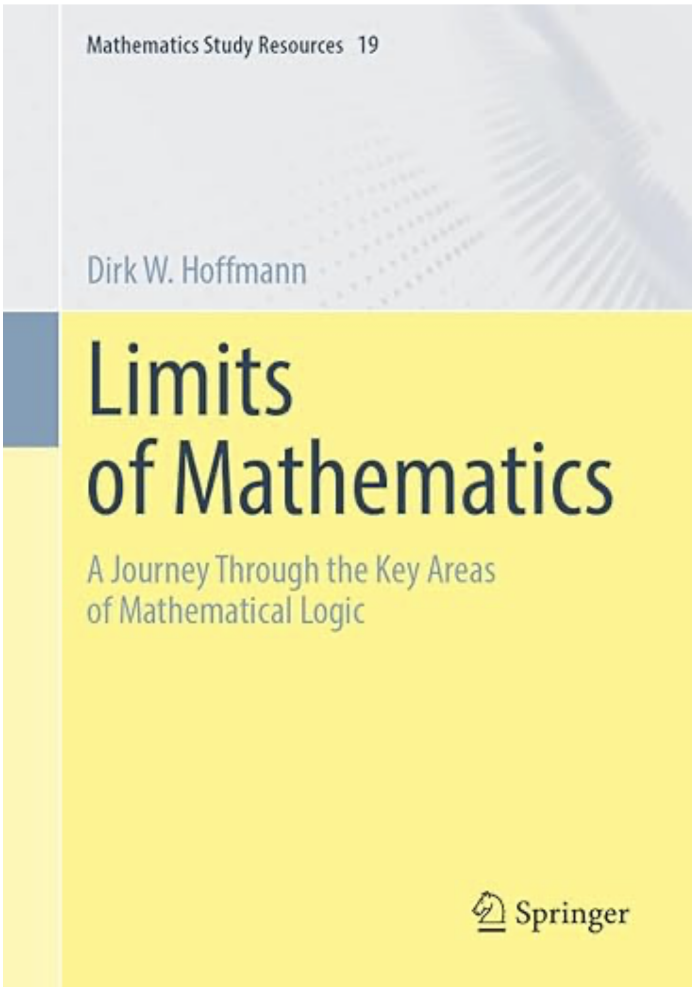
Solutions


Mathematics Study Resources 19
Dirk W. Hoffmann
Limits
of Mathematics
A Journey Through the Key Areas
of Mathematical Logic
Springer

Exercise 1.1

The axioms listed below are part of Frege's famous *Begriffsschrift*:

- Propositional logic

§14
$\vdash a \to (b \to a)$

§15
$\vdash (c \to (b \to a)) \to ((c \to b) \to (c \to a))$

§16
$\vdash (d \to (b \to a)) \to (b \to (d \to a))$

§17
$\vdash (b \to a) \to (\neg a \to \neg b)$

§18
$\vdash \neg\neg a \to a$

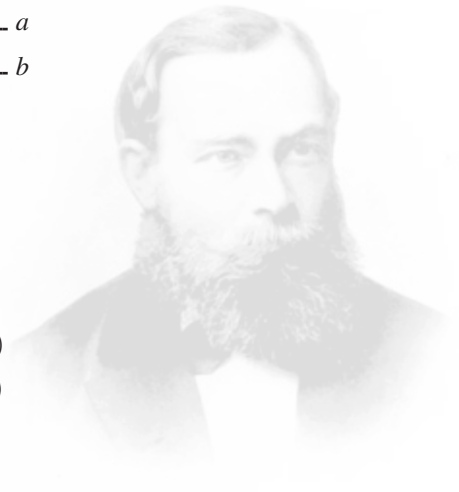
§19
$\vdash a \to \neg\neg a$

- Predicate logic

§20
$f(d) \vdash f(c) \to (c \equiv d)$

§21
$\vdash (c \equiv c)$

§22
$f(c) \vdash \mathfrak{a}.\ f(\mathfrak{a})$



Translate the formulas into contemporary notation.

§14 : $A \to (B \to A)$
§15 : $(C \to (B \to A)) \to ((C \to B) \to (C \to A))$
§16 : $(D \to (B \to A)) \to (B \to (D \to A))$
§17 : $(B \to A) \to (\neg A \to \neg B)$
§18 : $\neg\neg A \to A$
§19 : $A \to \neg\neg A$
§20 : $c = d \to (F(c) \to F(d))$
§21 : $c = c$
§22 : $\forall x\, F(x) \to F(c)$

Exercise 1.2

We have worked out in Section 1.2.2 how Cantor proved the countability of algebraic numbers in his 1874 publication. For this purpose, he assigned each equation of the form

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

a *height* N, which was defined as follows:

$$N = n - 1 + |a_n| + \ldots + |a_3| + |a_2| + |a_1| + |a_0|$$

Was Cantor constrained to choose the definition in this form, or could he have replaced it with one of the simplified definitions below?

Note: For Cantor's proof to work, the following must apply:

1) There are only finitely many algebraic equations of height N.

2) An algebraic equation of height N has at most N solutions.

a) $N = n$

No, condition 1) is not met.

b) $N = |a_n| + \ldots + |a_3| + |a_2| + |a_1| + |a_0|$

No, condition 2) is not met.

c) $N = n + |a_n| + \ldots + |a_3| + |a_2| + |a_1| + |a_0|$

Yes, this approach fulfills both 1) and 2), so Cantor could have used it to carry out his proof. Cantor had likely integrated the subtraction of 1 for reasons of mathematical aesthetics. This makes N a genuine upper bound for the degree of the polynomial. The bound is achieved with the polynomial x^n, which has the following height:

$$N = n - 1 + |a_n| = n - 1 + 1 = n$$

For the proof, it is irrelevant whether N is only an upper bound or a true upper bound for the degree of the algebraic equation. It is only relevant that the number of solutions does not exceed the height of the equation.

Exercise 1.3

In Section 1.2.1, you have learned how to turn the periodic decimal number $0.0\overline{238095}$ into the fraction $\frac{1}{42}$.

a) Utilize this method to prove the relationship $1 = 0.\overline{9}$.

To express the number

$$x = 0,\overline{9}$$

in the form $\frac{p}{q}$, we multiply both sides by 10:

$$10x = 9,\overline{9}$$

If we now subtract the first equation from the second, the periodic component disappears:

$$9x = 9$$

So for x we get the result:

$$x = \frac{9}{9} = 1$$

b) Did the trick used in the transformation make you feel uncomfortable? If so, you already have a good sense of the dangers of dealing with the actual infinite. Try to prove the relationship $1 = 0.\overline{9}$ without interpreting an infinite sequence of decimal digits as a closed entity.

The rational number represented by the infinite sequence $0,\overline{9}$ is formally defined as the limit of the partial sums

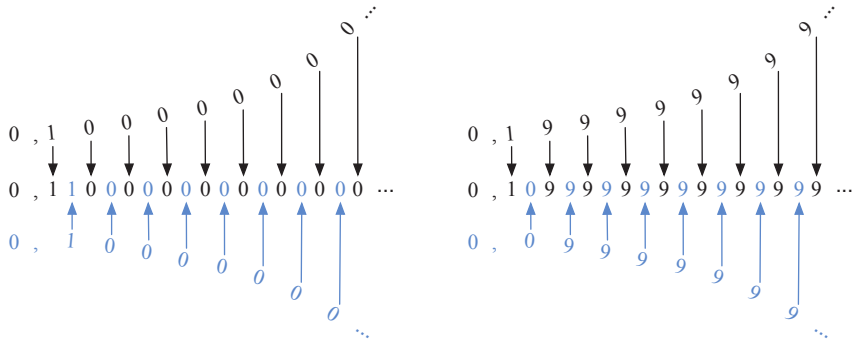
$$\sum_{i=1}^{n} 9 \cdot 10^{-i}.$$

Thus, we have:

$$\lim_{n\to\infty} \sum_{i=1}^{n} 9 \cdot 10^{-i} = 1$$

Exercise 1.4

Figure 1.20 shows how to merge two real numbers into a single number using the zipper method. The construction results in a bijective mapping between $\mathbb{R}^2$ and $\mathbb{R}$, thereby proving the equality of both sets.



In fact, we have been cheating here a little, as the representation of a real number as a decimal fraction is sometimes ambiguous. For instance, $0.11 = 0.10\overline{9}$.

Consequently, the constructed mapping between $\mathbb{R}^2$ and $\mathbb{R}$ is not injective, thus, a fortiori, not bijective. Solve this problem without abandoning the basic idea of zipper construction.

We have shown that the real numbers can be bijectively mapped onto the interval $(-1;1)$. Using the same arguments, it can be shown that there is a bijection between $\mathbb{R}$ and $(0;1)$. We use these facts to simplify this task by assuming that we only deal with real numbers x and y with $0 < x, y < 1$.

