantic concepts in terms of these models;
- we developed the sequent calculus for first-order languages and defined proof-theoretic notions on the basis of it;
- finally, we proved the soundness and completeness theorems as well as consequences of the latter (the theorem of Löwenheim-Skolem and the compactness theorem).

In particular, we know (remember the end of subsection 5.2): for all first-order formulas φ,

φ is logically true (i.e., true in all models) iff

φ is provable (by means of the sequent calculus)

Is it possible to prove similar results for sets of formulas other than the set of logically true formulas?

Here is a famous question that was discussed intensively in the 1920s/1930s: Is there a calculus $\mathcal{C}$ of axioms and rules, such that for all *arithmetical* first-order formulas φ,

φ is true in the standard model of arithmetic iff

φ is derivable in the calculus $\mathcal{C}$

As was proved by Kurt Gödel in 1931, the answer to this question is NO! (The same holds for all calculi of axioms and rules that include a "sufficient" amount of arithmetic.)

In a nutshell, the reasons for this fact are as follows:

1. A calculus – later we will say: a *recursively axiomatized theory* – yields a mechanical procedure – later we will say: a *register program* – by which all formulas that are derivable in the calculus can be generated – later we will say: *register-enumerated* – in a systematic manner.

2. If a set of formulas is (i) register-enumerable and (ii) contains for every formula φ of a given first-order language either φ or $\neg\varphi$, then there is a mechanical procedure that decides – later we will say: *register-decides* – for every formula of this language whether it is a member of that set or not.

3. Every register-decidable set of formulas can be represented by an arithmetical formula (on the basis of coding formulas by natural numbers in a mechanical – as we will say: *register-computable* – way).

4. Assume that the set of arithmetical formulas that are true in the standard model of arithmetic is identical to the set of formulas that can be derived in some arithmetical calculus: It follows from 1. that the set of true arithmetical formulas would be register-enumerable. Furthermore it is clear that the set of true arithmetical formulas contains for every arithmetical formula φ either φ or $\neg\varphi$, so by 2. the set would have to be register-decidable. Hence, 3. implies that the set of true arithmetical formulas would be represented by an arithmetical formula.

5. But the set of formulas that are true in the standard model of arithmetic is not represented by any arithmetical formula. For otherwise one could show that there would be an arithmetical sentence that would say about itself (via coding) "I am not true": But that sentence would be true in the standard model of arithmetic if and only if it is not true in the standard model of arithmetic (contradiction!).

In order to prove this thoroughly it is necessary to define the notions of register-decidability, register-enumerability, register-computability, and recursive axiomatizability precisely and to study their properties. Historically, this led to the development of a new branch of mathematical logic: computability theory (or recursion theory).

For the purposes of this course, we will restrict ourselves just to a *sketch* of the proof of Gödel's first incompleteness theorem (the second one will only be mentioned). Further details can be found in standard textbooks on proof theory and computability theory.

6.2 Formalising "Computer" in Terms of *Register Machine*

Our ultimate goal is to prove that the set of first-order arithmetical truths cannot be generated by means of explicitly stated rules which connect finitely many premises to a conclusion – unlike the set of formulas of a first-order language, the set of terms of a first-order language, and the set of logically true formulas of a first-order language which can all be generated by means of such rules. In order to prove this result, we have to determine what all the sets that *can* be enumerated on the basis of these "mechanical" rules have in common: roughly, the point will be that a program or a procedure on a *computer* would in principle be able – given unrestricted memory and time – to enumerate the members of any such set in a step-by-step manner. But in order to do make *that* precise, we first need a mathematical definition of "computer program" or "computer procedure".

Since the 1930s, various such definitions have been put forward in which the intutive notion of procedure is explained formally in terms of

- Turing machines
- Register machines
- Recursive functions
- the Lambda Calculus

  ⋮

As it was proven later, all of these definitions turn out to be essentially equivalent to each other: if a problem can be solved by a computer procedure in the one sense, then it can also be solved by a computer procedure in the other sense, and vice versa; if a set is enumerable by a procedure in the one sense, then the set is enumerable by a procedure in the other sense, and vice versa; etc. We are going to focus on one particularly simple formal notion of "computer program" or "computer procedure": computer programs in the sense of so-called *register programs* that are considered to run sequentially on corresponding *register machines* (the latter not being actual computers but mathematical idealisations thereof).

We start by fixing an alphabet $\mathcal{A} = \{a_0, \ldots, a_r\}$ ($r \in \mathbb{N}_0$).

Intuitively, our register machines can be thought of as (i) storing words over the alphabet $\mathcal{A}$ in their memory, as well as (ii) manipulating the stored words by simple operations as being determined by the program. We will regard the memory of a register machine as consisting of certain units or slots which are members of the set $R_0, R_1, R_2, \ldots$ of so-called "registers": at each step of a computation of the machine each of the machine's registers is assumed to contain exactly one word (such a word may be of arbitrary finite length; the content of a register may change in the course of the computation). Since we also want to allow for empty registers, we presuppose that the set $\mathcal{A}^*$ of words over $\mathcal{A}$ includes an "empty word" which we will denote by: $\Box$

Here is the exact statement of what we understand by a register program (over $\mathcal{A}$):

Definition 18 *A (register) program* P *is a finite sequence* $\alpha_0, \alpha_2, \ldots, \alpha_k$ *that satisfies the following conditions:*

- *For every* i *with* $0 \leq i \leq k$, α_i *is an instruction that starts with the label* i *and the remainder of which is of either of the following five forms:*

  1. *LET* $R_m = R_m + a_n$
     (for $m, n \in \mathbb{N}_0$, $n \leq r$*)*
     This is the "Add-Instruction": Add the symbol a_n *on the right-hand side of the word in register* R_m.
  2. *LET* $R_m = R_m - a_n$
     (for $m, n \in \mathbb{N}_0$, $n \leq r$*)*
     This is the "Subtract-Instruction": If the word stored in register R_m *ends with the symbol* a_n*, then delete* a_n *at that position; otherwise leave the word unchanged.*
  3. *IF* $R_m = \Box$ *THEN* L *ELSE* L_0 *OR ... OR* L_r
     (for $m \in \mathbb{N}_0$ *and* $L, L_0, \ldots, L_r \in \mathbb{N}_0$ *with* $L, L_0, \ldots, L_r \leq k$*)*
     This is the "Jump-Instruction": If the register R_m *contains the empty word, then go to the instruction labelled* L*; if the word in register* R_m *ends with the symbol* a_0*, then go to the instruction labelled* L_0*; ...; if the word in register* R_m *ends with the symbol* a_r*, then go to the instruction labelled* L_r.

4. *PRINT*
   The "Print-Instruction": Print (as output) the word stored in register R_0.

5. *HALT*
   The "Halt-Instruction": Stop the running of the program.

- α_k*, and only* α_k*, is a Halt-Instruction.*

(Note that within the instructions of a register program, '+' and '−' do not stand for the usual arithmetical operations but rather for syntactic operations on words stored in registers.)

Register programs might seem much simpler than programs in standard computer languages – and indeed they are – but one can nevertheless prove that register programs are actually just as powerful as the latter.

A register program P determines a "computation" on a register machine in the following way: assume a machine that contains all the registers mentioned in P and whose program is P. At the beginning, all registers except for R_0 contain the empty word; the content of R_0 is regarded as the "input" (which can be any word in $\mathcal{A}^*$ whatsoever, whether empty or non-empty). Then the register machine works through P in a stepwise fashion, starting with the instruction with label 0. After having executed an instruction with label L, the machine moves on to the instruction with label $L + 1$, except for the cases of a Jump-Operation (the result of which is a jump to an instruction with a particular label) or the Halt-Operation (the result of which is the end of the computation). Whenever a Print-Instruction is encountered, the content of R_0 at that point of computation is printed out (an "output"). The machine will only stop after executing the Halt-Operation with the maximal instruction label k.

Here is an example of a register program:

Example 21 *Let* $\mathcal{A} = \{|\}$*. So* $\mathcal{A}^*$ *consists of:* $\Box, |, ||, |||, \ldots$
Call the following program P_0*:*

0. *IF* $R_0 = \Box$ *THEN 6 ELSE 1*

1. *LET* $R_0 = R_0 - |$

2. *IF* $R_0 = \Box$ *THEN 5 ELSE 3*

3. *LET* $R_0 = R_0 - |$

4. *IF* $R_0 = \Box$ *THEN 6 ELSE 1*

5. *LET* $R_0 = R_0 + |$

6. *PRINT*

7. *HALT*

P_0 successively deletes strokes from the input word in R_0 until finally the empty word is obtained. The printed output is determined to be $\Box$ in case R_0 initially included an even number of stokes (where 0 counts as even), and the output is $|$ otherwise, i.e., in case R_0 initially consisted of an odd number of strokes.

Let us introduce a way of expressing such input-output patterns more succinctly: We say that a program P is started with a word $\zeta \in \mathcal{A}^*$ if, when P is initiated on a register machine, ζ is stored in R_0 (and all other registers contain the empty word). In such a case we will write:

$$\mathsf{P} : \zeta \to \ldots$$

In order to say something about the behaviour of a program given an input ζ, we can extend this notation in either of the following ways:

- If we want to express that P, started with ζ, eventuelly reaches the Halt-Instruction (rather than running forever), we say

$$\mathsf{P} : \zeta \to \textit{halt}$$

- If we want to express that P, started with ζ, eventuelly reaches the Halt-Instruction, but additionally we want to point out that in the course of the computation P has given exactly one output η, then we say

$$\mathsf{P} : \zeta \to \eta$$

- If we want to express that P, started with ζ, never reaches the Halt-Instruction (which is possible in light of the Jump-Instructions), then we say

$$\mathsf{P} : \zeta \to \infty$$

Using this terminology, we can thus describe the input-output pattern of P_0 in example 21 by means of

$$\mathsf{P}_0 : \underbrace{|| \ldots |}_{n} \to \square \;\text{ if } n \text{ is even}$$

$$\mathsf{P}_0 : \underbrace{|| \ldots |}_{n} \to | \;\text{ if } n \text{ is odd}$$

So P_0 *decides* in finitely steps whether the given input stroke sequence encodes an even number or not.

Let us consider another example:

Example 22 *Let $\mathcal{A} = \{a_0, \ldots, a_r\}$. Call the following program P_1:*

0. PRINT

1. LET $R_0 = R_0 + a_0$

2. IF $R_0 = \square$ THEN 0 ELSE 0 OR 0 ... OR 0

3. HALT

In this case we have that

$$\mathsf{P}_1 : \zeta \to \infty$$

If P_1 is started with $\zeta \in \mathcal{A}^*$, P_1 prints out successively the words $\zeta, \zeta a_0, \zeta a_0 a_0, \ldots$ In other words: P_1 *enumerates* the set $\{\zeta, \zeta a_0, \zeta a_0 a_0, \ldots\}$ of words over $\mathcal{A}$.

6.3 Register-Decidability, Register-Enumerability, Register-Computability

We are now able to study the notions of register-decidability, register-enumerability, and register-computability – which turn out to be necessary for the proof of the incompleteness theorem for arithmetic – in more detail. Let us start with register-decidability.

We have already seen a decision procedure at work in example 21. Now I want to present another example program which solves a decision problem, but instead of formulating in exact terms what the instructions in the corresponding register program are like, I will describe the program just on the informal level while counting on your programming abilities being so refined that you are able to turn this informal description into a register program – the only important point being that this can be done *in principle* (this is also how computer scientists first approach a software problem):

Example 23 *It is not difficult to set up a computer procedure that* decides *whether an arbitrary natural number n is prime:*

1. *Given input n.*

2. *If $n = 0$ or $n = 1$ then n is not prime: Output any string distinct from "Yes" (e.g., "No" or "Goedel").*

3. *Otherwise:*

   (a) *Test numbers $2, \ldots, n-1$ whether they divide n*
   (this can be programmed easily).

   (b) *If none of these numbers divides n, then n is prime: Output "Yes".*

   (c) *Otherwise: n is not prime. Output any string distinct from "Yes".*

If this procedure is started with the natural number 7 as an input, its output is "Yes"; if it is started with input 12, its output is distinct from "Yes", which is supposed to indicate that the answer to the question "Is 12 prime?" is negative.

One step in translating this informal program or algorithm into a proper register program would be to show that programs which were said to take natural numbers as their inputs can just as well be regarded as taking strings of symbols as their inputs (as we have already seen in example 21) – e.g., use the decimal representation of n, which is a string of numerals out of "0",...,"9", as the input that corresponds to n. Accordingly, the outputs of such procedures (such as "Yes") are strings of symbols.
We can also always in principle restrict ourselves to *finite* alphabets, since countably infinite alphabets can be "simulated" by finite alphabets: e.g., the infinite alphabet

$$\{``A_0", ``A_1", ``A_2", \ldots\}$$

can replaced by the infinite set

$$\{``A0", ``A1", ``A2", \ldots\}$$

of words over the finite alphabet $\{\text{“}A\text{”},\text{“0”},\ldots,\text{“9”}\}$.

Once the informal program from above is spelled out precisely in terms of a register program, then what the existence of the resulting example decision procedure shows is that the set of primes (or rather: the set of decimal strings for primes) is decidable by means of a register machine.

Put slightly more precisely, we can define:

Definition 19 *Let $\mathcal{W}$ a set of strings (words) over $\mathcal{A}$, i.e., $\mathcal{W} \subseteq \mathcal{A}^*$. Let* P *be a register program:*

- P *register-decides $\mathcal{W}$ iff*
  for every input $\zeta \in \mathcal{A}^*$, P *eventually stops, having previously given exactly one output $\eta \in \mathcal{A}^*$, such that*
  - $\eta = \Box$, *if* $\zeta \in \mathcal{W}$
  - $\eta \neq \Box$, *if* $\zeta \notin \mathcal{W}$.

  In other words:

$$\mathsf{P} : \zeta \to \Box \quad \text{if } \zeta \in W$$
$$\mathsf{P} : \zeta \to \eta \text{ for } \eta \neq \Box \quad \text{if } \zeta \notin W$$

- *$\mathcal{W}$ is register-decidable iff there is a register program which register-decides $\mathcal{W}$.*

Using this terminology, the set of decimal representations of prime numbers is register-decidable since the informally stated procedure from above can be turned into a precisely specified register decision procedure for this set over the given alphabet $\mathcal{A} = \{\text{‘0’},\ldots,\text{‘9’}\} \cup \{\Box\}$. ($\Box$ plays the role of “Yes” in the original informal program specification.)

Remark 18 *In order to save time and space we will keep on presenting programs in this informal and somewhat “sketchy” manner, but it will always be possible to transform such informally stated programs into proper register programs.*

But there are not only procedures that decide sets of natural numbers/sets of strings but also procedures which *enumerate* sets of natural numbers/sets of strings.

Example: Here is an informal computer procedure that enumerates the set of prime numbers:

1. Start with natural number $n = 1$.
2. Test whether n is prime (as in the example before):
   (a) If the test is positive, then: Output n.
   (b) Otherwise: No output.
3. Increase n by 1 and go to line 2.

This type of procedure does not need any input. If it is started, it simply generates all prime numbers, i.e., its overall output is

$$2, 3, 5, 7, 11, 13, 17, \ldots$$

without ever terminating.

Accordingly, we define:

Definition 20 *Let $\mathcal{W}$ a set of strings (words) over $\mathcal{A}$, i.e., $\mathcal{W} \subseteq \mathcal{A}^*$. Let* P *be a register program:*

- P *register-enumerates $\mathcal{W}$ iff*
  P*, started with $\Box$, eventually yields as outputs exactly the words in $\mathcal{W}$ (in some order, possibly with repetitions).*
- *$\mathcal{W}$ is register-enumerable iff there is a register program which register-enumerates $\mathcal{W}$.*

Remark 19

- *If* P *register-enumerates an* infinite *set, then* $\mathsf{P} : \Box \to \infty$.
- *Do not mix up the notions of an (i)* enumerable set *in the sense of* countable set *and the notion of a (ii)* register-enumerable set*, i.e.,* a set enumerable by a register program*: the former is solely about the cardinality of a set, whereas the latter expresses that the members of a set can be generated by a register program in a step-by-step manner. Obviously, every register-enumerable set of words over $\mathcal{A}$ is countable, but one can show that not every countable set of words over $\mathcal{A}$ is register-enumerable.*

So the set of decimal representations of prime numbers is not only decidable but also enumerable since the informally stated procedure from above can be turned into a precisely specified register program for this set over the given alphabet $\mathcal{A} = \{$'0',...,'9'$\}$.

More interesting examples of enumerable sets are given by the following little theorems (in the proofs of which we will only sketch the corresponding enumerating register programs):

Theorem 9 *Let $\mathcal{A}$ be a finite alphabet.*
Then $\mathcal{A}^*$ is register-enumerable.

Proof. Assume $\mathcal{A} = \{a_0, \ldots, a_n\}$. The strings over $\mathcal{A}$ can be ordered as follows: for $\zeta, \zeta' \in \mathcal{A}^*$, define

$\zeta < \zeta'$ iff

- the length of ζ is less than the length of ζ'
  or
- the length of ζ is identical to the length of ζ' and
  ζ precedes ζ' lexicographically, i.e.:

  ζ is of the form $\xi a_i \eta$,
  ζ' is of the form $\xi a_j \eta'$,
  where $i < j$ (and $\xi \in \mathcal{A}^*$ or empty, $\eta, \eta' \in \mathcal{A}^*$ or empty).

It is easy to set up a procedure that enumerates the members of $\mathcal{A}^*$ according to this order. This procedure can then be turned into a proper register-program. ■

Theorem 10 *Let $\mathcal{S}$ be a finite symbol set (which determines the corresponding first-order alphabet $\mathcal{A}_\mathcal{S}$ that can be "simulated" by a finite alphabet $\mathcal{A}$ as explained above).*
Then the set of sequents that are derivable in the sequent calculus over the symbol set $\mathcal{S}$ is register-enumerable.

Proof. First of all, order the set $\mathcal{T}_\mathcal{S}$ of terms and the set $\mathcal{F}_\mathcal{S}$ of formulas as in the proof of theorem 9.
Now for $n = 1, 2, 3, \ldots$ enumerate

- the first n terms and the first n formulas according to this order,
- the finitely many sequent calculus derivations of length $\leq n$ (i) which use only these formulas and terms and (ii) which only consist of sequents containing at most n formulas as members (this can be done by a procedure).

Every derivation in the sequent calculus is reached by this enumeration for some natural number n. For every enumerated derivation, output the last sequent of the derivation. This informal procedure can be transformed into a register-program. ■

Theorem 11 *Let $\mathcal{S}$ be a finite symbol set (which determines the corresponding first-order alphabet $\mathcal{A}_{\mathcal{S}}$ that can again be "simulated" by a finite alphabet $\mathcal{A}$ as explained above).*
Then $\{\varphi \in \mathcal{F}_{\mathcal{S}} | \models \varphi\}$ is register-enumerable.

Proof. By the completeness theorem, it is sufficient to show that $\{\varphi \in \mathcal{F}_{\mathcal{S}} | \vdash \varphi\}$ is enumerable. An enumeration procedure for $\{\varphi \in \mathcal{F}_{\mathcal{S}} | \vdash \varphi\}$ can be set up in the following way:
Enumerate the set of sequents that are derivable in the sequent calculus over the symbol set $\mathcal{S}$ as explained in the proof of theorem 10: if such an enumerated sequent consists only of a single formula, output the formula. Once, again, this procedure can be turned into a register-program. ■

Finally, we can define a notion of *register-computability* for functions:

Definition 21 *Let $\mathcal{A}$ and $\mathcal{B}$ be alphabets. Let f be a function that maps every string (word) over $\mathcal{A}$ to a string (word) over $\mathcal{B}$, i.e., $f : \mathcal{A}^* \to \mathcal{B}^*$. Let* P *be a register program (over $\mathcal{A} \cup \mathcal{B}$):*

- P *register-computes f iff*
  P *computes for every input $\zeta \in \mathcal{A}^*$ exactly one output $\eta \in \mathcal{B}^*$ (and stops afterwards), such that $\eta = f(\zeta)$.*
  In other words: For all $\zeta \in \mathcal{A}^*$: $\mathsf{P} : \zeta \to f(\zeta)$
- *f is register-computable iff there is a register-program* P *which register-computes f.*

Since we will concentrate on decidability and enumerability in the following, we do not go into more details about computability.

6.4 The Relationship Between Register-Enumerability and Register-Decidability

Theorem 11 told us that the set of logically true formulas (for a given finite symbol set) is register-enumerable. Question: Is it also register-decidable?

Certainly, this does not follow from its register-enumerability in any obvious manner: let φ be an arbitrary formula (the input); now consider the enumeration procedure that we sketched in the proof of theorem 11:

- If φ is logically true, then there is computation step at which it is enumerated.
  Therefore, there is a computation step at which we will know that φ is logically true – the logical truth of φ will be positively decided.

- But if φ is *not* logically true, then the procedure will go on forever without ever enumerating φ. There will not be a computation step at which we could negatively decide the logical truth of φ (at least not by inspecting the list of formulas enumerated so far).

In fact one can show that the set of logically true formulas for a first-order language is register-enumerable but *not* register-decidable (Church 1936)!

What can be shown, however, is that every register-decidable set is register-enumerable, which follows from our next theorem:

Theorem 12 *Let $\mathcal{A}$ be an alphabet. Let $\mathcal{W} \subseteq \mathcal{A}^*$:*
$\mathcal{W}$ is register-decidable if and only if both $\mathcal{W}$ and $\mathcal{A}^* \setminus \mathcal{W}$ are register-enumerable.

Proof.
($\Rightarrow$) Assume $\mathcal{W}$ is register-decidable:

- This implies that $\mathcal{W}$ is register-enumerable, because:
  By the register-decidability of $\mathcal{W}$, there is a decision register program P for $\mathcal{W}$. From P we can set up an enumeration procedure P' for $\mathcal{W}$: (i) let P' register-enumerate the strings of $\mathcal{A}^*$ according to the order that we defined in the proof of theorem 9; (ii) for each enumerated word let P' apply P to decide whether this word is in $\mathcal{W}$: if yes, output the word (otherwise let the program do nothing).

- Furthermore, $\mathcal{A}^* \setminus \mathcal{W}$ is register-enumerable:
  As before there is by assumption a decision register program P for $\mathcal{W}$. But from P we can easily construct a decision procedure P' for $\mathcal{A}^* \setminus \mathcal{W}$: simply let P' be like P except that whenever P is defined to yield output $\square$ then P' is chosen to have an output different from $\square$, while whenever P is defined to output a string distinct from $\square$ then P' is chosen to have $\square$ as its output. Hence, $\mathcal{A}^* \setminus \mathcal{W}$ is register-decidable. By the same reasoning as for $\mathcal{W}$, this implies that $\mathcal{A}^* \setminus \mathcal{W}$ is register-enumerable.

($\Leftarrow$) Assume $\mathcal{W}$ and $\mathcal{A}^* \setminus \mathcal{W}$ are register-enumerable: So there are enumeration register programs P and P' for $\mathcal{W}$ and $\mathcal{A}^* \setminus \mathcal{W}$, respectively. We can combine P and P' in order to determine a decision register program P'' for $\mathcal{W}$: let ζ be an arbitrary input; let P'' run P and P' alternately in a step-by-step manner: first step of P, first step of P', second step of P, second step of P',... Eventually, since either $\zeta \in \mathcal{W}$ or $\zeta \in \mathcal{A}^* \setminus \mathcal{W}$, either P or P' must enumerate ζ: if P enumerates ζ then let the output of P'' be $\square$; if P' enumerates ζ then let P'' output any string distinct from $\square$. ■

Remark 20 *Register-decidability and register-enumerability were defined with an implicit reference to an underlying alphabet $\mathcal{A}$. However, it is easy to see that register-decidability and register-enumerability are not really sensitive to the specific choice of this alphabet: consider alphabets $\mathcal{A}_1$, $\mathcal{A}_2$, such that $\mathcal{A}_1 \subseteq \mathcal{A}_2$, and assume $\mathcal{W} \subseteq \mathcal{A}_1^*$; then it holds that $\mathcal{W}$ is register-decidable/register-enumerable with respect to $\mathcal{A}_1$ iff $\mathcal{W}$ is register-decidable/register-enumerable with respect to $\mathcal{A}_2$.*

6.5 First-Order Theories and Axiomatizability

In the following we will presuppose symbol sets $\mathcal{S}$ that are finite (actually, it would suffice to focus on symbol sets that are *register-decidable* over a given finite alphabet).
Here is what we want to understand by a *theory*:

Definition 22 *$\Phi \subseteq \mathcal{F}_{\mathcal{S}}$ is a theory iff*

- *Φ is a set of sentences,*
- *Φ is satisfiable,*
- *Φ is closed under logical consequence, i.e., for all sentences $\varphi \in \mathcal{F}_{\mathcal{S}}$: if $\Phi \models \varphi$ then $\varphi \in \Phi$.*

(Actually, we should speak of an $\mathcal{S}$-theory, but as usual we will often suppress the reference to a symbol set.)

Models determine theories in the following sense:

Example 24 *For every $\mathcal{S}$-model $\mathfrak{M}$ the set*

$$Th(\mathfrak{M}) = \{\varphi \in \mathcal{F}_{\mathcal{S}} | \varphi \text{ is a sentence}, \mathfrak{M} \models \varphi\}$$

is a theory – the theory of the model $\mathfrak{M}$.

In particular, reconsider first-order arithmetic:

Let $\mathcal{S}_{\text{arith}} = \{\overline{0},\ \overline{1},\ \overline{+},\ \overline{\cdot}\}$;

standard model of arithmetic: $\mathfrak{M}_{arith} = (\mathbb{N}_0, \mathfrak{I})$ with $\mathfrak{I}$ as expected (so $\mathfrak{I}(\overline{0}) = 0, \mathfrak{I}(\overline{+}) = + \text{ on } \mathbb{N}_0, \ldots$).

Remark: From now on we will omit bars over signs again!

We once called Φ_{arith} the set of $\mathcal{S}_{\text{arith}}$-sentences that are satisfied by this model, i.e.:

$$\underbrace{\Phi_{\text{arith}}}_{\text{“arithmetic”}} = \{\varphi \in \mathcal{F}_{\mathcal{S}_{\text{arith}}} | \varphi \text{ sentence}, (\mathbb{N}_0, \mathfrak{I}) \vDash \varphi\}$$

In the terminology from above: $\Phi_{\text{arith}} = Th(\mathfrak{M}_{arith})$.

A different way of determining a theory is by means of a set of formulas:

Example 25 *Let Φ be a set of sentences over a symbol set $\mathcal{S}$.*

We define:

$$\Phi^{\models} = \{\varphi \in \mathcal{F}_{\mathcal{S}} | \varphi \text{ is a sentence}, \Phi \models \varphi\}$$

Obviously, by the definitions above it holds that:

- *If $\mathcal{T}$ is a theory, then $\mathcal{T}^{\models} = \mathcal{T}$.*
- *If Φ is a satisfiable set of $\mathcal{S}$-sentences, then $\Phi^{\models}$ is a theory.*

So theories can also be generated on the basis of satisfiable sets of sentence by means of logical consequence.

In particular, consider the following theory $\mathcal{T}_{PA}$ which is called (first-order) "Peano arithmetic":

Let $\mathcal{T}_{PA} = \Phi_{PA}^{\models}$, where Φ_{PA} is the following (infinite) set of sentences:

1. $\forall x \neg x + 1 \equiv 0$
2. $\forall x \forall y (x + 1 \equiv y + 1 \to x \equiv y)$
3. $\forall x \, x + 0 \equiv x$
4. $\forall x \forall y \, x + (y + 1) \equiv (x + y) + 1$
5. $\forall x \, x \cdot 0 \equiv 0$
6. $\forall x \forall y \, x \cdot (y + 1) \equiv x \cdot y + x$
7. *Induction:*

$$\forall x_0 \ldots \forall x_{n-1} \left(\left(\varphi \frac{0}{y} \wedge \forall y \left(\varphi \to \varphi \frac{y+1}{y} \right) \right) \to \forall y \, \varphi \right)$$

(for all variables $x_0, \ldots, x_{n-1}, y$, for all $\varphi \in \mathcal{F}_{\mathcal{S}}$ with $free(\varphi) \subseteq \{x_0, \ldots, x_{n-1}, y\}$)

Since $\mathfrak{M}_{arith}$ is a model for Φ_{PA}, Φ_{PA} is satisfiable and thus $\mathcal{T}_{PA}$ is a theory.

Many theorems in number theory can actually be derived from Φ_{PA}, i.e., are members of first-order Peano arithmetic $\mathcal{T}_{PA}$. As we will see, it follows nevertheless from Gödel's Incompleteness Theorems that

$$\mathcal{T}_{PA} \subsetneqq Th(\mathfrak{M}_{arith}) = \Phi_{\text{arith}}$$

So why should we be interested in theories such as $\mathcal{T}_{PA}$ at all? Because they are *axiomatizable*:

Definition 23

- *A theory $\mathcal{T}$ is axiomatizable iff there there is a register-decidable set Φ of $\mathcal{S}$-sentences such that $\mathcal{T} = \Phi^{\models}$.*

- *A theory $\mathcal{T}$ is finitely axiomatizable iff there there is a finite set Φ of $\mathcal{S}$-sentences such that $\mathcal{T} = \Phi^{\models}$.*

So e.g. the set of logically true $\mathcal{S}$-sentences is finitely axiomatizable (for $\Phi = \varnothing$). $\mathcal{T}_{PA}$ is axiomatizable (but one can show that it is not *finitely* axiomatizable).