Before interleaving the digit sequences of x and y like a zipper, we do the following:

1. We eliminate infinitely long chains of zeros
   Based on the example numbers shown, this means:

$$0{,}100000000\ldots \to 0{,}099999999\ldots$$
$$0{,}100000000\ldots \to 0{,}099999999\ldots$$

$$0{,}199999999\ldots \to 0{,}199999999\ldots \quad \text{(nichts zu tun)}$$
$$0{,}099999999\ldots \to 0{,}099999999\ldots \quad \text{(nichts zu tun)}$$

2. We form blocks. Each digit $\neq 0$ is a block, and the zeros are assigned to other blocks.

$$0{,}099999999\ldots \to 0{,}|09|9|9|9|9|9|9|9\ldots$$
$$0{,}099999999\ldots \to 0{,}|09|9|9|9|9|9|9|9\ldots$$

$$0{,}199999999\ldots \to 0,|1|9|9|9|9|9|9|9|9\ldots$$
$$0{,}099999999\ldots \to 0,|09|9|9|9|9|9|9|9\ldots$$

3. Now the blocks are fused together like a zipper:

$$\left.\begin{array}{l} 0,|09|9|9|9|9|9|9|9\ldots \\ 0,|09|9|9|9|9|9|9|9\ldots \end{array}\right\} \quad 0{,}090999999999\ldots$$

$$\left.\begin{array}{l} 0,|1|09|9|9|9|9|9|9\ldots \\ 0,|09|9|9|9|9|9|9|9\ldots \end{array}\right\} \quad 0{,}109099999999\ldots$$

This way, we have established a bijective mapping between $(0;1)^2$ and $(0;1)$.

The described difficulties with the zipper method have historical roots. It was first discussed in a correspondence between Cantor and Dedekind. You can find more about this in Herbert Meschkowsky's beautifully written book *Probleme des Unendlichen*, 1983, Bibliographisches Institut Mannheim.

Exercise 1.5

With *Goldbach's conjecture*, you have become acquainted with one of the most prominent yet unsolved problems in number theory. Its strong form reads like this:

"Every even natural number $n > 2$ can be written as the sum of two prime numbers."

Show that the strong Goldbach conjecture implies the following variant:

"Every odd natural number $n > 5$ can be written as the sum of three prime numbers."

Let us decompose n as follows:

$$n = (n-3)+3$$

Since n is an odd number, $n-3$ must be even. This means that $n-3$ can be broken down into two prime numbers according to Goldbach's strong conjecture, and the assertion follows.

Exercise 1.6

This exercise deals again with Goldbach's conjecture.

a) Suppose the conjecture is wrong. In that case, could it be disproved using ordinary arithmetic?

Yes. If it were wrong, a counterexample could be constructed.

b) Assume that Goldbach's conjecture is unprovable with the methods of ordinary mathematics. In this case, may we conclude that the conjecture is true or false?

Yes. If it were wrong, it could be refuted. If it cannot be proven, it must be true.

c) Can the result be carried over to the conjecture about infinitely many twin primes?

No. In this case, falsity cannot be shown by a counterexample. Therefore, we cannot easily conclude its truth or falsity from its unprovability.

d) Is Fermat's conjecture a mathematical theorem of Goldbach type?

Yes. Here, too, a statement is made in the form "For all natural numbers, ... applies".

Exercise 1.7

The *Erdős-Straus conjecture* states that the equation

$$\frac{4}{n} = \frac{1}{a} + \frac{1}{b} + \frac{1}{c}$$

has a solution in the natural numbers for every natural number $n > 1$.

Could we solve the conjecture with a decision procedure for Diophantine equations?

For each specific value of n, the conjecture could be decided. It can be transformed into

$$4abc = n(bc + ac + ab)$$

and further into the following Diophantine equation:

$$4abc - nbc - nac - nab = 0$$

Be careful though: Diophantine equations are usually solved in the integers. Here, we are looking for solutions in $\mathbb{N}^+$! We cannot simply assume that a decision procedure for solvability in $\mathbb{Z}$ can be used to prove Erdös's Conjecture. However, as we will show in 5, this works. A procedure that decides solvability in $\mathbb{Z}$ can always also be used to decide solvability in $\mathbb{N}$ and vice versa. The same applies to $\mathbb{N}^+$.

Exercise 1.8

Supplement the statements below.

The set…	empty	finite	countable	uncountable
$\{M \in \mathcal{P}(\mathbb{N}) \mid \mathbb{N} \subseteq M\}$ ist	◯	✔	◯	◯
$\{M \in \mathcal{P}(\mathbb{N}) \mid \lvert M\rvert = \lvert\mathbb{N}\rvert\}$ ist	◯	◯	◯	✔
$\{M \in \mathcal{P}(\mathbb{N}) \mid \lvert M\rvert < \lvert\mathbb{N}\rvert\}$ ist	◯	◯	✔	◯
$\{M \in \mathcal{P}(\mathbb{N}) \mid \lvert M\rvert > \lvert\mathbb{N}\rvert\}$ ist	✔	✔	◯	◯

Notes:

a) $\{M \in \mathcal{P}(\mathbb{N}) \mid \mathbb{N} \subseteq M\}$

The set is finite but not empty. It contains precisely one element, namely $\mathbb{N}$.

b) $\{M \in \mathcal{P}(\mathbb{N}) \mid |M| = |\mathbb{N}|\}$

The set is uncountable. There are uncountably many subsets of $\mathbb{N}$ with infinitely many elements.

c) $\{M \in \mathcal{P}(\mathbb{N}) \mid |M| < |\mathbb{N}|\}$

The set is countable. There are infinitely many finite subsets of $\mathbb{N}$, which we can, however, enumerate one after the other.

d) $\{M \in \mathcal{P}(\mathbb{N}) \mid |M| > |\mathbb{N}|\}$

The power set is uncountable, but it does not contain any uncountable elements. Consequently, the set is empty and therefore definitely finite.

Exercise 2.1

Through the formal systems $E, E_2, \ldots, E_6$, we have explored the basic properties of formal systems.

a) Recap the characteristics of the different formal systems by completing the table below:

	E	E_2	E_3	E_4	E_5	E_6
Consistent	✔	✔	✘	✔	✔	✔
Negation complete	✘	✘	✔	✔	✔	✔
Correct	✔	✔	✘	✔	✘	✔
Complete	✘	✘	✔	✔	✘	✔

b) Suppose the symbol '$>$' is no longer interpreted as "greater" but as "greater or equal". Does the system E_4 remain consistent, negation complete, correct, and complete?

The syntactic properties of consistency and negation completeness do not change, since they are independent of how we interpret the symbols. However, the calculus is no longer correct, since the formula $\neg(0 > 0)$ is derivable. Under the new meaning of '$>$', this formula is false. The calculus is no longer complete either, since the formula $0 > 0$, which has now become true, is not derivable from the axioms.

c) Is it possible to change the interpretations of the symbols '$>$' and '$=$' to turn E_3 into a correct formal system?

No. In E3, both $(0 = 0)$ and $\neg(0 = 0)$ are derivable. Both formulas can never be true at the same time. Thus, no consistent interpretation can exist for the symbols (in short: contradictory calculi can never be correct).

Exercise 2.2
✍✍✍

Consider the following five axioms defining a set of properties of two classes, K and L. They are taken from [142] and retained in their colloquial form:

1. Any two members of K are contained in just one member of L.
2. No member of K is contained in more than two members of L.
3. The members of K are not all contained in a single member of L.
4. Any two members of L contain just one member of K.
5. No member of L contains more than two members of K.

Various consequences can be drawn from the axioms using the standard mathematical rules of inference. Is it possible to derive a contradiction? How could a formal proof of consistency be conducted?

No, a contradiction cannot be deduced. To see this, let us give L and K a concrete meaning:

$$K = \text{The three vertices of a triangle}$$
$$L = \text{The three point sets that form the sides of the triangle}$$

Under the chosen interpretation, all axioms become true statements. Assuming that the usual rules of inference in mathematics can only derive true statements from true statements, the consistency of the axiomatic system is proven.

Exercise 2.3

In this exercise, we consider four formal systems over a rudimentary language. In total, 42 formulas can be formed by instantiating the formulas $\varphi_1(\xi)$, $\varphi_2(\xi)$, $\varphi_3(\xi)$, $\neg\varphi_1(\xi)$, $\neg\varphi_2(\xi)$, and $\neg\varphi_3(\xi)$ with the terms $\overline{0},\ldots,\overline{6}$. The following matrix indicates which of the 42 statements are true and which are false:

	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$
$\varphi_1(\xi)$	$\models$	$\not\models$	$\models$	$\not\models$	$\models$	$\not\models$	$\not\models$
$\varphi_2(\xi)$	$\models$	$\models$	$\models$	$\models$	$\models$	$\models$	$\models$
$\varphi_3(\xi)$	$\models$	$\models$	$\not\models$	$\models$	$\models$	$\not\models$	$\models$
$\neg\varphi_1(\xi)$	$\not\models$	$\models$	$\not\models$	$\models$	$\not\models$	$\models$	$\models$
$\neg\varphi_2(\xi)$	$\not\models$	$\not\models$	$\not\models$	$\not\models$	$\not\models$	$\not\models$	$\not\models$
$\neg\varphi_3(\xi)$	$\not\models$	$\not\models$	$\models$	$\not\models$	$\not\models$	$\models$	$\not\models$

The axioms and inference rules of the four calculi are unknown. However, we possess a matrix for each formal system, indicating which formulas can be derived from the axioms and which cannot. State, for each calculus, whether it is complete, correct, consistent, and negation complete.

K_1	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$
$\varphi_1(\xi)$	$\vdash$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$	$\nvdash$	$\nvdash$
$\varphi_2(\xi)$	$\vdash$	$\vdash$	$\vdash$	$\vdash$	$\vdash$	$\vdash$	$\vdash$
$\varphi_3(\xi)$	$\vdash$	$\vdash$	$\nvdash$	$\vdash$	$\vdash$	$\nvdash$	$\vdash$
$\neg\varphi_1(\xi)$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$	$\vdash$
$\neg\varphi_2(\xi)$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$
$\neg\varphi_3(\xi)$	$\nvdash$	$\nvdash$	$\vdash$	$\nvdash$	$\nvdash$	$\vdash$	$\nvdash$

The calculus is complete, correct, consistent and negation-complete.

K_2	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$
$\varphi_1(\xi)$	$\vdash$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$	$\nvdash$	$\nvdash$
$\varphi_2(\xi)$	$\vdash$	$\vdash$	$\vdash$	$\vdash$	$\nvdash$	$\vdash$	$\vdash$
$\varphi_3(\xi)$	$\vdash$	$\vdash$	$\nvdash$	$\vdash$	$\vdash$	$\nvdash$	$\vdash$
$\neg\varphi_1(\xi)$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$	$\vdash$
$\neg\varphi_2(\xi)$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$
$\neg\varphi_3(\xi)$	$\nvdash$	$\nvdash$	$\vdash$	$\nvdash$	$\nvdash$	$\vdash$	$\nvdash$

Due to $\nvdash \varphi_2(\overline{4})$, the calculus is no longer complete and no longer negation-complete.

K_3	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$
$\varphi_1(\xi)$	$\vdash$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$
$\varphi_2(\xi)$	$\vdash$	$\vdash$	$\vdash$	$\vdash$	$\vdash$	$\vdash$	$\vdash$
$\varphi_3(\xi)$	$\vdash$	$\vdash$	$\nvdash$	$\vdash$	$\vdash$	$\nvdash$	$\vdash$
$\neg\varphi_1(\xi)$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$	$\vdash$
$\neg\varphi_2(\xi)$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$
$\neg\varphi_3(\xi)$	$\nvdash$	$\nvdash$	$\vdash$	$\nvdash$	$\nvdash$	$\vdash$	$\nvdash$

Due to $\vdash \varphi_1(\overline{6})$, the calculus is no longer correct and no longer consistent.

K_4	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$
$\varphi_1(\xi)$	$\vdash$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$	$\nvdash$	$\nvdash$
$\varphi_2(\xi)$	$\vdash$	$\vdash$	$\vdash$	$\vdash$	$\vdash$	$\vdash$	$\vdash$
$\varphi_3(\xi)$	$\vdash$	$\vdash$	$\vdash$	$\vdash$	$\vdash$	$\nvdash$	$\vdash$
$\neg\varphi_1(\xi)$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$	$\nvdash$	$\vdash$	$\vdash$
$\neg\varphi_2(\xi)$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$
$\neg\varphi_3(\xi)$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\nvdash$	$\vdash$	$\nvdash$

Due to $\vdash \varphi_3(\overline{2})$ and $\nvdash \neg\varphi_3(\overline{2})$, the calculus is no longer complete and no longer correct. However, it is still consistent and negation-complete.

Exercise 2.4

Let the logic calculi P_1 and P_2 be defined by the following axioms and inference rules:

Axioms (calculus P_1)		Axioms (calculus P_2)	
$(11,110)$	(A1)	$(01,011)$	(A1')

Inference rules (calculus P_1)		Inference rules (calculus P_2)	
$\dfrac{(\varphi,\psi)}{(\varphi 011,\psi 100)}$	(S1)	$\dfrac{(\varphi,\psi)}{(\varphi 001,\psi 0)}$	(S1')
$\dfrac{(\varphi,\psi)}{(\varphi 11,\psi 110)}$	(S2)	$\dfrac{(\varphi,\psi)}{(\varphi 01,\psi 011)}$	(S2')
$\dfrac{(\varphi,\psi)}{(\varphi 010,\psi 011)}$	(S3)	$\dfrac{(\varphi,\psi)}{(\varphi 01,\psi 101)}$	(S3')
		$\dfrac{(\varphi,\psi)}{(\varphi 10,\psi 001)}$	(S4')

Both calculi operate on the same basic principle. The axiom specifies a pair of binary strings, and the inference rules govern how to extend those strings successively. Both calculi differ only in the binary substrings hard-coded into the axiom and the inference rules.

a) To which calculus belongs the following proof? On the right-hand side, indicate the applied inference rule for each derivation step.

1. $\vdash (01,011)$ (A1')
2. $\vdash (0110,011001)$ (S4')
3. $\vdash (011001,011001101)$ (S3')
4. $\vdash (01100110,011001101001)$ (S4')
5. $\vdash (0110011010,011001101001001)$ (S4')

b) Can a theorem of the form (φ,φ) be derived in P_1 and P_2?

Inside K_1 such a theorem is not derivable. To obtain a theorem of the form (x,x) we have to start with the axiom $(11,110)$. Since the left string is shorter than the right one, the right string must eventually catch up with the missing character. However, this is impossible because the right side contains at least as many characters as the left side in all word pairs.

Within K_2, the left and right sides of the axiom can actually be aligned. For example, a theorem of K_2 is

(01 10 01 10 10 01 001 01 10 01 10 01 10 10 01 10 10
01 001 10 10 01 001 01 10 001 001 01 10 10 10 01 001
01 001 001 001 01 10 01 10 001 01 001 10 10 01 001 10
001 001 01 10 001 001 01 001 001 01 001 01 001 10 001
001 01 ,

011 001 101 001 001 011 0 011 001 101 001 101 001 001
101 001 001 011 0 001 001 011 0 101 001 0 0 101 001
001 001 011 0 011 0 0 0 101 001 101 001 0 011 0 001
001 011 0 001 0 0 101 001 0 0 101 0 0 101 0 011 0
001 0 0 101)

Source for this derivation sequence: Uwe Schöning: *"Theoretische Informatik - kurz gefasst"*, Spektrum Akademischer Verlag, 2008

c) Is there a decision procedure for P_1 and P_2?

Yes! In each step, the derived sequence is extended. If an expression (φ, ψ) is given and n denotes its length, it must be derivable in less than n steps. Since there are only a finite number of such derivations, they can all be analyzed one after the other.

d) Does a procedure exist that decides for all calculi of the above type whether a theorem of the form (φ, φ) can be derived?

No. In that case, we would have solved the famous Post correspondence problem. It is one of the most well-known undecidable problems and plays a major role in the field of theoretical computer science.

Exercise 2.5

Complete the truth tables of the following propositional logic formulas. Are the formulas satisfiable, universally valid or unsatisfiable?

- $\varphi_1 = (\neg A \vee B) \wedge (\neg B \vee C) \wedge (\neg C \vee A)$ (satisfiable, not universally valid)

A	B	C	$\neg A \vee B$	$\neg B \vee C$	$\neg C \vee A$	$(\neg A \vee B) \wedge (\neg B \vee C)$	φ_1
0	0	0	1	1	1	1	1
0	0	1	1	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	1	1	0	1	0
1	0	0	0	1	1	0	0
1	0	1	0	1	1	0	0
1	1	0	1	0	1	0	0
1	1	1	1	1	1	1	1

- $\varphi_2 = (A \to B) \wedge (B \to C) \to (A \to C)$ (universally valid)

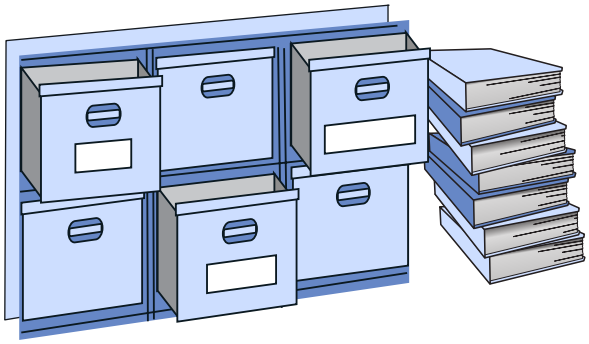
A	B	C	$A \to B$	$B \to C$	$A \to C$	$(A \to B) \wedge (B \to C)$	φ_2
0	0	0	1	1	1	1	1
0	0	1	1	1	1	1	1
0	1	0	1	0	1	0	1
0	1	1	1	1	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	1	0	1
1	1	0	1	0	0	0	1
1	1	1	1	1	1	0	1

- $\varphi_3 = (A \nleftrightarrow B) \wedge (B \nleftrightarrow C) \wedge (A \nleftrightarrow C)$ (unsatisfiable)

A	B	C	$A \nleftrightarrow B$	$B \nleftrightarrow C$	$A \nleftrightarrow C$	$(A \nleftrightarrow B) \wedge (B \nleftrightarrow C)$	φ_3
0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	0
0	1	0	1	1	0	1	0
0	1	1	1	0	1	0	0
1	0	0	1	0	1	0	0
1	0	1	1	1	0	1	0
1	1	0	0	1	1	0	0
1	1	1	0	0	0	0	0

Exercise 2.6

Dirichlet's drawer principle is named after the German mathematician Peter Gustav Lejeune Dirichlet. It states that a finite set M cannot be mapped injectively to a set N if N contains fewer elements than M. We are all familiar with the drawer principle from everyday life. If $m > n$ and we distribute m objects to n drawers, then at least one drawer must contain more than one object. Dirichlet's drawer principle is usually called the *pigeonhole principle* in the English-speaking world. The reasoning stays the same: If m pigeons are distributed over n pigeonholes and $m > n$, at least one pigeonhole must be occupied more than once.