Axiomatizable theories are important because they can be generated by a computer program:

Theorem 13 *Every axiomatizable theory is register-enumerable.*

Proof. Let $\mathcal{T}$ be a theory and let Φ be a register-decidable set of $\mathcal{S}$-sentences such that $\mathcal{T} = \Phi^{\models}$. The sentences of $\mathcal{T}$ can be register-enumerated as follows: let a register program generate systematically (as sketched in section 6.3) all sequents that are derivable in the sequent calculus and let the program check in each case whether all members of the antecedent of the generated sequent belong to Φ; this can be done by means of a register-decision program for Φ which exists by assumption. If all members of the antecedent belong to Φ, let the program check whether the consequent of the sequent is a sentence (this can obviously be decided as well): if yes, let the register program output the consequent of the sequent. ■
An axiomatizable theory is not necessarily register-decidable (the set of logically true $\mathcal{S}$-sentences is a counterexample; compare the last section). For special theories, however, the situation is different:

Definition 24 *A theory $\mathcal{T}$ (of $\mathcal{S}$-sentences) is complete iff for every $\mathcal{S}$-sentence: $\varphi \in \mathcal{T}$ or $\neg\varphi \in \mathcal{T}$.*

Obviously, every theory of the form $Th(\mathfrak{M})$ is complete (for arbitrary models $\mathfrak{M}$). Note that complete theories are still *theories* and thus cannot contain both φ and $\neg\varphi$ for any sentence φ, for otherwise they would not be satisfiable.

Complete theories have the following nice property with regard to register-decidability:

Theorem 14

1. *Every axiomatizable and complete theory is register-decidable.*

2. *Every enumerable and complete theory is register-decidable.*

Proof. By theorem 13 it is sufficient to prove 2. So let $\mathcal{T}$ be a register-enumerable complete theory. A decision register program for $\mathcal{T}$ can be set up as follows: some $\mathcal{S}$-string φ is given as an input. At first the program decides whether φ is an $\mathcal{S}$-sentence (the set of $\mathcal{S}$-sentences is of course register-decidable). If yes, let the program enumerate the members of $\mathcal{T}$ (such an enumeration procedure exists by assumption). Since $\mathcal{T}$ is complete, eventually either φ or $\neg\varphi$ is enumerated: in the first case let the procedure output $\square$, in the second case any string distinct from $\square$. ■

6.6 Arithmetical Representability and the Incompleteness Theorems

In the following, let Φ be a set of $\mathcal{S}_{\text{arith}}$-sentences, i.e., sentences of first-order arithmetic. On the basis of $\mathcal{S}_{\text{arith}} = \{\overline{0},\ \overline{1},\ \overline{+},\ \overline{\cdot}\}$ we can build up $\mathcal{S}_{\text{arith}}$-terms that can be used as standard names (numerals) for natural numbers: $\overline{0}, \overline{1}, (\overline{1}\overline{+}\overline{1}), (\overline{1}\overline{+}\overline{1})\overline{+}\overline{1}, \ldots$ Let us abbreviate the standard name for the natural number n by means of $\overline{n}$ (so $\overline{n}$ is an $\mathcal{S}_{\text{arith}}$-term that denotes n according to the standard interpretation of $\mathcal{S}_{\text{arith}}$).

In certain cases, a set Φ may be shown to "know" something about particular relations or functions of natural numbers in the sense that facts about these relations or functions are represented in Φ by means of formulas:

Definition 25

- *A relation $R \subseteq \mathbb{N}_0^r$ is representable in Φ iff there is an $\mathcal{S}_{\text{arith}}$-formula φ (the free variables of which are among $\{v_0, \ldots, v_{r-1}\}$) such that for all $n_0, \ldots, n_{r-1} \in \mathbb{N}_0$:*

  1. *if $R(n_0, \ldots, n_{r-1})$ then $\Phi \vdash \varphi\frac{\overline{n_0}, \ldots, \overline{n_{r-1}}}{v_0, \ldots, v_{r-1}}$*
  2. *if not $R(n_0, \ldots, n_{r-1})$ then $\Phi \vdash \neg\varphi\frac{\overline{n_0}, \ldots, \overline{n_{r-1}}}{v_0, \ldots, v_{r-1}}$*

  (we also say that in such a case φ represents R in Φ).

- *A function $F : \mathbb{N}_0^r \to \mathbb{N}_0$ is representable in Φ iff there is an $\mathcal{S}_{\text{arith}}$-formula φ (the free variables of which are among $\{v_0, \ldots, v_r\}$) such that for all $n_0, \ldots, n_r \in \mathbb{N}_0$:*

  1. *if $F(n_0, \ldots, n_{r-1}) = n_r$ then $\Phi \vdash \varphi\frac{\overline{n_0}, \ldots, \overline{n_r}}{v_0, \ldots, v_r}$*
  2. *if $F(n_0, \ldots, n_{r-1}) \neq n_r$ then $\Phi \vdash \neg\varphi\frac{\overline{n_0}, \ldots, \overline{n_r}}{v_0, \ldots, v_r}$*
  3. *$\Phi \vdash \exists! v_r \varphi\frac{\overline{n_0}, \ldots, \overline{n_{r-1}}}{v_0, \ldots, v_{r-1}}$*

  (we also say that in such a case φ represents F in Φ).

In some cases a set Φ may be shown to "know" something about procedures and computation in the sense that all register-decidable relations and all register-computable functions on $\mathbb{N}_0$ are represented in Φ by means of formulas. It is useful to introduce an abbreviation for this type of property of a set Φ of arithmetical sentences:

Definition 26

Repr Φ iff all register-decidable relations $R \subseteq \mathbb{N}_0^r$ (for $r = 1, 2, \ldots$) and all register-computable functions $F : \mathbb{N}_0^r \to \mathbb{N}_0$ are representable in Φ.

Remark: The register-decidability or register-enumerability of a relation $R \subseteq \mathbb{N}_0^r$ for $r > 1$ and the register-computability of a function $F : \mathbb{N}_0^r \to \mathbb{N}_0$ for $r > 1$ is defined analogously to our definitions of register-decidability, register-enumerability, and register-computability for *sets* of signs or natural numbers in section 6.3.

What examples of sets of sentences that have this property *Repr* do we know? Here are two important ones:

Theorem 15

- *Repr $Th(\mathfrak{M}_{arith})$, i.e., the set of true arithmetical first-order sentences has the property Repr.*

- *Repr $\mathcal{T}_{PA}$, i.e., the set of arithmetical sentences derivable in first-order Peano arithmetic has the property Repr.*

Proof. Without proof. (Note that the proof is highly non-trivial and needs a lot of work – a big part of Gödel's proof is devoted to these matters.) ■

For all further considerations we fix a register-computable coding of $\mathcal{S}_{\text{arith}}$-formulas by natural numbers (a Gödel numbering) such that every number is the Gödel number of a (unique) formula. So we can write

- g^φ for the Gödel number of φ (hence $g^\varphi \in \mathbb{N}_0$)

Via this encoding it is possible to translate statements about formulas into statements about natural numbers and vice versa. Some statements about natural numbers even turn out to be equivalent to statements about *their own* Gödel numbers – these sentences speak about their codes and thus, in some sense, about themselves. Indeed one can show that there are many such sentences:

Lemma 18 *(Fixed Point Lemma)*
Assume that Repr Φ*: Then for every* $\mathcal{S}_{\text{arith}}$*-formula* ψ *in which precisely the variable* v_0 *occurs freely there is an* $\mathcal{S}_{\text{arith}}$*-sentence* φ *(*$= \varphi_\psi$*) such that*

$$\Phi \vdash \varphi \leftrightarrow \psi \frac{\overline{g^\varphi}}{v_0}$$

(hence, up to provable equivalence, φ *expresses about "itself": "my" code has the property expressed by* ψ*;* $\overline{g^\varphi}$ *is the numeral of the code of* φ*).*

Proof.
Let $F : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ be defined as follows: if n is the Gödel number of some arithmetical formula χ in which precisely the variable v_0 occurs freely, then let $F(n,m) = g^{\chi \frac{\overline{m}}{v_0}}$; otherwise, let $F(n,m) = 0$.
Since we presuppose a register-computable coding function, the function F itself can easily be seen to be register-computable as well. Furthermore, for every $\mathcal{S}_{\text{arith}}$-formula χ in which precisely the variable v_0 occurs freely it follows that:

$$F(g^\chi, m) = g^{\chi \frac{\overline{m}}{v_0}}$$

Since *Repr* Φ, this function F can be represented in Φ by a formula α (the free variables of which are among $\{v_0, v_1, v_2\}$, where v_0 and v_1 stand for the two arguments of F and v_2 stands for the corresponding function value of F).
Now let ψ be given with precisely v_0 free in it. We introduce the following abbreviation: let

- $\beta = \forall v_2(\alpha\frac{v_0,v_0,v_2}{v_0,v_1,v_2} \to \psi\frac{v_2}{v_0})$
- $\varphi = \forall v_2(\alpha\frac{\overline{g^\beta},\overline{g^\beta},v_2}{v_0,v_1,v_2} \to \psi\frac{v_2}{v_0})$

Since β is an arithmetical formula in which precisely v_0 is free and since $\varphi = \beta\frac{\overline{g^\beta}}{v_0}$, it follows that $F(g^\beta, g^\beta) = g^\varphi$ and therefore by the representation of F in terms of α:

$$\Phi \vdash \alpha\frac{\overline{g^\beta},\overline{g^\beta},\overline{g^\varphi}}{v_0,v_1,v_2}$$

Now we can finally show that $\Phi \vdash \varphi \leftrightarrow \psi\frac{\overline{g^\varphi}}{v_0}$:

1. By definition of φ,

   $$\Phi \cup \{\varphi\} \vdash \alpha\frac{\overline{g^\beta},\overline{g^\beta},\overline{g^\varphi}}{v_0,v_1,v_2} \to \psi\frac{\overline{g^\varphi}}{v_0}$$

   Since we already know that the antecedent of this implication formula is derivable from Φ, it follows that

   $$\Phi \cup \{\varphi\} \vdash \psi\frac{\overline{g^\varphi}}{v_0}$$

   But by the sequent calculus this implies that $\Phi \vdash \varphi \to \psi\frac{\overline{g^\varphi}}{v_0}$.

2. Because F is represented in Φ by α, it must hold that

   $$\Phi \vdash \exists! v_2\alpha\frac{\overline{g^\beta},\overline{g^\beta},v_2}{v_0,v_1,v_2}$$

   We aready know that $\Phi \vdash \alpha\frac{\overline{g^\beta},\overline{g^\beta},\overline{g^\varphi}}{v_0,v_1,v_2}$, so it follows that

   $$\Phi \vdash \forall v_2(\alpha\frac{\overline{g^\beta},\overline{g^\beta},v_2}{v_0,v_1,v_2} \to v_2 \equiv \overline{g^\varphi})$$

   which entails (by the equality in the "then" part) that

   $$\Phi \vdash \psi\frac{\overline{g^\varphi}}{v_0} \to \forall v_2(\alpha\frac{\overline{g^\beta},\overline{g^\beta},v_2}{v_0,v_1,v_2} \to \psi\frac{v_2}{v_0})$$

   But this is just

   $$\Phi \vdash \psi\frac{\overline{g^\varphi}}{v_0} \to \varphi$$

So we have implications in both directions and thus we are done. ■

This fixed point lemma has grave consequences:

Lemma 19
Assume that Repr Φ *and let* Φ *be a consistent set of arithmetical sentences: If the set of Gödel numbers of sentences in* $\Phi^{\vdash}(= \Phi^{\models})$ *is representable in* Φ *(briefly: if* $\Phi^{\vdash}$ *is representable in* Φ*), then there is an* $\mathcal{S}_{\mathrm{arith}}$*-sentence* φ *such that neither* $\Phi \vdash \varphi$ *nor* $\Phi \vdash \neg\varphi$*.*

(For the definition of $\Phi^{\models}$ *see the last section; note that by soundness and completeness we can write* $\Phi^{\vdash}$ *instead of* $\Phi^{\models}$*.)*

Proof. Suppose χ (in which precisely v_0 is free) represents the set $\Phi^{\vdash}$ in Φ. Then it follows that for arbitrary arithmetical sentences α:

1. if g^{α} is a member of the set of codes of sentences in $\Phi^{\vdash}$, then $\Phi \vdash \chi\frac{\overline{g^{\alpha}}}{v_0}$
2. if g^{α} is not a member of the set of codes of sentences in $\Phi^{\vdash}$, then $\Phi \vdash \neg\chi\frac{\overline{g^{\alpha}}}{v_0}$.

By the consistency of Φ this entails

$$\Phi \vdash \chi\frac{\overline{g^{\alpha}}}{v_0} \text{ iff } \Phi \vdash \alpha$$

Now let $\psi = \neg\chi$: by lemma 18 there is a "fixed point sentence" φ for ψ such that

$$\Phi \vdash \varphi \leftrightarrow \neg\chi\frac{\overline{g^{\varphi}}}{v_0}$$

(so φ expresses: "my" code is not a member of the codes of sentences in $\Phi^{\vdash}$, i.e., "I" am not derivable from Φ).
But now we can conclude:

- If $\Phi \vdash \varphi$, then $\Phi \vdash \chi\frac{\overline{g^{\varphi}}}{v_0}$ and hence by the fixed point property (and applying negation) $\Phi \vdash \neg\varphi$, contradicting the consistency of Φ.
- If $\Phi \vdash \neg\varphi$, then by the fixed point property (and applying negation) $\Phi \vdash \chi\frac{\overline{g^{\varphi}}}{v_0}$ and thus $\Phi \vdash \varphi$, again contradicting the consistency of Φ.

So neither $\Phi \vdash \varphi$ nor $\Phi \vdash \neg\varphi$. ■
But now we can finally put things together in order to derive:

Theorem 16 *(Gödel's First Incompleteness Theorem)*

1. *Let Φ be a consistent and register-decidable set of arithmetical sentences for which it is the case that Repr Φ: then there is an $\mathcal{S}_{\text{arith}}$-sentence φ such that neither $\Phi \vdash \varphi$ nor $\Phi \vdash \neg\varphi$.*

2. *$Th(\mathfrak{M}_{arith})$ is not axiomatizable.*

Proof.

1. Let Φ be described above and assume that for every $\mathcal{S}_{\text{arith}}$-sentence φ either $\Phi \vdash \varphi$ or $\Phi \vdash \neg\varphi$. So by section 6.5, $\Phi^{\vdash} (= \Phi^{\models})$ is a complete axiomatizable theory which, by theorem 14, is register-decidable. Hence, by *Repr* Φ, the set (of codes of members of) $\Phi^{\vdash}$ is representable in Φ, which contradicts lemma 19.

2. If $Th(\mathfrak{M}_{arith})$ were axiomatizable, then it would be a complete axiomatizable theory, which, by the same argument as before, would be register-decidable. Moreover, $Th(\mathfrak{M}_{arith})$ has the property *Repr*. But as we have just seen this would contradict lemma 19.

■

Remark: Actually, this is a *version* of Gödel's First Incompleteness Theorem – Gödel's original First Incompleteness Theorem is slightly stronger and uses more "fine-grained" assumptions.

Gödel's Second Incompleteness Theorem extends this result by showing that consistent axiomatizable theories that contain "enough" arithmetic (e.g., first-order Peano arithmetic) cannot prove their own consistency, where the corresponding consistency statement can be expressed as an arithmetical sentence that speaks about the codes of arithmetical sentences.

That's it... hope you liked it!! (Stay logical!)

7 Solutions to the Problem Sets

7.1 Solutions to Problem Set 1

1. (a) Show (this is a recapitulation of something you should know about countable sets):

   If the sets $M_0, M_1, M_2, \ldots$ are countable,
   then $\bigcup_{n\in\mathbb{N}} M_n$ is countable as well.