Peter Gustav Lejeune Dirichlet
(1805 – 1859)

Formalize Dirichlet's drawer principle for n objects and $n-1$ drawers. For this purpose, introduce a propositional variable A_{ij} for each possible combination of objects and drawers, which equals 1 if and only if the i-th object is in the j-th drawer.

1) "The i-th element is in one of the drawers"

$$\bigvee_{j=1}^{n-1} \mathsf{A}_{ij}$$

2) "Each element is in one of the drawers"

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{n-1} \mathsf{A}_{ij}$$

3) "The i-th and k-th element are both in drawer j"

$$A_{ij} \wedge A_{kj}$$

4) "There are at least two elements in drawer j"

$$\bigvee_{i=1}^{n-1} \bigvee_{k=i+1}^{n} (\mathsf{A}_{ij} \wedge \mathsf{A}_{kj})$$

5) “There are at least two elements in a drawer”

$$\bigvee_{j=1}^{n-1} \bigvee_{i=1}^{n-1} \bigvee_{k=i+1}^{n} (A_{ij} \wedge A_{kj})$$

Connecting the formulas with the implication operator, we obtain the statement of the pigeonhole principle: If n elements are placed in $n-1$ drawers, then one of the drawers contains at least two elements:

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{n-1} A_{ij} \rightarrow \bigvee_{j=1}^{n-1} \bigvee_{i=1}^{n-1} \bigvee_{k=i+1}^{n} (A_{ij} \wedge A_{kj})$$

Exercise 2.7

The propositional quantifiers '$\forall$' and '$\exists$' are defined as follows:

$$\forall x\, \varphi \;:=\; \varphi[x \leftarrow 0] \wedge \varphi[x \leftarrow 1]$$
$$\exists x\, \varphi \;:=\; \varphi[x \leftarrow 0] \vee \varphi[x \leftarrow 1]$$

a) How do these quantifiers differ from those used in predicate logic?

In predicate logic, '$\forall$' and '$\exists$' quantify individual objects, i.e. elements of the domain. In PL0, there are no individuals, and propositional variables are nothing more than 0-ary predicates. In short: The propositional quantifiers quantify over (0-ary) predicates and the predicate quantifiers over individual elements.

b) Let φ be a PL0 formula with x being the only variable. Check the veracity of the following statements:

$$\exists x\, \varphi \equiv 1 \;\Leftrightarrow\; \varphi \text{ is satisfiable} \qquad \forall x\, \varphi \equiv 1 \;\Leftrightarrow\; \varphi \text{ is universally true}$$
$$\neg\exists x\, \varphi \equiv 1 \;\Leftrightarrow\; \varphi \text{ is unsatisfiable} \qquad \neg\forall x\, \varphi \equiv 1 \;\Leftrightarrow\; \varphi \text{ ist unsatisfiable}$$

All statements except the last one are correct.

c) Which of the following equivalences are valid?

$$\forall x\,(\varphi \wedge \psi) \;\equiv\; \forall x\, \varphi \wedge \forall x\, \psi$$

The statement is true:

$$\begin{aligned}
\forall x\,(\varphi \wedge \psi) &\equiv (\varphi \wedge \psi)[x \leftarrow 0] \wedge (\varphi \wedge \psi)[x \leftarrow 1] \\
&\equiv (\varphi[x \leftarrow 0] \wedge \psi[x \leftarrow 0]) \wedge (\varphi[x \leftarrow 1] \wedge \psi[x \leftarrow 1]) \\
&\equiv (\varphi[x \leftarrow 0] \wedge \varphi[x \leftarrow 1]) \wedge (\psi[x \leftarrow 0] \wedge \psi[x \leftarrow 1]) \\
&\equiv \forall x\, \varphi \wedge \forall x\, \psi
\end{aligned}$$

$$\forall x\,(\varphi \vee \psi) \;\equiv\; \forall x\, \varphi \vee \forall x\, \psi$$

The statement is false:

$$\begin{aligned}
\forall x\,(\varphi \vee \psi) &\equiv (\varphi \vee \psi)[x \leftarrow 0] \wedge (\varphi \vee \psi)[x \leftarrow 1] \\
&\equiv (\varphi[x \leftarrow 0] \vee \psi[x \leftarrow 0]) \wedge (\varphi[x \leftarrow 1] \vee \psi[x \leftarrow 1]) \\
&\not\equiv (\varphi[x \leftarrow 0] \wedge \varphi[x \leftarrow 1]) \vee (\psi[x \leftarrow 0] \wedge \psi[x \leftarrow 1]) \\
&\equiv \forall x\, \varphi \vee \forall x\, \psi
\end{aligned}$$

$$\exists x\,(\varphi \wedge \psi) \;\equiv\; \exists x\,\varphi \wedge \exists x\,\psi$$

The statement is false:

$$\begin{aligned}
\exists x\,(\varphi \wedge \psi) &\equiv (\varphi \wedge \psi)[x \leftarrow 0] \vee (\varphi \wedge \psi)[x \leftarrow 1] \\
&\equiv (\varphi[x \leftarrow 0] \wedge \psi[x \leftarrow 0]) \vee (\varphi[x \leftarrow 1] \wedge \psi[x \leftarrow 1]) \\
&\not\equiv (\varphi[x \leftarrow 0] \vee \varphi[x \leftarrow 1]) \wedge (\psi[x \leftarrow 0] \vee \psi[x \leftarrow 1]) \\
&\equiv \exists x\,\varphi \wedge \forall x\,\psi
\end{aligned}$$

$$\exists x\,(\varphi \vee \psi) \;\equiv\; \exists x\,\varphi \vee \exists x\,\psi$$

The statement is true:

$$\begin{aligned}
\exists x\,(\varphi \vee \psi) &\equiv (\varphi \vee \psi)[x \leftarrow 0] \vee (\varphi \vee \psi)[x \leftarrow 1] \\
&\equiv (\varphi[x \leftarrow 0] \vee \psi[x \leftarrow 0]) \vee (\varphi[x \leftarrow 1] \vee \psi[x \leftarrow 1]) \\
&\equiv (\varphi[x \leftarrow 0] \vee \varphi[x \leftarrow 1]) \vee (\psi[x \leftarrow 0] \vee \psi[x \leftarrow 1]) \\
&\equiv \exists x\,\varphi \vee \forall x\,\psi
\end{aligned}$$

Exercise 2.8

Some time ago, I asked the following question in an exam:

> *Consider φ as a propositional logic formula containing the variable x, with no other variables present. Verify whether the following assertion holds true: "φ is either equivalent to the formula x or to the formula $\neg x$."*

Several students have solved the task roughly like this:

> The statement is correct. Proof: Symbolically, the statement "φ is either equivalent to the formula x or to the formula $\neg x$" is equivalent to:
>
> $$(\varphi \leftrightarrow x) \vee (\varphi \leftrightarrow \neg x) \equiv 1$$
>
> The statement can be proven with a few elementary transformations:
>
> $$\begin{aligned}(\varphi \leftrightarrow x) \vee (\varphi \leftrightarrow \neg x) &\equiv \neg\varphi\neg x \vee \varphi x \vee \neg\varphi x \vee \varphi\neg x \\ &\equiv \varphi(x \vee \neg x) \vee \neg\varphi(x \vee \neg x) \\ &\equiv \varphi \vee \neg\varphi \\ &\equiv 1\end{aligned}$$

The students have missed the existence of formulas such as $\varphi = x \vee \neg x$ or $\varphi = x \wedge \neg x$, which are neither equivalent to x nor to $\neg x$. Can you point out the error in their proof?

The established relationship

$$\varphi \text{ is equivalent to } x \text{ or equivalent to } \neg x \;\Leftrightarrow\; (\varphi \leftrightarrow x) \vee (\varphi \leftrightarrow \neg x) \equiv 1$$

is false. This becomes evident if we proceed step by step.

$$\varphi \text{ is equivalent to } x \;\Leftrightarrow\; \varphi \leftrightarrow x \equiv 1$$
$$\varphi \text{ is equivalent to } \neg x \;\Leftrightarrow\; \varphi \leftrightarrow \neg x \equiv 1$$

From this it follows that

$$\varphi \text{ is equivalent to } x \text{ or equivalent to } \neg x \;\Leftrightarrow\; \varphi \leftrightarrow x \equiv 1 \text{ or } \varphi \leftrightarrow \neg x \equiv 1$$

which is equivalent to

$$\varphi \text{ is equivalent to } x \text{ or equivalent to } \neg x \;\Leftrightarrow\; (\forall x\,(\varphi \leftrightarrow x) \vee \forall x\,(\varphi \leftrightarrow \neg x)) \equiv 1$$

In this equation, '$\forall$' and '$\exists$' are the quantifiers introduced in the previous exercise.