   Proof.
   Without restriction of generality, we may assume that $M_n \neq \varnothing$ for all $n \in \mathbb{N} = \{0, 1, 2, \ldots\}$ (otherwise simply omit all empty sets M_n from our countable sequence of sets and reenumerate them). Furthermore, we may assume that each set M_n is of the form $\{a_0^n, a_1^n, a_2^n, \ldots\}$.
   Now we can think of the sets M_n as being listed as rows of an array of the following kind:

   $$\begin{array}{llllll} M_0: & a_0^0 & a_1^0 & a_2^0 & \ldots \\ M_1: & a_0^1 & a_1^1 & a_2^1 & \ldots \\ M_2: & a_0^2 & a_1^2 & a_2^2 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \end{array}$$

   The entries of this array can be enumerated in the following "diagonal" manner: $(1.)a_0^0, (2.)a_1^0, (3.)a_0^1, (4.)a_0^2, (5.)a_1^1, (6.)a_2^0, (7.)a_3^0, \ldots$
   Since every member of $\bigcup_{n\in\mathbb{N}} M_n$ occurs in this array, this proves that there is an onto mapping from $\mathbb{N}$ to $\bigcup_{n\in\mathbb{N}} M_n$. Therefore, $\bigcup_{n\in\mathbb{N}} M_n$ is countable (compare p. 8 in the lecture notes). ■

   (b) Prove the following lemma by means of 1a:

   If $\mathcal{A}$ is a countable alphabet, then the set $\mathcal{A}^*$ (of finite strings over $\mathcal{A}$) is countable, too.

   Proof.
   $\mathcal{A}$ is of the form $\{a_0, a_1, a_2, \ldots\}$. Since $\mathcal{A}^*$ is the set of strings over $\mathcal{A}$ with finite length $n = 1, 2, 3, \ldots$, we can regard $\mathcal{A}^*$ as the union $\bigcup_n \mathcal{A}^n = \mathcal{A}^1 \cup \mathcal{A}^2 \cup \mathcal{A}^3 \cup \ldots$
   Each of the sets $\mathcal{A}^n$ is countable, because: $\mathcal{A}$ is countable by assumption. If $\mathcal{A}^n$ is countable, then of course also $\{a_k\} \times \mathcal{A}^n$ is countable for arbitrary $k = 0, 1, 2, \ldots$, and since $\mathcal{A}^{n+1} = \bigcup_{k\in\mathbb{N}}(\{a_k\}$

$\times \mathcal{A}^n$) it follows from 1a that also $\mathcal{A}^{n+1}$ is countable. Hence, by induction over n, each set $\mathcal{A}^n$ is countable.
Thus, by 1a again, since every $\mathcal{A}^n$ is countable, their union $\bigcup_{n\in\mathbb{N}} \mathcal{A}^n$ is countable and we are done. ■

2. Let $\mathcal{S}$ be an arbitrary symbol set. We consider the following calculus $\mathcal{C}$ of rules:

   - $\dfrac{}{x \;\; x}$

   (for arbitrary variables x)

   - $\dfrac{x \;\; t_i}{x \;\; f(t_1, \ldots, t_n)}$

   (for arbitrary variables x, for arbitrary $\mathcal{S}$-terms $t_1, \ldots, t_n$, for arbitrary n-ary function signs $f \in \mathcal{S}$, for arbitrary $i \in \{1, \ldots, n\}$).

   Show that for all variables x and all $\mathcal{S}$-terms t holds: The string

   $$x \;\; t$$

   is derivable in $\mathcal{C}$ if and only if $x \in var(t)$ (i.e., x is a variable in t).
   Proof. ($\Rightarrow$) Let x be an arbitrary variable. We show that if the string $x \;\; t$ is derivable on the basis of the rules of $\mathcal{C}$, then x is a variable in t. This is proven by induction over the strings $x \;\; t$ that can be derived in $\mathcal{C}$ (the property P that we prove such strings $x \;\; t$ to have is in this case: *the variable x occurs somewhere in t, i.e., the string before the blank occurs somewhere in the string after the blank*):
   Induction basis:
   All strings $x \;\; t$ with $\frac{}{x \;\; t}$ have the property P. This is because the only such strings are of the form $x \;\; x$ and x occurs in x.
   Induction step:
   Assume the string $x \;\; t_i$ has the property P, i.e., x occurs in the $\mathcal{S}$-term t_i. But then x certainly also occurs in the $\mathcal{S}$-term $f(t_1, \ldots, t_n)$, since t_i is a substring of $f(t_1, \ldots, t_n)$ and x is a substring of t_i by the inductive assumption. So $f(t_1, \ldots, t_n)$ has the propery P as well.

(⇐) We fix an arbitrary variable x and prove by induction over $\mathcal{S}$-terms t that if x occurs in t then the string $x\ \ t$ is derivable by means of the rules of $\mathcal{C}$ (so the property P that we prove terms t to have is in this case: *if x occurs in t, then the string $x\ \ t$ is derivable in the calculus $\mathcal{C}$ of rules*):
Induction basis:
Variables have the property P: if x occurs in t, (i) then in the case where $t = x$, it is indeed the case that the string $x\ \ x$ is derivable in $\mathcal{C}$, (ii) while the other case, i.e., where t is a variable different from x, is excluded, since x does not occur in any variable different from x.
Constants satisfy the property P vacuously (x does not occur in any constant t).
Induction step:
Assume $\mathcal{S}$-terms $t_1, \ldots, t_n$ have the property P.
So if x occurs in t_1 then the string $x\ \ t_1$ is derivable in the calculus $\mathcal{C}$, and the same holds for $t_2, \ldots, t_n$. Now consider any string of the form $f(t_1, \ldots, t_n)$ where f is an arbitrary n-ary function sign in $\mathcal{S}$: if x occurs in $f(t_1, \ldots, t_n)$, then it must occur in one of the terms $t_1, \ldots, t_n$, say in t_i: by the inductive assumption, it follows that the string $x\ \ t_i$ is derivable in $\mathcal{C}$. But then the second rule of $\mathcal{C}$ can be used to derive the string $x\ \ f(t_1, \ldots, t_n)$. So we have shown that if x occurs in $f(t_1, \ldots, t_n)$, then $x\ \ f(t_1, \ldots, t_n)$ is derivable in $\mathcal{C}$. This means that $f(t_1, \ldots, t_n)$ has the property P. ■

3. Prove that the following strings are $\mathcal{S}$-terms (for given $\mathcal{S}$ with $c, f, g \in \mathcal{S}$, where f is a binary function sign, g is a unary function sign, x and y are variables):

   (a) $f(x, c)$:

(6)	x	(T1)
(7)	c	(T2)
(8)	$f(x, c)$	(T3, with 1., 2.)

   (b) $g(f(x, c))$:

(1)	$f(x, c)$	(see 3a)
(2)	$g(f(x, c))$	(T3, with 1.)

(c) $f(f(x,c), f(x, f(x,y)))$:

(1)	$f(x,c)$	(see 3a)
(2)	x	(T1)
(3)	y	(T1)
(4)	$f(x,y)$	(T3, with 2., 3.)
(5)	$f(x, f(x,y))$	(T3, with 2.,4.)
(6)	$f(f(x,c), f(x, f(x,y)))$	(T3, with 1.,5.)

4. Prove that the following strings are $\mathcal{S}$-formulas (with x, y, c, f, g as in 3 and where P, $Q \in \mathcal{S}$, such that P is a unary predicate and Q is a binary predicate):

(a) $\neg P(f(x,c))$:

(1)	$P(f(x,c))$	(F2; use 3a)
(2)	$\neg P(f(x,c))$	(F3, with 1.)

(b) $\exists x \forall y (P(g(f(x,c))) \to Q(y,y))$:

(1)	$P(g(f(x,c)))$	(F2; use 3b)
(2)	$Q(y,y)$	(F2)
(3)	$(P(g(f(x,c))) \to Q(y,y))$	(F4, with 1., 2.)
(4)	$\forall y (P(g(f(x,c))) \to Q(y,y))$	(F5, with 3.)
(5)	$\exists x \forall y (P(g(f(x,c))) \to Q(y,y))$	(F5, with 4.)

(c) $(\forall x \neg P(f(x,c)) \lor Q(f(x,c), f(f(x,c), f(x, f(x,y)))))$:

(1)	$\neg P(f(x,c))$	(use 4a)
(2)	$\forall x \neg P(f(x,c))$	(F5, with 1.)
(3)	$Q(f(x,c), f(f(x,c), f(x, f(x,y))))$	(F2; use 3a, 3c)
(4) $(\forall x \neg P(f(x,c)) \lor Q(f(x,c), f(f(x,c), f(x, f(x,y)))))$		(F4, with 2., 3.)

5. Prove by induction: the string $\forall x f(x,c)$ is not an $\mathcal{S}$-term (where $\mathcal{S}$ is an arbitrary symbol set).

Proof. We prove this (as I told you rather trivial) statement by induction over $\mathcal{S}$-terms t (the property P which we show terms have is in this case: *t is different from the string* $\forall x f(x, c)$):
Induction basis:
Every variable and every $\mathcal{S}$-constant is certainly different from $\forall x f(x, c)$.
Induction step:
Assume $\mathcal{S}$-terms $t_1, \ldots, t_n$ are different from $\forall x f(x, c)$.
It is certainly the case that $f(t_1, \ldots, t_n)$ is different from $\forall x f(x, c)$, since the former begins with a function sign while the latter begins with a quantifier.
(The triviality of the result shows up in the way that *we did not even have to use the inductive assumption* in order to prove that $f(t_1, \ldots, t_n)$ differs from $\forall x f(x, c)$.) ■

6. Let x, y, z be variables, $f \in \mathcal{S}$ a unary function sign, P, Q, $R \in \mathcal{S}$ with P being a binary predicate, Q a unary predicate, and R a ternary predicate. Determine for the following $\mathcal{S}$-formulas φ the corresponding set of variables that occur freely in φ (i.e., the sets $free(\varphi)$):
(We state 6a in detail but for 6b and 6c only the final solutions.)

   (a) $\forall x \exists y (P(x, z) \to \neg Q(y)) \to \neg Q(y)$:
   $free(\forall x \exists y (P(x, z) \to \neg Q(y)) \to \neg Q(y)) =$
   $free(\forall x \exists y (P(x, z) \to \neg Q(y))) \cup free(\neg Q(y)) =$
   $[free(\exists y (P(x, z) \to \neg Q(y))) \setminus \{x\}] \cup free(Q(y)) =$
   $[[free((P(x, z) \to \neg Q(y))) \setminus \{y\}] \setminus \{x\}] \cup \{y\} =$
   $[free((P(x, z) \to \neg Q(y))) \setminus \{x, y\}] \cup \{y\} =$
   $[[free(P(x, z)) \cup free(\neg Q(y))] \setminus \{x, y\}] \cup \{y\} =$
   $[[\{x, z\} \cup free(Q(y))] \setminus \{x, y\}] \cup \{y\} =$
   $[[\{x, z\} \cup \{y\}] \setminus \{x, y\}] \cup \{y\} =$
   $[\{x, y, z\} \setminus \{x, y\}] \cup \{y\} =$
   $\{z\} \cup \{y\} =$
   $\{y, z\}$ (so this formula is not a sentence)

   (b) $\forall x \forall y (Q(c) \land Q(f(x))) \to \forall y \forall x (Q(y) \land R(x, x, y))$:
   $free(\forall x \forall y (Q(c) \land Q(f(x))) \to \forall y \forall x (Q(y) \land R(x, x, y))) = \varnothing$
   (so this formula is a sentence)

   (c) $Q(z) \leftrightarrow \exists z (P(x, y) \land R(c, x, y))$:
   $free(Q(z) \leftrightarrow \exists z (P(x, y) \land R(c, x, y))) = \{x, y, z\}$
   (so this formula is not a sentence)

7.2 Solutions to Problem Set 2

1. Let $\mathcal{S} = \{P, R, f, g, c_0, c_1\}$, where P is a unary predicate, R is a binary predicate, and f and g are binary function signs. Let $\mathfrak{M} = (D, \mathfrak{I})$ be an $\mathcal{S}$-model with $D = \mathbb{R}$, such that $\mathfrak{I}(P) = \mathbb{N}$, $\mathfrak{I}(R)$ is the "larger than" ($>$) relation on $\mathbb{R}$, $\mathfrak{I}(f)$ is the addition mapping on $\mathbb{R}$, $\mathfrak{I}(g)$ is the multiplication mapping on $\mathbb{R}$, $\mathfrak{I}(c_0) = 0$, and $\mathfrak{I}(c_1) = 1$. Finally, let s be a variable assignment over $\mathfrak{M}$ with the property that $s(x) = 5$ and $s(y) = 3$ (where x, y, and z from below, are fixed pairwise distinct variables).

   Determine the following semantic values by step-by-step application of the definition clauses for $Val_{\mathfrak{M},s}$; subsequently, translate the terms/formulas into our usual mathematical "everyday" language:

   (a) $Val_{\mathfrak{M},s}(g(x, f(y, c_1)))$:

   $Val_{\mathfrak{M},s}(g(x, f(y, c_1))) = 5 \cdot (3 + 1) = 20$

   (b) $Val_{\mathfrak{M},s}(f(g(x, y), g(x, c_1)))$:

   $Val_{\mathfrak{M},s}(f(g(x, y), g(x, c_1))) = 5 \cdot 3 + 5 \cdot 1 = 20$

   (c) $Val_{\mathfrak{M},s}(\forall x \forall y(R(x, c_0) \rightarrow \exists z(P(z) \land R(g(z, x), y))))$:

   Here we take a more detailed look:

   $Val_{\mathfrak{M},s}(\forall x \forall y(R(x, c_0) \rightarrow \exists z(P(z) \land R(g(z, x), y)))) = 1$ iff

   for all $d \in D$:
   $Val_{\mathfrak{M},s\frac{d}{x}}(\forall y(R(x, c_0) \rightarrow \exists z(P(z) \land R(g(z, x), y)))) = 1$ iff

   for all $d \in D$, for all $d' \in D$:
   $Val_{\mathfrak{M},s\frac{d}{x}\frac{d'}{y}}(R(x, c_0) \rightarrow \exists z(P(z) \land R(g(z, x), y))) = 1$ iff

   for all $d \in D$, for all $d' \in D$:
   $Val_{\mathfrak{M},s\frac{d}{x}\frac{d'}{y}}(R(x, c_0)) = 0$ or
   $Val_{\mathfrak{M},s\frac{d}{x}\frac{d'}{y}}(\exists z(P(z) \land R(g(z, x), y))) = 1$ iff

for all $d \in D$, for all $d' \in D$:
it is not the case that $(s\frac{d}{x}\frac{d'}{y}(x), \mathfrak{I}(c_0)) \in \mathfrak{I}(R)$ or
there is a $d'' \in D$, s.t. $Val_{\mathfrak{M}, s\frac{d}{x}\frac{d'}{y}\frac{d''}{z}}(P(z) \wedge R(g(z,x),y)) = 1$ iff

for all $d \in D$, for all $d' \in D$:
it is not the case that $d > 0$ or
there is a $d'' \in D$, s.t.
$Val_{\mathfrak{M}, s\frac{d}{x}\frac{d'}{y}\frac{d''}{z}}(P(z)) = 1$ and $Val_{\mathfrak{M}, s\frac{d}{x}\frac{d'}{y}\frac{d''}{z}}(R(g(z,x),y)) = 1$
iff

for all $d \in D$, for all $d' \in D$:
$d \leq 0$ or
there is a $d'' \in D$, s.t.
$s\frac{d}{x}\frac{d'}{y}\frac{d''}{z}(z) \in \mathfrak{I}(P)$ and
$(Val_{\mathfrak{M}, s\frac{d}{x}\frac{d'}{y}\frac{d''}{z}}(g(z,x)), s\frac{d}{x}\frac{d'}{y}\frac{d''}{z}(y)) \in \mathfrak{I}(R)$
iff

for all $d \in D$, for all $d' \in D$:
$d \leq 0$ or
there is a $d'' \in D$, s.t. $d'' \in \mathbb{N}$ and $d'' \cdot d > d'$

("For all $x \in \mathbb{R}$ with $x > 0$ and all $y \in \mathbb{R}$ there is an $n \in \mathbb{N}$, such that $nx > y$")

For which variable assignments s over $\mathfrak{M}$ is it the case that

$P(z) \wedge R(z, c_1) \wedge \forall x(P(x) \wedge \exists y(P(y) \wedge g(x,y) \equiv z) \rightarrow x \equiv c_1 \vee x \equiv z)$

is true at $\mathfrak{M}$ and s:

For those variable assignments s for which $s(z)$ is a *prime number*!