The students, on the other hand, have established the following equivalence, which differs from the derived formula:

$$\varphi \text{ is equivalent to } x \text{ or equivalent to } \neg x \;\Leftrightarrow\; \forall x\,((\varphi \leftrightarrow x) \vee (\varphi \leftrightarrow \neg x)) \equiv 1$$

Exercise 2.9

Russell and Whitehead built the propositional calculus of the *Principia Mathematica* upon the following five axioms:


PRINCIPIA MATHEMATICA

1. $\varphi \vee \varphi \to \varphi$ (Taut)
2. $\psi \to \varphi \vee \psi$ (Add)
3. $\varphi \vee \psi \to \psi \vee \varphi$ (Perm)
4. $\varphi \vee (\psi \vee \chi) \to \psi \vee (\varphi \vee \chi)$ (Assoc)
5. $(\psi \to \chi) \to (\varphi \vee \psi \to \varphi \vee \chi)$ (Sum)

Hilbert's student Paul Bernays has shown that the propositional axioms of the *Principia* are not independent, as the fourth axiom (Assoc) is derivable from the others.

The following derivation sequence is an adaptation of the original proof from the 1926 paper *Axiomatische Untersuchung des Aussagen-Kalküls der Principia Mathematica* [9], in which Paul Bernays summarized the results of his habilitation thesis from 1918.

1. $\vdash \chi \to \varphi \vee \chi$ (Add)
2. $\vdash (\chi \to \varphi \vee \chi) \to (\psi \vee \chi \to \psi \vee (\varphi \vee \chi))$ (Sum)
3. $\vdash \psi \vee \chi \to \psi \vee (\varphi \vee \chi)$ (MP, 1,2)
4. $\vdash (\psi \vee \chi \to \psi \vee (\varphi \vee \chi)) \to (\varphi \vee (\psi \vee \chi) \to \varphi \vee (\psi \vee (\varphi \vee \chi)))$ (Sum)
5. $\vdash \varphi \vee (\psi \vee \chi) \to \varphi \vee (\psi \vee (\varphi \vee \chi))$ (MP, 3,4)
6. $\vdash \varphi \vee (\psi \vee (\varphi \vee \chi)) \to (\psi \vee (\varphi \vee \chi)) \vee \varphi$ (Perm)
7. $\vdash \varphi \vee (\psi \vee \chi) \to (\psi \vee (\varphi \vee \chi)) \vee \varphi$ (MB, 5,6)
8. $\vdash \varphi \to \chi \vee \varphi$ (Add)
9. $\vdash \chi \vee \varphi \to \varphi \vee \chi$ (Perm)
10. $\vdash \varphi \to \varphi \vee \chi$ (MB, 8,9)
11. $\vdash \varphi \vee \chi \to \psi \vee (\varphi \vee \chi)$ (Add)
12. $\vdash \varphi \to \psi \vee (\varphi \vee \chi)$ (MB, 10,11)
13. $\vdash (\varphi \to \psi \vee (\varphi \vee \chi)) \to ((\psi \vee (\varphi \vee \chi)) \vee \varphi \to (\psi \vee (\varphi \vee \chi)) \vee (\psi \vee (\varphi \vee \chi)))$ (Sum)
14. $\vdash (\psi \vee (\varphi \vee \chi)) \vee \varphi \to (\psi \vee (\varphi \vee \chi)) \vee (\psi \vee (\varphi \vee \chi))$ (MP, 12,13)
15. $\vdash (\psi \vee (\varphi \vee \chi)) \vee (\psi \vee (\varphi \vee \chi)) \to \psi \vee (\varphi \vee \chi)$ (Taut)
16. $\vdash (\psi \vee (\varphi \vee \chi)) \vee \varphi \to \psi \vee (\varphi \vee \chi)$ (MB, 14,15)
17. $\vdash \varphi \vee (\psi \vee \chi) \to \psi \vee (\varphi \vee \chi)$ (MB, 7,16)

Try to reconstruct the partially printed proof.

Exercise 2.10

In Section 2.3.1, we have shown that every propositional logic formula can be rewritten to contain no other operators than '$\neg$' or '$\rightarrow$'. Because of this property, we refer to the set $\{\neg, \rightarrow\}$ as a *complete operator set*.

In this exercise, we consider the binary operators '$\overline{\wedge}$' (nand) and '$\overline{\vee}$' (nor) with:

$$I \models (\varphi \mathbin{\overline{\wedge}} \psi) \;:\Leftrightarrow\; I \not\models \varphi \text{ or } I \not\models \psi$$
$$I \models (\varphi \mathbin{\overline{\vee}} \psi) \;:\Leftrightarrow\; I \not\models \varphi \text{ and } I \not\models \psi$$

a) Prove that $\{ \overline{\wedge} \}$ and $\{ \overline{\vee} \}$ are complete operator sets.

We show that '$\neg$' and '$\rightarrow$' can be reduced to '$\overline{\wedge}$' and '$\overline{\vee}$':

$$\neg A \equiv A \mathbin{\overline{\wedge}} A$$
$$A \rightarrow B \equiv \neg A \vee B \equiv \neg(A \wedge \neg B) \equiv A \mathbin{\overline{\wedge}} (B \mathbin{\overline{\wedge}} B)$$

$$\neg A \equiv A \mathbin{\overline{\vee}} A$$
$$A \rightarrow B \equiv \neg A \vee B \equiv \neg((A \mathbin{\overline{\vee}} A) \mathbin{\overline{\vee}} B) \equiv ((A \mathbin{\overline{\vee}} A) \mathbin{\overline{\vee}} B) \mathbin{\overline{\vee}} ((A \mathbin{\overline{\vee}} A) \mathbin{\overline{\vee}} B)$$

b) Prove that $\{ \overline{\wedge} \}$ and $\{ \overline{\vee} \}$ are the only complete operator sets with a single element.

Let $\{\oplus\}$ be a complete operator system. We show $\oplus = \overline{\wedge}$ or $\oplus = \overline{\vee}$.

First, it is easy to see that $1 \oplus 1$ cannot be equivalent to 1. Otherwise, any expression would be equivalent to 1 if we assigned the value 1 to all variables. Thus, we have $1 \oplus 1 \equiv 0$. By the same argument, we can convince ourselves of the relationship $0 \oplus 0 \equiv 1$. The truth table of $A \oplus B$ must therefore look like this:

A	B	A ⊕ B
0	0	1
0	1	
1	0	
1	1	0

This leaves only 4 possibilities.

1. Case

A	B	A $\oplus$ B
0	0	1
0	1	0
1	0	0
1	1	0

In this case, we have $\oplus = \overline{\vee}$.

2. Case

A	B	A $\oplus$ B
0	0	1
0	1	0
1	0	1
1	1	0

Thus, we have $A \oplus B \equiv \neg B$. With that, $\{\neg\}$ would also be a complete operator system, which is obviously not the case.

3. Case

A	B	A $\oplus$ B
0	0	1
0	1	0
1	0	1
1	1	0

Thus, we have $A \oplus B \equiv \neg A$. With that, $\{\neg\}$ would also be a complete operator system, which is obviously not the case.

4. Fall

A	B	A $\oplus$ B
0	0	1
0	1	1
1	0	1
1	1	0

In this case, we have $\oplus = \overline{\wedge}$.

Exercise 2.11 Prove the following formulas in the propositional logic calculus:

a) $(\varphi \to (\psi \to \chi)) \to (\psi \to (\varphi \to \chi))$

1. $\{\varphi \to (\psi \to \chi)\} \vdash \varphi \to (\psi \to \chi)$ (Satz 2.4)
2. $\{\varphi \to (\psi \to \chi), \varphi\} \vdash \psi \to \chi$ (DT)
3. $\{\varphi \to (\psi \to \chi), \varphi, \psi\} \vdash \chi$ (DT)
4. $\{\varphi \to (\psi \to \chi), \psi\} \vdash \varphi \to \chi$ (DT)
5. $\{\varphi \to (\psi \to \chi)\} \vdash \psi \to (\varphi \to \chi)$ (DT)
6. $\vdash (\varphi \to (\psi \to \chi)) \to (\psi \to (\varphi \to \chi))$ (DT)

b) $\varphi \to (\psi \to \neg(\varphi \to \neg\psi))$

1. $\vdash \varphi \to (\neg\neg\psi \to \neg(\varphi \to \neg\psi))$ (T7)
2. $\{\varphi\} \vdash \neg\neg\psi \to \neg(\varphi \to \neg\psi)$ (DT)
3. $\vdash \psi \to \neg\neg\psi$ (T5)
4. $\{\psi\} \vdash \neg\neg\psi$ (DT)
5. $\{\varphi, \psi\} \vdash \neg(\varphi \to \neg\psi)$ (MP, 2,4)
6. $\{\varphi\} \vdash \psi \to \neg(\varphi \to \neg\psi)$ (DT)
7. $\vdash \varphi \to (\psi \to \neg(\varphi \to \neg\psi))$ (DT)

Exercise 2.12

Section 2.5 has introduced first-order predicate logic with equality. Does this logic permit the construction of formulas with the following meanings?

$$(U,I) \models \varphi_{\geq n} \;\Leftrightarrow\; U \text{ contains at least } n \text{ elements}$$

$$\exists x_1 \ldots \exists x_n \bigwedge_{\substack{1 \leq i,j \leq n \\ i \neq j}} x_i \neq x_j$$

$$(U,I) \models \varphi_{\leq n} \;\Leftrightarrow\; U \text{ contains at most } n \text{ elements}$$

$$\exists x_1 \ldots \exists x_n \, \forall y \bigvee_{i=1}^{n} y = x_i$$

$$(U,I) \models \varphi_{=n} \;\Leftrightarrow\; U \text{ contains precisely } n \text{ elements}$$

$$\exists x_1 \ldots \exists x_n \bigwedge_{\substack{1 \leq i,j \leq n \\ i \neq j}} x_i \neq x_j \wedge \exists x_1 \ldots \exists x_n \, \forall y \bigvee_{i=1}^{n} y = x_i$$

Exercise 2.13

We call a relation R

- *reflexive*, if $R(x,x)$ holds for all x,
- *left-comparative*, if $R(x,y) \wedge R(x,z)$ implies $R(y,z)$,
- *symmetric*, if $R(x,y)$ implies $R(y,x)$.

a) Formalize the statement *"Every reflexive, left-comparative relation is symmetric"* in first-order predicate logic.

Formalization:

$$\mathsf{R(x,x)} \rightarrow ((\mathsf{R(x,y)} \wedge \mathsf{R(x,z)} \rightarrow \mathsf{R(y,z)}) \rightarrow (\mathsf{R(x,y)} \rightarrow \mathsf{R(y,x)}))$$

b) Try to derive a formal proof using the predicate-logic calculus. For the sake of simplicity, consider all propositional tautologies as already proven.

In the proof below, we use instances of the following two propositional tautologies:

$\varphi \rightarrow (\psi \rightarrow \varphi \wedge \psi)$ (TAUT 1)

$((\varphi \wedge \psi \rightarrow \chi) \wedge \psi) \rightarrow (\varphi \rightarrow \chi)$ (TAUT 2)

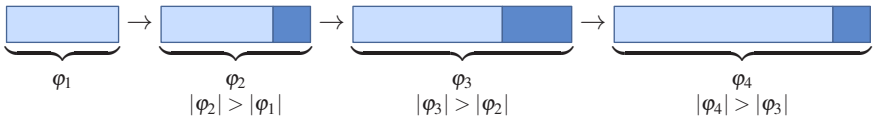
Let $M := \{\mathsf{R(x,x)}, \mathsf{R(x,y)} \wedge \mathsf{R(x,z)} \rightarrow \mathsf{R(y,z)}\}$

1. $M \vdash \mathsf{R(x,y)} \wedge \mathsf{R(x,z)} \rightarrow \mathsf{R(y,z)}$ (Theorem 2.4)
2. $M \vdash \forall \mathsf{z}\,(\mathsf{R(x,y)} \wedge \mathsf{R(x,z)} \rightarrow \mathsf{R(y,z)})$ (G, 1)
3. $\vdash \forall \mathsf{z}\,(\mathsf{R(x,y)} \wedge \mathsf{R(x,z)} \rightarrow \mathsf{R(y,z)}) \rightarrow (\mathsf{R(x,y)} \wedge \mathsf{R(x,x)} \rightarrow \mathsf{R(y,x)})$ (A4)
4. $M \vdash \mathsf{R(x,y)} \wedge \mathsf{R(x,x)} \rightarrow \mathsf{R(y,x)}$ (MP, 2,3)
5. $\vdash (\mathsf{R(x,y)} \wedge \mathsf{R(x,x)} \rightarrow \mathsf{R(y,x)}) \rightarrow (\mathsf{R(x,x)} \rightarrow (\mathsf{R(x,y)} \wedge \mathsf{R(x,x)} \rightarrow \mathsf{R(y,x)}) \wedge \mathsf{R(x,x)})$ (TAUT 1)
6. $M \vdash \mathsf{R(x,x)} \rightarrow (\mathsf{R(x,y)} \wedge \mathsf{R(x,x)} \rightarrow \mathsf{R(y,x)}) \wedge \mathsf{R(x,x)}$ (MP, 4,5)
7. $M \vdash \mathsf{R(x,x)}$ (Satz 2.4)
8. $M \vdash (\mathsf{R(x,y)} \wedge \mathsf{R(x,x)} \rightarrow \mathsf{R(y,x)}) \wedge \mathsf{R(x,x)}$ (MP, 6,7)
9. $\vdash ((\mathsf{R(x,y)} \wedge \mathsf{R(x,x)} \rightarrow \mathsf{R(y,x)}) \wedge \mathsf{R(x,x)}) \rightarrow (\mathsf{R(x,y)} \rightarrow \mathsf{R(y,x)})$ (TAUT 2)
10. $M \vdash \mathsf{R(x,y)} \rightarrow \mathsf{R(y,x)}$ (MP, 8,9)
11. $\mathsf{R(x,x)} \vdash (\mathsf{R(x,y)} \wedge \mathsf{R(x,z)} \rightarrow \mathsf{R(y,z)}) \rightarrow (\mathsf{R(x,y)} \rightarrow \mathsf{R(y,x)})$ (DT)
12. $\vdash \mathsf{R(x,x)} \rightarrow ((\mathsf{R(x,y)} \wedge \mathsf{R(x,z)} \rightarrow \mathsf{R(y,z)}) \rightarrow (\mathsf{R(x,y)} \rightarrow \mathsf{R(y,x)}))$ (DT)

Source:
Menzel, Schmitt: *Skript zur Vorlesung Formale Systeme*, WS 94/95, Universität Karlsruhe.

Exercise 2.14

Assume that the inference rules of a formal system ensure that the generated theorems become longer in each derivation step; that is, the conclusion always consists of more symbols than the premises.



Does such a system always have a decision procedure?

Yes. Since the theorems become longer in each derivation step, the derivation of a theorem of length n can take at most n steps. To determine whether a formula is a theorem, we only have to determine its length l and then examine all derivations that consist of at most l steps. The number of derivations is finite, so the procedure always terminates.

Exercise 2.15

Section 2.6 has demonstrated that second-order predicate logic is sufficiently expressive to define the concept of finiteness. The definition exploited the idea of claiming that every injective function is surjective. Is it also possible to define the concept by claiming that every surjective function is injective?

Yes. For every finite set M, every surjective function $f : M \to M$ must also be injective. If M is infinite, however, there are surjective functions $f : M \to M$ that are not injective. One such function is

$$f : \mathbb{N} \to \mathbb{N}$$

with

$$f(x) = \lfloor \frac{x}{2} \rfloor$$

Exercise 2.16

Name the mathematical concepts defined by the following formulas:

a) $\forall \mathsf{f}\,(\forall \mathsf{x}\,\exists \mathsf{y}\,(\mathsf{x} \doteq \mathsf{f}(\mathsf{y})) \rightarrow \forall \mathsf{x}\,\forall \mathsf{y}\,(\mathsf{f}(\mathsf{x}) \doteq \mathsf{f}(\mathsf{y}) \rightarrow \mathsf{x} \doteq \mathsf{y}))$

The formula states that every surjective function is also injective. It thus defines the notion *"finite"*.

b) $\exists \mathfrak{R}\,(\forall \mathsf{x}\,\forall \mathsf{y}\,\forall \mathsf{z}\,(\mathfrak{R}(\mathsf{x},\mathsf{y}) \wedge \mathfrak{R}(\mathsf{y},\mathsf{z}) \rightarrow \mathfrak{R}(\mathsf{x},\mathsf{z})) \wedge \forall \mathsf{x}\,(\neg \mathfrak{R}(\mathsf{x},\mathsf{x}) \wedge \exists \mathsf{y}\,\mathfrak{R}(\mathsf{x},\mathsf{y})))$

The formula states that there exists a transitive relation in which each element is related to another element but never to itself.

Such a relation exists if and only if the set of individuals is infinite. The formula defines the notion *"infinite"*.

Exercise 3.1

Formalize the following propositions within Peano arithmetic:

a) *"There exist natural numbers x and y satisfying $x^2 + y^2 = 9$."*

$\exists x\, \exists y\, x \times x + y \times y = 9$

b) *"$x^3 + y^3 = z^3$ has no solution in the positive natural numbers."*

$\neg \exists x\, \exists y\, \exists z\, (x \neq 0 \wedge y \neq 0 \wedge z \neq 0 \wedge x \times x \times x + y \times y \times y = z \times z \times z)$

c) *"x is a power of two."*

$\forall y\, (y | x \rightarrow (y = \overline{1} \vee \exists z\, y = \overline{2} \cdot z))$

Idea: A number is a power of two if and only if all its divisors (except 1) are multiples of 2.