2. Let $\mathcal{S} = \{P, f\}$, where P is a unary predicate and f is a binary function sign.

   For each of the following formulas in $\mathcal{F}_{\mathcal{S}}$ find an $\mathcal{S}$-model and a corresponding variable assignment relative to which the formula is true and find an $\mathcal{S}$-model and a corresponding variable assignment relative to which the formula is false:

(a) $\forall v_1 f(v_2, v_1) \equiv v_2$:

For $D = \mathbb{N}, \mathfrak{I}(f) = \cdot, s(v_2) = 0$: *true*
For $D = \mathbb{N}, \mathfrak{I}(f) = +, s(v_2) = 0$: *false*

(b) $\exists v_2 \forall v_1 f(v_2, v_1) \equiv v_2$: analogous to 2a

(c) $\exists v_2 (P(v_2) \wedge \forall v_1 P(f(v_2, v_1)))$:

For $D = \mathbb{N}$, $\mathfrak{I}(f) = \cdot$, $\mathfrak{I}(P) =$ set of even natural numbers: *true*
For $D = \mathbb{N}$, $\mathfrak{I}(f) = \cdot$, $\mathfrak{I}(P) =$ set of odd natural numbers: *false*

3. Let D be finite and non-empty, let $\mathcal{S}$ be finite. Show that there are only finitely many $\mathcal{S}$-models with domain D.

Proof. Let k be the cardinality of D. For every constant c in $\mathcal{S}$ there are k possible ways of choosing $\mathfrak{I}(c)$. For every n-ary predicate P in $\mathcal{S}$ there are $2^{(k^n)}$ possible ways of choosing $\mathfrak{I}(P)$. Finally, for every n-ary function sign f in $\mathcal{S}$ there are $k^{(k^n)}$ possible ways of choosing $\mathfrak{I}(f)$. Since there are only finitely many symbols in $\mathcal{S}$, the numbers of possible interpretations mappings $\mathfrak{I}$ on $\mathcal{S}$ (and thus the number of $\mathcal{S}$-models $\mathfrak{M}$) is a finite product of finite numbers of the form k or $2^{(k^n)}$ or $k^{(k^n)}$; but such a product is of course finite. ■

4. A formula in which $\neg$, $\rightarrow$, $\leftrightarrow$ do not occur is called *positive*.

Prove: For every positive formula there is a model and a variable assignment which taken together satisfy the formula (independent of what $\mathcal{S}$ is like).

Hint: You might consider "trivial" models the domains of which only have one member.

Proof. Let $D = \{1\}$.
Let $\mathfrak{I}(P) = D^n = \{(\underbrace{1, 1, \ldots, 1}_{n})\}$ (for n-ary predicates P).

Let $\mathfrak{I}(f) : D^n \rightarrow D$ s.t. $(\underbrace{1, 1, \ldots, 1}_{n}) \mapsto 1$ (for n-ary function signs f).

Let $\mathfrak{M} = (D, \mathfrak{I})$.

By the definition of *positive formula*, all and only positive formulas can be derived in the following positive-formula calculus of rules:

(P1) $\overline{\equiv(t_1,t_2)}$ (for $\mathcal{S}$-terms t_1, t_2)

(P2) $\overline{P(t_1,\ldots,t_n)}$ (for $\mathcal{S}$-terms $t_1, \ldots, t_n$, for n-ary $P \in \mathcal{S}$)

(P3) $\underbrace{\frac{\varphi, \psi}{(\varphi \vee \psi)}}_{\text{disjunction}}$ $\underbrace{\frac{\varphi, \psi}{(\varphi \wedge \psi)}}_{\text{conjunction}}$

(P4) $\underbrace{\frac{\varphi}{\forall x\, \varphi}}_{\text{universally quantified}}$ $\underbrace{\frac{\varphi}{\exists x\, \varphi}}_{\text{existentially quantified}}$ (for arbitrary variables x)

By induction over positive formulas we can show that for every positive formula φ it holds that:

For every variable assignment s over $\mathfrak{M}$, $Val_{\mathfrak{M},s}(\varphi) = 1$.

(Note that there is actually just one variable assignment s over $\mathfrak{M}$!)

P1&P2: $Val_{\mathfrak{M},s}(P(t_1, \ldots, t_n)) = 1$ because
$\underbrace{(Val_{\mathfrak{M},s}(t_1), \ldots, Val_{\mathfrak{M},s}(t_n))}_{(1,1,\ldots,1)} \in \underbrace{\mathfrak{I}(P)}_{D^n}$ ✓
(analogously for $\equiv$)

P3: Assume that $Val_{\mathfrak{M},s}(\varphi) = Val_{\mathfrak{M},s}(\varphi) = 1$:
but then it follows that $Val_{\mathfrak{M},s}(\varphi \wedge \psi) = 1$ and $Val_{\mathfrak{M},s}(\varphi \vee \psi) = 1$ (for arbitrary s).

P4: Assume that $Val_{\mathfrak{M},s}(\varphi) = 1$:
$Val_{\mathfrak{M},s}(\forall x \varphi) = 1$ iff
for all $d \in D$ $Val_{\mathfrak{M},s\frac{d}{x}}(\varphi) = 1$ iff (since $s\frac{d}{x} = s$)
for all $d \in D$ $Val_{\mathfrak{M},s}(\varphi) = 1$ iff
$Val_{\mathfrak{M},s}(\varphi) = 1$, which is the case by inductive assumption.
So we are done (analogously for $\exists$).

■

5. Prove the *coincidence lemma* by induction over terms and formulas:

Let $\mathcal{S}_1, \mathcal{S}_2$ be two symbol sets. Let $\mathfrak{M}_1 = (D, \mathfrak{I}_1)$ be an $\mathcal{S}_1$-model, let $\mathfrak{M}_2 = (D, \mathfrak{I}_2)$ be an $\mathcal{S}_2$-model.
Let s_1 be a variable assignment over $\mathfrak{M}_1$, s_2 a variable assignment over $\mathfrak{M}_2$.
Finally, let $\mathcal{S} = \mathcal{S}_1 \cap \mathcal{S}_2$:

(a) For all terms $t \in \mathcal{T}_\mathcal{S}$:

If $\mathfrak{I}_1(c) = \mathfrak{I}_2(c)$ for all c in t
$\mathfrak{I}_1(f) = \mathfrak{I}_2(f)$ for all f in t
$s_1(x) = s_2(x)$ for all x in t

then: $Val_{\mathfrak{M}_1,s_1}(t) = Val_{\mathfrak{M}_2,s_2}(t)$

(b) For all formulas $\varphi \in \mathcal{F}_\mathcal{S}$:

If $\mathfrak{I}_1(c) = \mathfrak{I}_2(c)$ for all c in φ
$\mathfrak{I}_1(f) = \mathfrak{I}_2(f)$ for all f in φ
$\mathfrak{I}_1(P) = \mathfrak{I}_2(P)$ for all P in φ
$s_1(x) = s_2(x)$ for all $x \in free(\varphi)$

then: $Val_{\mathfrak{M}_1,s_1}(\varphi) = Val_{\mathfrak{M}_2,s_2}(\varphi)$

Proof. By induction over terms and formulas.
First we show that all $\mathcal{S}$-terms t have the following property P:

For all $\mathfrak{M}_1 = (D, \mathfrak{I}_1), \mathfrak{M}_2 = (D, \mathfrak{I}_2), s_1, s_2$:
If $\mathfrak{I}_1(c) = \mathfrak{I}_2(c)$ for all c in t
$\mathfrak{I}_1(f) = \mathfrak{I}_2(f)$ for all f in t
$s_1(x) = s_2(x)$ for all x in t
then: $Val_{\mathfrak{M}_1,s_1}(t) = Val_{\mathfrak{M}_2,s_2}(t)$

This can be proven as follows:

$t = c$: If $\mathfrak{I}_1(c) = \mathfrak{I}_2(c)$ for all c in t, $\mathfrak{I}_1(f) = \mathfrak{I}_2(f)$ for all f in t, $s_1(x) = s_2(x)$ for all x in t, then
$Val_{\mathfrak{M}_1,s_1}(c) = \mathfrak{I}_1(c) = \mathfrak{I}_2(c) = Val_{\mathfrak{M}_2,s_2}(c)$.

$t = x$: If $\mathfrak{I}_1(c) = \mathfrak{I}_2(c)$ for all c in t, $\mathfrak{I}_1(f) = \mathfrak{I}_2(f)$ for all f in t, $s_1(x) = s_2(x)$ for all x in t, then
$Val_{\mathfrak{M}_1,s_1}(x) = s_1(x) = s_2(x) = Val_{\mathfrak{M}_2,s_2}(x)$.

$t = f(t_1, \ldots, t_n)$: Assume that $t_1, \ldots, t_n$ have property P:

If $\mathfrak{I}_1(c) = \mathfrak{I}_2(c)$ for all c in t, $\mathfrak{I}_1(f) = \mathfrak{I}_2(f)$ for all f in t,
$s_1(x) = s_2(x)$ for all x in t, then
$Val_{\mathfrak{M}_1,s_1}(f(t_1, \ldots, t_n)) =$
$\mathfrak{I}_1(f)(Val_{\mathfrak{M}_1,s_1}(t_1), \ldots, Val_{\mathfrak{M}_1,s_1}(t_n)) =$
(by the "if"-part and by the inductive assumption)
$\mathfrak{I}_2(f)(Val_{\mathfrak{M}_2,s_2}(t_1), \ldots, Val_{\mathfrak{M}_2,s_2}(t_n)) =$
$Val_{\mathfrak{M}_2,s_2}(f(t_1, \ldots, t_n))$.

Next we show that all $\mathcal{S}$-formulas φ have the following property P:

For all $\mathfrak{M}_1 = (D, \mathfrak{I}_1), \mathfrak{M}_2 = (D, \mathfrak{I}_1), s_1, s_2$:
If $\mathfrak{I}_1(c) = \mathfrak{I}_2(c)$ for all c in φ
$\mathfrak{I}_1(f) = \mathfrak{I}_2(f)$ for all f in φ
$\mathfrak{I}_1(P) = \mathfrak{I}_2(P)$ for all P in φ
$s_1(x) = s_2(x)$ for all $x \in free(\varphi)$
then: $Val_{\mathfrak{M}_1,s_1}(\varphi) = Val_{\mathfrak{M}_2,s_2}(\varphi)$

This can be proven as follows (we show it for representative cases):

$\varphi = P(t_1, \ldots, t_n)$: If $\mathfrak{I}_1(c) = \mathfrak{I}_2(c)$ for all c in φ, $\mathfrak{I}_1(f) = \mathfrak{I}_2(f)$ for all f in φ, $\mathfrak{I}_1(P) = \mathfrak{I}_2(P)$ for all P in φ, $s_1(x) = s_2(x)$ for all $x \in free(\varphi)$ then
$Val_{\mathfrak{M}_1,s_1}(P(t_1, \ldots, t_n)) = 1$ iff
$(Val_{\mathfrak{M}_1,s_1}(t_1), \ldots, Val_{\mathfrak{M}_1,s_1}(t_n)) \in \mathfrak{I}_1(P)$ iff
(by the "if"-part and by what we have shown before for terms t; note that $free(\varphi) = var(\varphi)$ for atomic φ)
$(Val_{\mathfrak{M}_2,s_2}(t_1), \ldots, Val_{\mathfrak{M}_2,s_2}(t_n)) \in \mathfrak{I}_2(P)$ iff
$Val_{\mathfrak{M}_2,s_2}(P(t_1, \ldots, t_n)) = 1$.

$\varphi = \neg\psi$: Assume that ψ has the property P:

If $\mathfrak{I}_1(c) = \mathfrak{I}_2(c)$ for all c in φ, $\mathfrak{I}_1(f) = \mathfrak{I}_2(f)$ for all f in φ, $\mathfrak{I}_1(P) = \mathfrak{I}_2(P)$ for all P in φ, $s_1(x) = s_2(x)$ for all $x \in free(\varphi)$ then
$Val_{\mathfrak{M}_1,s_1}(\neg\psi) = 1$ iff
$Val_{\mathfrak{M}_1,s_1}(\psi) = 0$ iff
(by the inductive assumption)
$Val_{\mathfrak{M}_2,s_2}(\psi) = 0$ iff
$Val_{\mathfrak{M}_2,s_2}(\neg\psi) = 1$.

$\varphi = \exists x\psi$: Assume that ψ has the property P:

If $\mathfrak{I}_1(c) = \mathfrak{I}_2(c)$ for all c in φ, $\mathfrak{I}_1(f) = \mathfrak{I}_2(f)$ for all f in φ, $\mathfrak{I}_1(P) = \mathfrak{I}_2(P)$ for all P in φ, $s_1(x) = s_2(x)$ for all $x \in free(\varphi)$ then
$Val_{\mathfrak{M}_1,s_1}(\exists x\psi) = 1$ iff
there is a $d \in D$, such that $Val_{\mathfrak{M}_1,s_1\frac{d}{x}}(\psi) = 1$ iff
(by the inductive assumption)
there is a $d \in D$, such that $Val_{\mathfrak{M}_2,s_2\frac{d}{x}}(\psi) = 1$ iff
$Val_{\mathfrak{M}_2,s_2}(\exists x\psi) = 1$.

■

7.3 Solutions to Problem Set 3

1. The convergence of a real-valued sequence (x_n) to a limit x is usually defined as follows:

   (Conv) For all $\epsilon > 0$ there is a natural number n, such that for all natural numbers $m > n$ it holds that: $|x_m - x| < \epsilon$

   Represent (Conv) in a first-order language by choosing an appropriate symbol set $\mathcal{S}$ and define the corresponding $\mathcal{S}$-model.