Exercise 3.2

Which of the following formulas corresponds to the statement *"7 is a prime number"*?

Both statements do.

a) $\forall z\,(z \mid \overline{7} \to (z = \overline{1} \vee z = \overline{7}))$

A natural number $x > 1$ is a prime number if it is only divisible by 1 and itself.

b) $\neg\exists (y > 1)\,\exists (z > 1)\,\overline{7} = y \times z$

A natural number $x > 1$ is a prime number if it cannot be written as the product of two natural numbers greater than 1.

Which of the following formulas corresponds to the statement *"x is a prime number"*?

Neither of the two.

c) $\forall z\,((z \mid x) \to (z = \overline{1} \vee z = x))$

According to this formula, the number 1 would also be a prime number.

d) $\neg\exists (y > 1)\,\exists (z > 1)\,x = y \times z$

By this formula, 1 would again be a prime number.

Exercise 3.3

Formalize the following statements using Peano arithmetic:

a) *"Every even natural number $n > 2$ can be expressed as the sum of two prime numbers."*

$$\forall \mathsf{n}\ ((\mathrm{even}(\mathsf{n}) \wedge \mathrm{greaterTwo}(\mathsf{n})) \rightarrow \exists \mathsf{y}\ \exists \mathsf{z}\ (\mathrm{prime}(\mathsf{y}) \wedge \mathrm{prime}(\mathsf{z}) \wedge \mathsf{n} = \mathsf{y} + \mathsf{z})) \text{ mit}$$

$$\begin{aligned} \mathrm{even}(\mathsf{n}) &:= (\exists \mathsf{z}\ \mathsf{n} = \overline{2} \times \mathsf{z}) \\ \mathrm{greaterTwo}(\mathsf{n}) &:= (\neg \mathsf{n} = 0 \wedge \neg \mathsf{n} = \overline{1} \wedge \neg \mathsf{n} = \overline{2}) \end{aligned}$$

b) *"For infinitely many numbers n, both n and $n+2$ are prime numbers."*

$$\forall \mathsf{x}\ \exists \mathsf{n}\ \exists \mathsf{m}\ (\mathsf{n} > \mathsf{x} \wedge \mathsf{m} = \mathsf{n} + \overline{2} \wedge \mathrm{prime}(\mathsf{n}) \wedge \mathrm{prime}(\mathsf{m}))$$

Both statements are familiar to us from Chapter 1. The first is Goldbach's famous conjecture; the second is the conjecture about the existence of infinitely many prime twins.

Exercise 3.4

Some textbooks introduce the axiom of induction

$$\varphi(0) \to (\forall x\,(\varphi(x) \to \varphi(s(x))) \to \forall x\,\varphi(x))$$

in a slightly different form:

$$(\varphi(0) \wedge \forall x\,(\varphi(x) \to \varphi(s(x)))) \to \forall x\,\varphi(x)$$

Demonstrate the equivalence of both definitions.

Let

$$\begin{aligned}\varphi &:= \varphi(0)\\ \psi &:= \forall x\,(\varphi(x) \to \varphi(s(x)))\\ \chi &:= \forall x\,\varphi(x)\end{aligned}$$

The first variant can be transformed as follows:

$$\begin{aligned}&\varphi \to (\psi \to \chi)\\ \equiv\ &\varphi \to (\neg\psi \vee \chi)\\ \equiv\ &\neg\varphi \vee \neg\psi \vee \chi\end{aligned}$$

The second variant yields the same result:

$$\begin{aligned}&(\varphi \wedge \psi) \to \chi\\ \equiv\ &\neg(\varphi \wedge \psi) \vee \chi\\ \equiv\ &\neg\varphi \vee \neg\psi \vee \chi\end{aligned}$$

Exercise 3.5

At the top of the ZF axiom list, we introduced the axiom of extensionality. In formal notation, it reads as follows:

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

Suppose we remove the equals sign from the language and understand the axiom of extensionality not as a proper axiom but as the equal sign's definition. In doing so, we can only treat the equals sign as a syntactic abbreviation rather than a native language element. Can we still derive the same theorems as before?

Yes. The axiom of extensionality reduces the equality relation entirely to the element relation. Treating the symbol "=" as a syntactic abbreviation does not change the expressiveness of set theory. The fact that equality is nonetheless firmly integrated into the language is mainly for historical reasons. In the original Zermelo set theory, the axiom of extensionality was an axiom and not a definition. Some books take the "more modern" approach and exclude the equality symbol from the language (E.g. Mendelson: *Introduction to mathematical logic*). Here, the axiom of extensionality is indeed a definition rather than an axiom.

Exercise 3.6

At the fourth position of the ZF axiom list, we have introduced the axiom of the union:

$$\forall \mathsf{x}\, \exists \mathsf{y}\, \forall \mathsf{z}\, (\mathsf{z} \in \mathsf{y} \leftrightarrow \exists (\mathsf{w} \in \mathsf{x})\, \mathsf{z} \in \mathsf{w})$$

For two sets, x and y, it guarantees the existence of the union set $x \cup y$. In connection with this axiom, we have also introduced the notation $x \cap y$ as a syntactic abbreviation for the intersection. Is the existence of the intersection also guaranteed by the axiom of the union? If not, which axioms are additionally required?

We do not need the union axiom to prove the existence of the intersection. Its existence follows directly from the exclusion axiom. It guarantees that

$$\{z \in x \mid z \in x \wedge z \in y\}$$

is a set, which is the intersection $x \cap y$.

Exercise 3.7

The axiom of foundation is an indispensable part of ZF set theory. It states that every non-empty set x contains an element y that has no elements in common with x.

a) What is the significance of the axiom for ZF set theory?

The axiom of foundation replaces an older axiom introduced by Fraenkel, which narrows down the scope of sets to the smallest scope that is compatible with the remaining axioms. A direct consequence of the axiom of foundation is that no infinitely descending $\in$-chains can exist, eliminating the possibility of self-inclusion or ring-inclusion. The axiom of foundation is needed, for example, to legitimize $\in$-induction, which is a common proof method in set theory. With it, we can prove a statement about sets by assuming the validity of the statement for all elements of a set x and then showing that the statement is also valid for x.

b) Is the axiom compatible with Russell's ramified type theory?

Yes, it is always fulfilled there! Choose the element with the lowest type from x. This element can only contain elements whose type is lower. These elements cannot occur in x.

Exercise 3.8

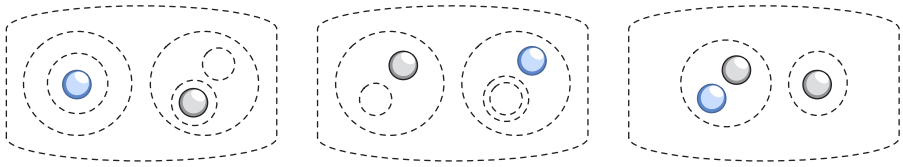
Section 3.2.1.3 has shown that the concept of an ordered pair can be captured with the following formula:

$$\langle \xi, \nu \rangle := \{ \{\xi\}, \{\xi, \nu\} \}$$

However, this representation is only one of many possibilities. For example, Norbert Wiener suggested in 1914 to represent ordered pairs as follows [150, 221, 222]:

$$\langle \xi, \nu \rangle := \{ \{\emptyset, \{\xi\}\}, \{\{\nu\}\} \}$$

a) Which of the following set diagrams visualizes the given definitions of the ordered pair?



Left: $\langle \xi, \nu \rangle := \{ \{\emptyset, \{\xi\}\}, \{\{\nu\}\} \}$

Right: $\langle \xi, \nu \rangle := \{ \{\xi\}, \{\xi, \nu\} \}$

b) What is the construction pattern of the remaining diagram? Is it suitable for representing ordered pairs as well?

Mid: $\langle \xi, \nu \rangle := \{ \{\emptyset, \xi\}, \{\{\emptyset\}, \nu\} \}$

Yes, the presentation also serves its purpose.

Exercise 3.9

We know from Section 3.2.1.2 that the relation '$<$' does not well-order $\mathbb{Z}$, the set of integers.

Redefine the ordering relation '$<$' such that $\mathbb{Z}$ becomes a well-ordered set.

A well-ordering can be created as follows:

$$x < y \;\Leftrightarrow\; |x| < |y| \vee (|x| = |y| \wedge x < y)$$

This produces the sequence

$$0 < -1 < 1 < -2 < 2 < -3 < 3 < \ldots$$

Exercise 3.10

The *Cartesian product* of two sets is defined as follows:

$$\nu \times \mu \;:=\; \{\langle x,y\rangle \mid x \in \nu \wedge y \in \mu\}$$

a) Formalize the Cartesian product in the system of ZF set theory by specifying a formula $\mathfrak{K}(\xi,\nu,\mu)$ that is true if and only if ξ,ν,μ satisfy $\xi = \nu \times \mu$.