   Hint: (i) Real sequences are functions from $\mathbb{N}$ to $\mathbb{R}$, i.e., you may consider x_m as being of the form $f(m)$; f can be regarded as being defined on $\mathbb{R}$ even though only its values for members of $\mathbb{N}$ are "relevant". (ii) $|x_m - x|$ may either be considered as the result of applying a binary "distance" function to the arguments x_m and x or as the result of applying two functions – subtraction and absolute value – to these arguments.

   Answer: We choose $\mathcal{S} = \{0, d, >, N, f\}$.

   Let $D = \mathbb{R}$, $\mathfrak{I}(0) = 0$, $\mathfrak{I}(d) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ with $\mathfrak{I}(d)(a, b) = |a - b|$, $\mathfrak{I}(>)$ is the $>$-relation on $\mathbb{R}$, $\mathfrak{I}(N) = \mathbb{N}$, $\mathfrak{I}(f)$ is a function from $\mathbb{R}$ to $\mathbb{R}$; note that with regard to arguments $n \in \mathbb{N}$, the function $\mathfrak{I}(f)$ is a real-valued sequence on $\mathbb{N}$ that maps n to $\mathfrak{I}(f)(n)$.
   So we can represent (Conv) from above as:

   $\forall v_0(v_0 > 0 \to \exists v_1(N(v_1) \land \forall v_2(N(v_2) \land v_2 > v_1 \to v_0 > d(f(v_2), x))))$

   (where x is a variable distinct from v_0, v_1, v_2).

2. (This problem counts for award of CREDIT POINTS.)
   Show that for arbitrary $\mathcal{S}$-formulas φ, ψ, ρ, and arbitrary sets Φ of $\mathcal{S}$-formulas the following is the case:

   (a) $(\varphi \lor \psi) \vDash \rho$ iff $\varphi \vDash \rho$ and $\psi \vDash \rho$:
   ("$\Rightarrow$") Assume that $(\varphi \lor \psi) \vDash \rho$.
   So for all $\mathfrak{M}, s$: if $\mathfrak{M}, s \models \varphi \lor \psi$ then $\mathfrak{M}, s \models \rho$.
   Now suppose $\mathfrak{M}, s \models \varphi$: then $\mathfrak{M}, s \models \varphi \lor \psi$ and thus by assumption $\mathfrak{M}, s \models \rho$. It follows that $\varphi \vDash \rho$.
   (Analogously for $\psi \vDash \rho$).

("$\Leftarrow$") Assume that $\varphi \vDash \rho$ and $\psi \vDash \rho$, i.e., for all $\mathfrak{M}, s$: if $\mathfrak{M}, s \models \varphi$ then $\mathfrak{M}, s \models \rho$, and for all $\mathfrak{M}, s$: if $\mathfrak{M}, s \models \psi$ then $\mathfrak{M}, s \models \rho$.
Now suppose $\mathfrak{M}, s \models \varphi \vee \psi$: then either (i) $\mathfrak{M}, s \models \varphi$ or (ii) $\mathfrak{M}, s \models \psi$; in either case, by assumption, $\mathfrak{M}, s \models \rho$.
It follows that $\varphi \vee \psi \vDash \rho$.

(b) $\Phi \cup \{\varphi\} \vDash \psi$ iff $\Phi \vDash (\varphi \to \psi)$:
("$\Rightarrow$") Assume that $\Phi \cup \{\varphi\} \vDash \psi$. So for all $\mathfrak{M}, s$: if $\mathfrak{M}, s \models \Phi \cup \{\varphi\}$ then $\mathfrak{M}, s \models \psi$.
Now suppose $\mathfrak{M}, s \models \Phi$; then there are two possible cases:
Case 1: $\mathfrak{M}, s \not\models \varphi$. But then $\mathfrak{M}, s \models \varphi \to \psi$.
Case 2: $\mathfrak{M}, s \models \varphi$. But then $\mathfrak{M}, s \models \Phi \cup \{\varphi\}$, which implies by assumption that $\mathfrak{M}, s \models \psi$ and thus $\mathfrak{M}, s \models \varphi \to \psi$.
In either case, $\mathfrak{M}, s \models \varphi \to \psi$.
So it follows that $\Phi \vDash \varphi \to \psi$.
("$\Leftarrow$") Assume that $\Phi \vDash \varphi \to \psi$. Hence, for all $\mathfrak{M}, s$: if $\mathfrak{M}, s \models \Phi$ then $\mathfrak{M}, s \models \varphi \to \psi$.
Now suppose $\mathfrak{M}, s \models \Phi \cup \{\varphi\}$; then $\mathfrak{M}, s \models \Phi$, so by assumption $\mathfrak{M}, s \models \varphi \to \psi$. Since both $\mathfrak{M}, s \models \varphi$ and $\mathfrak{M}, s \models \varphi \to \psi$, it follows that $\mathfrak{M}, s \models \psi$.
But that means $\Phi \cup \{\varphi\} \vDash \psi$.

(c) $\varphi \vDash \psi$ (i.e., $\{\varphi\} \vDash \psi$) iff $(\varphi \to \psi)$ is logically true:
By 2b, $\varnothing \cup \{\varphi\} \vDash \psi$ iff $\varnothing \vDash \varphi \to \psi$, i.e.,
$\varphi \vDash \psi$ iff $\varnothing \vDash \varphi \to \psi$.
But by the lemma in our section on semantic concepts, the latter is equivalent to saying that $\varphi \to \psi$ is logically true.

3. (a) Prove for arbitrary $\mathcal{S}$-formulas φ, ψ:
$\exists x \forall y \varphi \vDash \forall y \exists x \varphi$
Proof. Strictly, we have to deal with two cases: (i) the variables x and y being distinct, or (ii) $x = y$.
Case 1: Let $x \neq y$ and $\mathfrak{M}, s \models \exists x \forall y \varphi$:
By the definition of Val it follows that
there is a $d_1 \in D$, such that $\mathfrak{M}, s\frac{d_1}{x} \models \forall y \varphi$, which in turn implies that there is a $d_1 \in D$, such that for all $d_2 \in D$: $\mathfrak{M}, (s\frac{d_1}{x})\frac{d_2}{y} \models \varphi$.
Not let $d_4 \in D$ be chosen arbitrarily. Furthermore, let d_3 be such that for all $d_2 \in D$: $\mathfrak{M}, (s\frac{d_3}{x})\frac{d_2}{y} \models \varphi$ (such a d_3 must exist by what we said before).

But then it must also be the case that $\mathfrak{M}, (s\frac{d_3}{x})\frac{d_4}{y} \models \varphi$.
So we found that for all $d_4 \in D$ there is a $d_3 \in D$ with:
$\mathfrak{M}, (s\frac{d_3}{x})\frac{d_4}{y} \models \varphi$.
Since $x \neq y$, we can also write this as follows:
for all $d_4 \in D$ there is a $d_3 \in D$ with $\mathfrak{M}, (s\frac{d_4}{y})\frac{d_3}{x} \models \varphi$.
By the definition of Val again, we have:
for all $d_4 \in D$, $\mathfrak{M}, s\frac{d_4}{y} \models \exists x\varphi$, and thus $\mathfrak{M}, s \models \forall y \exists x \varphi$.

Case 2: Let $x = y$ and $\mathfrak{M}, s \models \exists x \forall y \varphi$:
As above it follows that there is a $d_1 \in D$, such that for all $d_2 \in D$:
$\mathfrak{M}, (s\frac{d_1}{x})\frac{d_2}{y} \models \varphi$.
Because of $x = y$, $(s\frac{d_1}{x})\frac{d_2}{y} = s\frac{d_2}{x}$.
So we actually have that for all $d_2 \in D$: $\mathfrak{M}, s\frac{d_2}{y} \models \varphi$.
Therefore, since $D \neq \varnothing$, there is a $d_2 \in D$ such that $\mathfrak{M}, s\frac{d_2}{y} \models \varphi$,
which implies trivially that
for all $d_1 \in D$ there is a $d_2 \in D$ such that $\mathfrak{M}, (s\frac{d_1}{x})\frac{d_2}{y} \models \varphi$, i.e.,
$\mathfrak{M}, s \models \forall y \exists x \varphi$.

Summing up both cases, we find that $\exists x \forall y \varphi \vDash \forall y \exists x \varphi$. ■

(b) Show that the following is *not* the case for all $\mathcal{S}$-formulas φ, ψ:
$\forall y \exists x \varphi \vDash \exists x \forall y \varphi$
Proof. Consider $\varphi = P(x, y)$, $D = \mathbb{N}$, $\mathfrak{I}(P) = >$-relation on $\mathbb{N}$, $\mathfrak{M} = (D, \mathfrak{I})$:
then $\mathfrak{M} \models \forall y \exists x P(x, y)$, but $\mathfrak{M} \not\models \exists x \forall y P(x, y)$. ■

4. (a) Prove for all $\mathcal{S}$-formulas φ, ψ:
$\exists x(\varphi \vee \psi)$ is logically equivalent to $\exists x\varphi \vee \exists x\psi$.
(Proof: Immediate from the definition of Val.)

(b) Show that the following is *not* the case for all $\mathcal{S}$-formulas φ, ψ:
$\exists x(\varphi \wedge \psi)$ is logically equivalent to $\exists x\varphi \wedge \exists x\psi$.
Proof. Consider $\varphi = P(x)$, $\psi = Q(x)$, $D = \mathbb{N}$, let $\mathfrak{I}(P)$ be the set of even natural numbers, $\mathfrak{I}(Q)$ be the set of odd natural numbers, $\mathfrak{M} = (D, \mathfrak{I})$:
then $\mathfrak{M} \models \exists x P(x) \wedge \exists x Q(x)$, but $\mathfrak{M} \not\models \exists x(P(x) \wedge Q(x))$. ■

5. Let Φ be an $\mathcal{S}$-formula set, let φ und ψ be $\mathcal{S}$-formulas. Show:
If $\Phi \cup \{\varphi\} \vDash \psi$ and $\Phi \vDash \varphi$, then $\Phi \vDash \psi$.

Proof. Assume that $\Phi \cup \{\varphi\} \vDash \psi$ and $\Phi \vDash \varphi$:
let $\mathfrak{M}, s \models \Phi$; then by the second assumption $\mathfrak{M}, s \models \varphi$, hence $\mathfrak{M}, s \models \Phi \cup \{\varphi\}$, which implies by the first assumption that $\mathfrak{M}, s \models \psi$.
It follows that $\Phi \vDash \psi$. ■

6. A set Φ of $\mathcal{S}$-sentences is called "independent if and only if there is no $\varphi \in \Phi$ such that: $\Phi \setminus \{\varphi\} \vDash \varphi$ (i.e., φ is not "redundant", because it is impossible to conclude φ from $\Phi \setminus \{\varphi\}$).

   Prove: (a) the set of the three group axioms and (b) the set of the three axioms for equivalence structures are both independent (see chapter one for these axioms).

Proof. Concerning (a):

G1 $\forall x \forall y \forall z \, (x \circ y) \circ z = x \circ (y \circ z)$

G2 $\forall x \, x \circ e = x$

G3 $\forall x \exists y \, x \circ y = e$

(i) $\{\mathrm{G1}, \mathrm{G2}\} \not\models \mathrm{G3}$: consider $D = \mathbb{Z}$, $\mathfrak{I}(\circ) =$ multiplication in $\mathbb{Z}$, $\mathfrak{I}(e) = 1$; then $\mathfrak{M} = (D, \mathfrak{I}) \models \{\mathrm{G1}, \mathrm{G2}\}$, but $\mathfrak{M} = (D, \mathfrak{I}) \not\models \mathrm{G3}$.

(ii) $\{\mathrm{G1}, \mathrm{G3}\} \not\models \mathrm{G2}$: consider $D = \mathbb{Q} \setminus \{0\}$, $\mathfrak{I}(\circ) =$ multiplication in $\mathbb{Q}$, $\mathfrak{I}(e) = 5$; then $\mathfrak{M} = (D, \mathfrak{I}) \models \{\mathrm{G1}, \mathrm{G3}\}$, but $\mathfrak{M} = (D, \mathfrak{I}) \not\models \mathrm{G2}$.

(iii) $\{\mathrm{G2}, \mathrm{G3}\} \not\models \mathrm{G1}$: consider $D = \{d_0, d_1, d_2\}$ for pairwise distinct d_0, d_1, d_2, $\mathfrak{I}(e) = d_0$, such that $\mathfrak{I}(\circ)$ is given by the following multiplication table:

$\mathfrak{I}(\circ)$	d_0	d_1	d_2
d_0	d_0	d_1	d_2
d_1	d_1	d_0	d_1
d_2	d_2	d_2	d_0

then $\mathfrak{M} = (D, \mathfrak{I}) \models \{\mathrm{G2}, \mathrm{G3}\}$, but $\mathfrak{M} = (D, \mathfrak{I}) \not\models \mathrm{G1}$ (to see the latter, consider a multiplication of d_2, d_1, and d_2).

Concerning (b):

A1 $\forall x\, x \approx x$

A2 $\forall x \forall y\, (x \approx y \to y \approx x)$

A3 $\forall x \forall y \forall z\, (x \approx y \land y \approx z \to x \approx z)$

(i) $\{A1, A2\} \not\models$ A3: consider $D = \{d_0, d_1, d_2\}$ for pairwise distinct d_0, d_1, d_2, $\mathfrak{I}(\approx) = \{(d_0, d_0), (d_1, d_1), (d_2, d_2), (d_0, d_1), (d_1, d_0), (d_1, d_2), (d_2, d_1)\}$;
then $\mathfrak{M} = (D, \mathfrak{I}) \models \{A1, A2\}$, but $\mathfrak{M} = (D, \mathfrak{I}) \not\models$ A3.

(ii) $\{A1, A3\} \not\models$ A2: consider $D = \{d_0, d_1\}$ for distinct d_0, d_1, with $\mathfrak{I}(\approx) = \{(d_0, d_0), (d_1, d_1), (d_0, d_1)\}$;
then $\mathfrak{M} = (D, \mathfrak{I}) \models \{A1, A3\}$, but $\mathfrak{M} = (D, \mathfrak{I}) \not\models$ A2.

(iii) $\{A2, A3\} \not\models$ A1: consider $D = \{d_0\}$, $\mathfrak{I}(\approx) = \varnothing$;
then $\mathfrak{M} = (D, \mathfrak{I}) \models \{A2, A3\}$, but $\mathfrak{M} = (D, \mathfrak{I}) \not\models$ A1. ■

7.4 Solutions to Problem Set 4

1. (a) $[\exists v_0 \exists v_1 (P(v_0, v_2) \wedge P(v_1, v_3))] \frac{v_2 \quad v_2 \quad v_2}{v_0 \quad v_1 \quad v_3}$

$$= \exists v_0 \, [\exists v_1 (P(v_0, v_2) \wedge P(v_1, v_3))] \frac{v_2 \quad v_0}{v_3 \quad v_0}$$

$$= \exists v_0 \exists v_1 \, [(P(v_0, v_2) \wedge P(v_1, v_3))] \frac{v_2 \quad v_1}{v_3 \quad v_1}$$

$$= \exists v_0 \exists v_1 (P(v_0, v_2) \wedge P(v_1, v_2))$$

(b) $[\exists v_0 \exists v_1 (P(v_0, v_2) \wedge P(v_1, v_3))] \frac{v_3 \quad f(v_2, v_3)}{v_2 \quad v_3}$

$$= \exists v_0 \, [\exists v_1 (P(v_0, v_2) \wedge P(v_1, v_3))] \frac{v_3 \quad f(v_2, v_3) \quad v_0}{v_2 \quad v_3 \quad v_0}$$

$$= \exists v_0 \exists v_1 \, [(P(v_0, v_2) \wedge P(v_1, v_3))] \frac{v_3 \quad f(v_2, v_3) \quad v_1}{v_2 \quad v_3 \quad v_1}$$

$$= \exists v_0 \exists v_1 (P(v_0, v_3) \wedge P(v_1, f(v_2, v_3)))$$

(c) $[\exists v_0 \exists v_1 (P(v_0, v_2) \wedge P(v_1, v_3))] \frac{v_2 \quad v_0 \quad f(v_2, v_3)}{v_0 \quad v_2 \quad v_3}$

$$= \exists v_4 \, [\exists v_1 (P(v_0, v_2) \wedge P(v_1, v_3))] \frac{v_0 \quad f(v_2, v_3) \quad v_4}{v_2 \quad v_3 \quad v_0}$$

$$= \exists v_4 \exists v_1 \, [(P(v_0, v_2) \wedge P(v_1, v_3))] \frac{v_0 \quad f(v_2, v_3) \quad v_4 \quad v_1}{v_2 \quad v_3 \quad v_0 \quad v_1}$$

$$= \exists v_4 \exists v_1 (P(v_4, v_0) \wedge P(v_1, f(v_2, v_3)))$$

(d) $[\forall v_0 \exists v_1 (P(v_0, v_1) \wedge P(v_0, v_2)) \vee \exists v_2 f(v_2, v_2) \equiv v_0] \frac{v_0 \quad f(v_0, v_1)}{v_0 \quad v_2}$:

At first we consider the left part of the given $\vee$-formula:

$$[\forall v_0 \exists v_1 (P(v_0, v_1) \wedge P(v_0, v_2))] \frac{v_0 \quad f(v_0, v_1)}{v_0 \quad v_2}$$

$$= \forall v_3 \, [\exists v_1 (P(v_0, v_1) \wedge P(v_0, v_2))] \frac{f(v_0, v_1) \quad v_3}{v_2 \quad v_0}$$

$$= \forall v_3 \exists v_4 \, [(P(v_0, v_1) \wedge P(v_0, v_2))] \frac{f(v_0, v_1) \quad v_3 \quad v_4}{v_2 \quad v_0 \quad v_1}$$

$$= \forall v_3 \exists v_4 (P(v_3, v_4) \wedge P(v_3, f(v_0, v_1)))$$

Secondly, we consider the right part of the given $\vee$-formula:

$$[\exists v_2 f(v_2, v_2) \equiv v_0]\frac{v_0 \quad f(v_0, v_1)}{v_0 \quad v_2}$$

$$= \exists v_2 \, [f(v_2, v_2) \equiv v_0]\frac{v_2}{v_2}$$

$$= \exists v_2 f(v_2, v_2) \equiv v_0$$

By connecting the two partial solutions by means of $\vee$ we get the final result, i.e.:

$$\forall v_3 \exists v_4 (P(v_3, v_4) \wedge P(v_3, f(v_0, v_1))) \vee \exists v_2 f(v_2, v_2) \equiv v_0$$

2. Let $t_0, \ldots, t_n$ be $\mathcal{S}$-terms, $x_0, \ldots, x_n$ pairwise distinct variables, φ an $\mathcal{S}$-formula and y a variable.

   Prove:

   (a) If π is a permutation of the numbers $0, \ldots, n$, then:
   $$\varphi\frac{t_0, \ldots, t_n}{x_0, \ldots, x_n} = \varphi\frac{t_{\pi(0)}, \ldots, t_{\pi(n)}}{x_{\pi(0)}, \ldots, x_{\pi(n)}}$$
   Proof. Strictly, this is shown first by induction over terms t and then by induction over formulas φ (the latter yields the proof of the claim above). But this time – instead of writing down all the details of the two inductive proofs – we will just informally "check" whether the order of terms/variables can have any effect on the outcome of a substitution. [If you have done so using just two or three lines of comment, that's fine.]
   In the basic cases of substitution within terms, there is no such effect, i.e.:

   - $[x]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} := \begin{cases} t_i & \text{for } x = x_i \ (0 \leq i \leq n) \\ x & \text{else} \end{cases} = [x]\frac{t_{\pi(0)},\ldots,t_{\pi(n)}}{x_{\pi(0)},\ldots,x_{\pi(n)}}$
   - $[c]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} := c = [c]\frac{t_{\pi(0)},\ldots,t_{\pi(n)}}{x_{\pi(0)},\ldots,x_{\pi(n)}}$

   Therefore, the order in which terms are substituted for variables cannot affect the outcome of a substitution within a function term (using the inductive assumption for its subterms) and thus it does not have any effect within any term whatsoever. Because the substitution of terms for variables within atomic formulas is by definition given by the substitution of terms for variables within terms, the order also does not play a role for substitutions within atomic formulas. Moreover, since the substitution of terms for variables within negation, disjunction, conjunction, implication, and equivalence formulas is by definition reduced to the substitution of terms for variables within their subformulas, the order of variables does not play a role for substitution within them as long as it does not play a role for the subformulas (this is where the inductive assumption on subformulas would be applied). The remaining case is the case for quantified formulas: there we have

- $[\exists x\varphi]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} := \exists u\ [\varphi]\frac{t_{i_1},\ldots,t_{i_k},u}{x_{i_1},\ldots,x_{i_k},x} = [\exists x\varphi]\frac{t_{\pi(0)},\ldots,t_{\pi(n)}}{x_{\pi(0)},\ldots,x_{\pi(n)}}$

given that the order in which terms are substituted for variables within the subformula φ is irrelevant, which is guaranteed by the inductive assumption again (accordingly for universally quantified formulas). So we are done. ■

(b) If $y \in var(t\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n})$, then

i. $y \in var(t_0) \cup \ldots \cup var(t_n)$ or

ii. $y \in var(t)$ and $y \neq x_0,\ldots,x_n$.

Proof. By induction over terms t:
Induction basis:

- $t = x$:
  Case 1: $x \neq x_0,\ldots,x_n$.
  Then $[t]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} = x$, so if $y \in var(t\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n})$ then y must be identical to x and thus $y \neq x_0,\ldots,x_n$.

  Case 2: $x = x_i$ (for some $i \in \{0,\ldots,n\}$).
  Then $[t]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n} = t_i$; therefore, if $y \in var(t\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n})$ then $y \in var(t_i) \subseteq var(t_0) \cup \ldots \cup var(t_n)$.
- $t = c$:
  there is no $y \in var(t\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n})$, so we are done (trivially).

Now consider $t = f(t'_1,\ldots,t'_m)$ and assume that $t'_1,\ldots,t'_m$ have the property stated in 2a.

Suppose $y \in var(t\frac{t_0, \ldots, t_n}{x_0, \ldots, x_n})$:

It follows that $y \in var\left(f\left([t'_1]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}, \ldots, [t'_m]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}\right)\right)$, and hence that $y \in var([t'_1]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n}) \cup \ldots \cup var([t'_m]\frac{t_0,\ldots,t_n}{x_0,\ldots,x_n})$.

By the inductive assumption, it follows that

$(y \in var(t_0) \cup \ldots \cup var(t_n)$ or $y \in var(t'_1)$ and $y \neq x_0, \ldots, x_n)$ or

$\vdots$

$(y \in var(t_0) \cup \ldots \cup var(t_n)$ or $y \in var(t'_m)$ and $y \neq x_0, \ldots, x_n)$,

which finally implies:

$y \in var(t_0) \cup \ldots \cup var(t_n)$ or
$y \in var(t)$ and $y \neq x_0, \ldots, x_n$

(since $t = f(t'_1, \ldots, t'_m)$). ■

7.5 Solutions to Problem Set 5

1. (This problem counts for award of CREDIT POINTS.)

   Are the following rules correct?

   (a) $$\begin{array}{lll} \Gamma & \varphi_1 & \psi_1 \\ \Gamma & \varphi_2 & \psi_2 \\ \hline \Gamma & \varphi_1 \vee \varphi_2 & \psi_1 \vee \psi_2 \end{array}$$

   This rule is *correct*:

   Assume that $\Gamma\,\varphi_1\,\psi_1$, $\Gamma\,\varphi_2\,\psi_2$ are correct, i.e., $\Gamma \cup \{\varphi_1\} \models \psi_1$ and $\Gamma \cup \{\varphi_2\} \models \psi_2$. Now we show that in this case also $\Gamma\,\varphi_1 \vee \varphi_2\;\psi_1 \vee \psi_2$ is correct, i.e., $\Gamma \cup \{\varphi_1 \vee \varphi_2\} \models \psi_1 \vee \psi_2$. For consider arbitrary $\mathfrak{M}, s$ with $\mathfrak{M}, s \models \Gamma \cup \{\varphi_1 \vee \varphi_2\}$: it follows that either (i) $\mathfrak{M}, s \models \Gamma \cup \{\varphi_1\}$ or (ii) $\mathfrak{M}, s \models \Gamma \cup \{\varphi_2\}$. Since $\Gamma \cup \{\varphi_1\} \models \psi_1$ and $\Gamma \cup \{\varphi_2\} \models \psi_2$ it must be the case that either (i) $\mathfrak{M}, s \models \Gamma \cup \{\psi_1\}$ or (ii) $\mathfrak{M}, s \models \Gamma \cup \{\psi_2\}$. In either case, $\mathfrak{M}, s \models \Gamma \cup \{\psi_1 \vee \psi_2\}$ and thus we are done.

   (b) $$\begin{array}{lll} \Gamma & \varphi_1 & \psi_1 \\ \Gamma & \varphi_2 & \psi_2 \\ \hline \Gamma & \varphi_1 \vee \varphi_2 & \psi_1 \wedge \psi_2 \end{array}$$

   This rule is *not* correct:

   Consider the following exemplary instance of the rule:
   $$\begin{array}{ll} P(c) & P(c) \\ \neg P(c) & \neg P(c) \\ \hline P(c) \vee \neg P(c) & P(c) \wedge \neg P(c) \end{array}$$
   (Γ is chosen to be empty).

   Obviously, both premises are correct while the conclusion is not (as far as the latter is concerned, any model whatsoever for the symbol set $\mathcal{S} = \{P, c\}$ is a counterexample).

2. Derive the following (auxiliary) rules from the rules of the sequent calculus:

   (a) $$\begin{array}{ll} \Gamma & \varphi \\ \hline \Gamma & \neg\neg\varphi \end{array}$$

1. $\Gamma\varphi$ (Premise)
2. $\Gamma\neg\varphi\neg\varphi$ (Ass.)
3. $\Gamma\varphi\,\neg\neg\varphi$ (CP 2) with 2.
4. $\Gamma\neg\neg\varphi$ (CS) with 1., 3.

(b) $\dfrac{\Gamma \quad \neg\neg\varphi}{\Gamma \quad \varphi}$

1. $\Gamma\neg\neg\varphi$ (Premise)
2. $\Gamma\neg\varphi\neg\varphi$ (Ass.)
3. $\Gamma\neg\varphi\,\neg\neg\varphi$ (Ant.) with 1.
4. $\Gamma\varphi$ (CD) with 2., 3.

(c) $\dfrac{\begin{array}{ll}\Gamma & \varphi \\ \Gamma & \psi\end{array}}{\Gamma \quad \varphi \wedge \psi}$

1. $\Gamma\varphi$ (Premise)
2. $\Gamma\psi$ (Premise)
3. $\Gamma\neg\varphi \vee \neg\psi \;\; \neg\varphi \vee \neg\psi$ (Ass.)
4. $\Gamma\neg\varphi \vee \neg\psi \;\; \varphi$ (Ant.) with 1.
5. $\Gamma\neg\varphi \vee \neg\psi \;\; \neg\neg\varphi$ (2a) with 4.
6. $\Gamma\neg\varphi \vee \neg\psi \;\; \neg\psi$ (DS) with 3., 5.
7. $\Gamma\neg\varphi \vee \neg\psi \;\; \psi$ (Ant.) with 2.
8. $\Gamma\neg\varphi \vee \neg\psi \;\; \neg(\neg\varphi \vee \neg\psi)$ (Triv.) with 7., 6.
9. $\Gamma\neg(\neg\varphi \vee \neg\psi) \;\; \neg(\neg\varphi \vee \neg\psi)$ (Ant.)
10. $\Gamma \underbrace{\neg(\neg\varphi \vee \neg\psi)}_{\varphi\wedge\psi}$ (PC) with 8., 9.

(d) $\dfrac{\Gamma \quad \varphi \quad \psi}{\Gamma \quad \varphi \to \psi}$

1. $\Gamma\varphi\psi$ (Premise)
2. $\Gamma\varphi\ \neg\varphi \lor \psi$ ($\lor$-Con.) with 1.
3. $\Gamma\neg\varphi\ \neg\varphi$ (Ass.)
4. $\Gamma\neg\varphi\ \neg\varphi \lor \psi$ ($\lor$-Con.) with 3.
5. $\Gamma\underbrace{\neg\varphi \lor \psi}_{\varphi\to\psi}$ (PC) with 2., 4.

(e) $\dfrac{\Gamma \quad \varphi \land \psi}{\Gamma \quad \varphi}$

1. $\Gamma\underbrace{\neg(\neg\varphi \lor \neg\psi)}_{\varphi\land\psi}$ (Premise)
2. $\Gamma\neg\varphi\ \neg\varphi$ (Ass.)
3. $\Gamma\neg\varphi\ \neg\varphi \lor \neg\psi$ ($\lor$-Con.) with 2.
4. $\Gamma\neg\varphi\ \neg(\neg\varphi \lor \neg\psi)$ (Ant.) with 1.
5. $\Gamma\neg\varphi\ \varphi$ (Triv.) with 3., 4.
6. $\Gamma\varphi\ \varphi$ (Ass.)
7. $\Gamma\varphi$ (PC) with 6., 5.

(f) $\dfrac{\Gamma \quad \varphi \land \psi}{\Gamma \quad \psi}$

Analogous to 2e!

3. Are the following rules correct?

(a) $\dfrac{\varphi \qquad \psi}{\exists x\varphi \quad \exists x\psi}$

This rule is *correct*:

Assume that $\varphi\,\psi$ is correct, i.e., $\{\varphi\} \models \psi$. We show that in this case also $\exists x\varphi\,\exists x\psi$ is correct, i.e., $\{\exists x\varphi\} \models \exists x\psi$. For consider arbitrary $\mathfrak{M}, s$ with $\mathfrak{M}, s \models \exists x\varphi$: it follows that there is a d in the domain of $\mathfrak{M}$, such that $\mathfrak{M}, s\frac{d}{x} \models \varphi$. Since $\{\varphi\} \models \psi$, we have $\mathfrak{M}, s\frac{d}{x} \models \psi$ and hence $\mathfrak{M}, s \models \exists x\psi$.

(b) $\dfrac{\Gamma \quad \varphi \qquad \psi}{\Gamma \quad \forall x\varphi \quad \exists x\psi}$

This rule is *correct*:

Assume that $\Gamma\, \varphi\, \psi$ is correct, i.e., $\Gamma \cup \{\varphi\} \models \psi$. We prove that this implies that $\Gamma\, \forall x\varphi\, \exists x\psi$ is correct, i.e., $\Gamma \cup \{\forall x\varphi\} \models \exists x\psi$. Consider arbitrary $\mathfrak{M}, s$ with $\mathfrak{M}, s \models \Gamma \cup \{\forall x\varphi\}$: it follows that for all d in the domain of $\mathfrak{M}$: $\mathfrak{M}, s\frac{d}{x} \models \varphi$. Therefore, it is of course also the case that $\mathfrak{M}, s \models \varphi$ (simply take $d := s(x)$). So $\mathfrak{M}, s \models \Gamma \cup \{\varphi\}$ and by $\Gamma \cup \{\varphi\} \models \psi$ it follows that $\mathfrak{M}, s \models \Gamma \cup \{\psi\}$. So there must also be a d in the domain of $\mathfrak{M}$, such that $\mathfrak{M}, s\frac{d}{x} \models \psi$ (namely $d := s(x)$). We conclude that $\mathfrak{M}, s \models \exists x\psi$ and we are done.

4. Derive the following (auxiliary) rules from the sequent calculus:

(a) $\dfrac{\Gamma \quad \forall x\varphi}{\Gamma \quad \varphi\frac{t}{x}}$

1. $\Gamma\, \underbrace{\neg\exists x\neg\varphi}_{\forall x\varphi}$ (Premise)
2. $\Gamma \neg\varphi\frac{t}{x}\;\; \neg\varphi\frac{t}{x}$ (Ass.)
3. $\Gamma \neg\varphi\frac{t}{x}\;\; \exists x\neg\varphi$ ($\exists$-Con.) with 2.
4. $\Gamma \neg\varphi\frac{t}{x}\;\; \neg\exists x\neg\varphi$ (Ant.) with 1.
5. $\Gamma \varphi\frac{t}{x}$ (CD) with 3., 4.

(b) $\dfrac{\Gamma \quad \forall x\varphi}{\Gamma \quad \varphi}$

Use 4a with $t := x$!

(c) $\dfrac{\Gamma \quad \varphi\frac{t}{x} \quad \psi}{\Gamma \quad \forall x\varphi \quad \psi}$

1. $\Gamma \varphi\frac{t}{x}\;\; \psi$ (Premise)
2. $\Gamma \neg\exists x\neg\varphi\;\; \neg\varphi\frac{t}{x}\;\; \neg\varphi\frac{t}{x}$ (Ass.)
3. $\Gamma \neg\exists x\neg\varphi\;\; \neg\varphi\frac{t}{x}\;\; \exists x\neg\varphi$ ($\exists$-Con.) with 2.

4. $\Gamma \neg\exists x \neg\varphi \;\; \neg\varphi\frac{t}{x} \;\; \neg\exists x \neg\varphi$ (Ass.)
5. $\Gamma \neg\exists x \neg\varphi \;\; \varphi\frac{t}{x}$ (CD) with 3., 4.
6. $\Gamma \neg\exists x \neg\varphi \;\; \varphi\frac{t}{x} \;\; \psi$ (Ant.) with 1.
7. $\Gamma \underbrace{\neg\exists x \neg\varphi}_{\forall x\varphi} \;\; \psi$ (CS) with 6., 5.

(d) $\dfrac{\Gamma \quad \varphi\frac{y}{x}}{\Gamma \quad \forall x\varphi}$ if y is not free in the sequent $\Gamma\ \forall x\varphi$.

1. $\Gamma \varphi\frac{y}{x}$ (Premise)
2. $\Gamma \neg\varphi\frac{y}{x} \;\; \neg\varphi\frac{y}{x}$ (Ass.)
3. $\Gamma \neg\varphi\frac{y}{x} \;\; \varphi\frac{y}{x}$ (Ant.) with 1.
4. $\Gamma \neg\varphi\frac{y}{x} \;\; \neg\exists x \neg\varphi$ (Triv.) with 3., 2.
5. $\Gamma \exists x \neg\varphi \;\; \neg\exists x \neg\varphi$ ($\exists$-Ant.) with 4. ✓
6. $\Gamma \neg\exists x \neg\varphi \;\; \neg\exists x \neg\varphi$ (Ass.)
7. $\Gamma \underbrace{\neg\exists x \neg\varphi}_{\forall x\varphi}$ (PC) with 5., 6.

(e) $\dfrac{\Gamma \quad \varphi \qquad \psi}{\Gamma \quad \forall x\varphi \quad \psi}$

1. $\Gamma \varphi \;\; \psi$ (Premise)
2. $\Gamma \neg\varphi \;\; \neg\varphi$ (Ass.)
3. $\Gamma \neg\varphi \;\; \exists x \neg\varphi$ ($\exists$-Con.) with 2.
4. $\Gamma \neg\exists x \neg\varphi \;\; \varphi$ (CP 3) with 3.
5. $\Gamma \neg\exists x \neg\varphi \;\; \varphi \;\; \psi$ (Ant.) with 1.
6. $\Gamma \underbrace{\neg\exists x \neg\varphi}_{\forall x\varphi} \;\; \psi$ (CS) with 4., 5.

(f) $\dfrac{\Gamma \quad \varphi}{\Gamma \quad \forall x\varphi}$ if x is not free in the sequent Γ.

Use 4d with $y := x$!

7.6 Solutions to Problem Set 6

1. Let $\mathcal{S}$ be an arbitrary symbol set.

   Let $\Phi = \{v_0 \equiv t \,| t \in \mathcal{T}_\mathcal{S}\} \cup \{\exists v_1 \exists v_2 \neg\, v_1 \equiv v_2\}$.

   Show:

   - Φ is consistent
   - there is no formula set $\Psi \subseteq \mathcal{F}_\mathcal{S}$ with $\Psi \supseteq \Phi$, such that Ψ is consistent and contains instances.

   Proof.

   (a) Φ is consistent as it is satisfied by e.g. the following model $\mathfrak{M} = (D, \mathfrak{I})$:
   Let $D = \{1, 2\}$; let $\mathfrak{I}(c) = 1$ for all constants c in $\mathcal{S}$, let $s(x) = 1$ for all variables x, and for all n-ary function signs f in $\mathcal{S}$ let $\mathfrak{I}(f) : D^n \to D$ with $\mathfrak{I}(f)(d) = 1$ for arbitrary $d \in D$. Since obviously $Val_{\mathfrak{M},s}(t) = 1$ for all terms t and since D contains more than just one element, it follows that $\mathfrak{M}, s \models \Phi$.

   (b) Consider an arbitrary formula set $\Psi \subseteq \mathcal{F}_\mathcal{S}$ with $\Psi \supseteq \Phi$, such that Ψ contains instances:
   Because Ψ contains instances, there must be $\mathcal{S}$-terms t, t', such that

   - $\exists v_1 \exists v_2 \neg v_1 \equiv v_2 \to \exists v_2 \neg t \equiv v_2$
   - $\exists v_2 \neg t \equiv v_2 \to \neg t \equiv t'$

   Since $\exists v_1 \exists v_2 \neg v_1 \equiv v_2 \in \Phi \subseteq \Psi$, it follows that $\neg t \equiv t'$ is derivable from Ψ (by two applications of modus ponens).
   But at the same time $v_0 \equiv t$ and $v_0 \equiv t'$ are members of $\Phi \subseteq \Psi$ by assumption. By applying the rules Symm. (symmetry of identity) and Trans. (transitivity of identity) of the sequent calculus (or alternatively the rule Sub. of substitution of identicals), we can thus derive $t \equiv t'$ from Ψ. But this means that Ψ is inconsistent and so we are done.

   ■

2. (This problem counts for award of CREDIT POINTS.)
Explain why the following logical implication holds:

$$\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y) \vDash$$

$$\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \leftrightarrow x = y)$$

Answer: $\forall x \forall y (x = y \rightarrow \forall z (z \in x \leftrightarrow z \in y))$ is logically true, i.e., satisfied in *every* model whatsoever! Therefore, it is also logically implied by every formula, which means that

$$\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y) \vDash$$

$$\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y) \wedge \forall x \forall y (x = y \rightarrow \forall z (z \in x \leftrightarrow z \in y))$$

is the case. But the latter formula is obviously logically equivalent to

$$\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \leftrightarrow x = y)$$

(Note that strictly we should have used $\equiv$ again instead of $=$, but never mind...)

3. Prove: A map with countably many countries can be coloured by using at most four colours if and only if each of its finite submaps can be coloured by using at most four colours.

(Hint: choose a symbol set $\mathcal{S}$ in which ever constant represents a country, in which there are four unary predicates that represent four colours, and in which there is a binary predicate that stands for the neighbourhood relation between countries; represent maps as sets of formulas for this symbol set; apply the compactness theorem.)

Proof. Let a map M with countably many countries be given. We can concentrate on the interesting case, i.e., where M has a countably *infinite* set of countries $0, 1, 2, \ldots$.

Choose $\mathcal{S} = \mathcal{S}_M$ as follows:

- $c_0, c_1, c_2, \ldots$: constants representing countries $0, 1, 2, \ldots$ in M
- R, B, G, W: unary predicates representing colours
- N: binary predicate representing the neighbourhood of countries in M

Let $\Phi_M \subseteq \mathcal{F}_\mathcal{S}$ be defined as follows:

- $\neg c_i \equiv c_j \in \Phi_M$ for all pairwise distinct indices $i, j \in \mathbb{N}_0$
- $N(c_i, c_j) \in \Phi_M$ iff
  the i-th country is a neighbour of the j-th country in M
- $\neg N(c_i, c_j) \in \Phi_M$ iff
  the i-th country is not a neighbour of the j-th country in M
- The following sentence is a member of Φ_M:
  $\forall x((R(x) \land \neg B(x) \land \neg G(x) \land \neg W(x)) \lor$
  $(\neg R(x) \land B(x) \land \neg G(x) \land \neg W(x)) \lor$
  $(\neg R(x) \land \neg B(x) \land G(x) \land \neg W(x)) \lor$
  $(\neg R(x) \land \neg B(x) \land \neg G(x) \land W(x)))$
- The following sentence is a member of Φ_M:
  $\forall x \forall y(N(x, y) \rightarrow$
  $(\neg(R(x) \land R(y)) \land$
  $\neg(B(x) \land B(y)) \land$
  $\neg(G(x) \land G(y)) \land$
  $\neg(W(x) \land W(y))))$
- Φ_M does not have any further members.

Then M is represented by Φ_M, every finite submap M_n of M with countries $0, \ldots, n$ is represented by the set of sentences of Φ_{M_n} in which none of the constants $c_{n+1}, c_{n+2}, \ldots$ occurs.

So we can prove the right-to-left direction of the statement above (the other direction is trivial): assume every finite submap M_n can be coloured by means of four colours (or less). Translated into sets of formulas, this means that every subset of Φ_M that is of the form Φ_{M_n} is satisfiable. Since every finite subset of Φ_M is a subset of a set of the form Φ_{M_n} for some n, it follows that every finite subset of Φ_M is satisfiable. Hence, by the compactness theorem, Φ_M is satisfiable and thus M can be coloured by means of four colours (or less).

■

4. Let P be a binary predicate in $\mathcal{S}$.

   Prove that the formula

   $\forall x \neg P(x,x) \wedge \forall x \forall y \forall z(P(x,y) \wedge P(y,z) \rightarrow P(x,z)) \wedge \forall x \exists y P(x,y)$

   can only be satisfied by *infinite* models.

   Proof. Assume $\mathfrak{M} = (D, \mathfrak{I})$ satisfies this formula:

   - $D \neq \varnothing$ (by definition of *model*), so there is a $d_0 \in D$ which think of held fixed.
   - Since $\mathfrak{M} \models \forall x \exists y P(x,y)$ there must be a $d_1 \in D$ such that $(d_0, d_1) \in \mathfrak{I}(P)$. d_1 cannot be identical to d_0 because $\mathfrak{M} \models \forall x \neg P(x,x)$.
   - Let us assume it is true that there are pairwise distinct $d_0, \ldots, d_n$ (for $0 \leq n$) which are members of D and for which it is the case that $(d_0, d_1), (d_1, d_2), \ldots, (d_{n-1}, d_n) \in \mathfrak{I}(P)$:
     Since $\mathfrak{M} \models \forall x \exists y P(x,y)$ there must be a $d_{n+1} \in D$ such that $(d_n, d_{n+1}) \in \mathfrak{I}(P)$.

     d_{n+1} cannot be identical to any of the d_k for $k \leq n+1$ for otherwise it would follow that $(d_k, d_{k+1}), (d_{k+1}, d_{k+2}), \ldots, (d_{n-1}, d_n), (d_n, d_k) \in \mathfrak{I}(P)$: but then by $\mathfrak{M} \models \forall x \forall y \forall z(P(x,y) \wedge P(y,z) \rightarrow P(x,z))$ it would be the case that $(d_k, d_k) \in \mathfrak{I}(P)$, which would contradict $\mathfrak{M} \models \forall x \neg P(x,x)$.
     It follows that there are pairwise distinct $d_0, \ldots, d_{n+1}$ which are members of D and for which it is the case that $(d_0, d_1), (d_1, d_2), \ldots, (d_n, d_{n+1}) \in \mathfrak{I}(P)$.

   But this implies that for every n there are pairwise distinct members $d_0, \ldots, d_n$ of D. Hence, D is infinite. ■

5. Prove: there is no formula φ, such that for all models $\mathfrak{M} = (D, \mathfrak{I})$ and for all variable assignments s holds:

   $\mathfrak{M}, s \models \varphi$ if and only if D is infinite.

   (Hint: use the compactness theorem.)

   Proof. Assume for contradiction that there is such a φ:

   so, $\mathfrak{M}, s \models \neg\varphi$ iff D is finite (for arbitrary $\mathfrak{M}, s$).

Now consider the formula set $\Phi = \{\neg\varphi\} \cup \{\psi_{\geq n} | n \geq 2\}$, where

$$\psi_{\geq n} := \exists v_1 \ldots \exists v_n (\neg v_1 \equiv v_2 \wedge \ldots \wedge \neg v_i \equiv v_j \wedge \ldots \wedge \neg v_{n-1} \equiv v_n)$$

(for $i \neq j$ with $1 \leq i, j, \leq n$).

Note that $\mathfrak{M}, s \vDash \psi_{\geq n}$ if and only if D has n or more members (for arbitrary $\mathfrak{M}, s$).

It follows that every finite subset of Φ is satisfiable (by choosing a model with a sufficiently large domain). Therefore, the compactness theorem implies that Φ is satisfiable. So there are $\mathfrak{M}, s$ which satisfy Φ; fix one such model $\mathfrak{M} = (D, \mathfrak{I})$:

Since $\mathfrak{M}, s \models \neg\varphi$, D must be finite, i.e., for some n the set D is of cardinality n: but then $\mathfrak{M}, s \not\models \psi_{\geq n+1}$. However, $\psi_{\geq n+1} \in \Phi$ and Φ is satisfied by $\mathfrak{M}, s$, which implies $\mathfrak{M}, s \models \psi_{\geq n+1}$. Contradiction. ■