$\mathfrak{K}(\xi,\nu,\mu) \;:=\; \forall x\,(x \in \xi \leftrightarrow \exists x_1\,\exists x_2\,(x = \langle x_1,x_2\rangle \wedge x_1 \in \nu \wedge x_2 \in \mu))$

b) Provide a formula $\mathfrak{R}(\xi,\nu)$ that is true precisely if ξ is a relation over the set ν. Base your definition on the formula $\mathfrak{K}(\xi,\nu,\mu)$ derived in Part a).

$\mathfrak{R}(\xi,\nu) \;:=\; \exists z\,(\mathfrak{K}(z,\nu,\nu) \wedge \xi \subseteq z)$

c) On page 159, you learned about the formula $\mathfrak{R}(\xi)$, which is true precisely if ξ is a relation over some set. Try to find an alternative definition for $\mathfrak{R}(\xi)$ that utilizes the notion of the Cartesian product.

Warning: If we could actually reduce $\mathfrak{R}(\xi)$ to $\mathfrak{K}$, what would we have to use for the variables ν and μ? If ξ is an arbitrary relation, then its elements are ordered pairs of arbitrary sets. μ and ν would be the set of all sets, which fortunately does not exist in ZF.

d) Building on the concept of relations, we have shown how to describe partial functions within Zermelo-Fraenkel set theory. Can total, injective, and surjective functions be formalized accordingly?

Yes. Let f be a function. Then the following relationships apply:

$$f \text{ is total} \Leftrightarrow f \text{ is a left-total and right-unique relation}$$
$$f \text{ is injective} \Leftrightarrow f \text{ is a left-unique and right-unique relation}$$
$$f \text{ is surjective} \Leftrightarrow f \text{ is a right-total and right-unique relation}$$

Exercise 3.11 ✍✍✍

In this exercise, we take on the formal proof of Theorem 3.3, establishing the component equality of ordered pairs.

a) Consider the proof steps in lines 59 - 64. Was it essential to prove the entire statement $x = y = u = v$?

No, the statement in the original proof is stronger than necessary. Instead of concluding $x = y = u = v$, it is sufficient to show $y = v$. Nothing else is needed in the proof.

b) The colloquially formulated original proof ends by distinguishing two cases. First, it considered the case $v \neq u$, then the case $v = u$. Could we have dispensed with this distinction in the formal proof?

Yes, we could have done without it, since the auxiliary theorem (H11) covers both cases. However, we will not be able to get rid of the case distinction altogether. It would reappear as soon as we tried to prove the auxiliary theorem.

Exercise 3.12

Which of the following sets are transitive? Which are ordinal numbers? Which are cardinal numbers?

a) $\emptyset$

b) $\{\emptyset\}$

c) $\{\{\emptyset\},\{\{\emptyset\}\}\}$

d) $\{\emptyset,\{\{\emptyset\}\}\}$

e) $\{\emptyset,\{\emptyset\},\{\{\emptyset\}\}\}$

f) $\{\emptyset,\{\emptyset\},\{\emptyset,\{\emptyset\}\}\}$

a), b), e), and f) are transitive sets.

a), b), and f) are ordinal numbers.

a), b), and f) are cardinal numbers.

Exercise 3.13 Prove or refute the following assertions:

a) If x is an ordinal number, so is $\mathcal{P}(x)$.

The statement is false. The set

$$x = \{\, \emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\} \,\}$$

is an ordinal number. Furthermore,

$$y = \{\, \emptyset, \{\emptyset, \{\emptyset\}\} \,\}$$

is an element of the power set $\mathcal{P}(x)$. y is not transitive and $\mathcal{P}(x)$ is therefore not an ordinal number.

b) The addition of ordinal numbers is commutative, that is, $\alpha + \beta = \beta + \alpha$.

The statement is false. It is $1 + \omega = \omega$, but $\omega + 1 = s(\omega) \neq \omega$

c) Ordinal multiplication is commutative, that is, $\alpha \cdot \beta = \beta \cdot \alpha$.

The statement is false. It is $2 \times \omega = \omega$, but $\omega \times 2 \neq \omega$

Exercise 3.14

Which of the following statements are equivalent?

a) X is empty
b) X is finite
c) X is countable
d) X is denumerable
e) X is infinite
f) X is uncountable

g) $|X| = \emptyset$ Equivalent to a) (set is empty)
h) $|X| = \aleph_0$ Equivalent to d) (set is denumerable)
i) $|X| < \aleph_0$ Equivalent to b) (set is finite)
j) $|X| > \aleph_0$ Equivalent to f) (set is uncountable)
k) $|X| \leq \aleph_0$ Equivalent to c) (finite or denumerable set)
l) $|X| \geq \aleph_0$ Equivalent to e) (infinite set)
m) $|X| \in \aleph_0$ Equivalent to b) (set is finite)
n) $|X| \subseteq \aleph_0$ Equivalent to c) (set is finite or countable)
o) $|X| \subset \aleph_0$ Equivalent to b) (set is finite)
p) $\aleph_0 \in |X|$ Equivalent to f) (set is uncountable)
q) $\aleph_0 \subseteq |X|$ Equivalent to e) (set is infinite)
r) $\aleph_0 \subset |X|$ Equivalent to f) (set is uncountable)

Exercise 4.1

On page 198, you have learned how to map formulas of Peano arithmetic to natural numbers. This exercise will give you an idea about the size of the numbers we deal with. Find the formulas that correspond to the following Gödel numbers:

a) 279451225562907928022831663325000000000000

$= 2^{11} \cdot 3^2 \cdot 5^{13} \cdot 7^4 \cdot 11^2 \cdot 13^{15} \cdot 17^4$
$\widehat{=} \forall x \, \exists y \, x = y$

b) 92078385275490529327904268091440882663711938445312000 0

$= 2^{23} \cdot 3^{17} \cdot 5^4 \cdot 7^{19} \cdot 11^{15} \cdot 13^4$
$\widehat{=} s(y) = y$

Note: Factorizing such large numbers by hand is almost impossible. Revert to software tools such as Mathematica, Maple, or WolframAlpha for this task.

Exercise 4.2

Use the method from Page 198 to calculate the Gödel number of the formula $\exists x\, s(x) = x$. Repeat the calculation for the formula $\overline{1} + 0 = \overline{1}$.

Note: It is sufficient to write down the Gödel numbers in factorized notation. Entirely written out, both numbers have well over a hundred decimal digits.

$$
\begin{aligned}
&\exists x\, s(x) = x \\
&= 2^{13} \cdot 3^2 \cdot 5^{23} \cdot 7^{17} \cdot 11^2 \cdot 13^{19} \cdot 17^{15} \cdot 19^2 \\
&= 37373116939301289993545612318515250189642832234669 \\
&\quad 52984065263671875000000000000
\end{aligned}
$$

$$
\begin{aligned}
&\overline{1} + 0 = \overline{1} \\
&= s(0) + 0 = s(0) \\
&= 2^{23} \cdot 3^{17} \cdot 5^{21} \cdot 7^{19} \cdot 11^{25} \cdot 13^{21} \cdot 17^{15} \cdot 19^{23} \cdot 23^{17} \cdot 29^{21} \cdot 31^{19} \\
&= 18250787262870278170563135334902730052938878568308 \\
&\quad 29355360272955164268213994223583004153528889022462 \\
&\quad 80795219655567579012132799734179404967706047310581 \\
&\quad 06438331565659275652074134314352491526624023881831 \\
&\quad 96532000000000000000000000
\end{aligned}
$$

Exercise 4.3

In [197], Raymond Smullyan has proposed a clever type of Gödelization. His coding initially assigns one of the following numeric constants to each formula character:

0	′	(	)	f	,	v	∼	⊃	∀	=	≤	♯
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
1	0	2	3	4	5	6	7	8	9	10	11	12

Smullyan then interprets the assigned constants as digits of a number in base 13. In particular, if a formula φ consists of n characters and c_i denotes the i-th character, the following Gödel number is assigned:

$$\ulcorner \varphi \urcorner := \sum_{i=1}^{n} c_i \cdot 13^{n-i}.$$

a) Determine the Gödel number for the formula $\forall \mathsf{v}\, \mathsf{v} = \mathsf{v}$.

Symbol sequence : 9 6 6 10 6

Godel number : $9 * 13^4 + 6 * 13^3 + 6 * 13^2 + 10 * 13^1 + 6$

$= 271381$

b) Smullyan's logic employs the apostrophe as a symbol for the successor function: $0'$ represents the number 1, $0''$ represents the number 2, and so on. Which Gödel number corresponds to the expression representing the natural number n?

The corresponding digit sequence consists of a 1 followed by n zeros.

This results in the Goedel number 13^n. Clever!

Exercise 4.4

Let K be a formal system capable of arithmetically representing relations and functions like we are used to from Peano arithmetic. Further, let A_1 and A_2 be the following propositions:

A_1: If φ represents a relation R semantically, it also represents R syntactically.
A_2: If φ represents a relation R syntactically, it also represents R semantically.

Mark all correct statements:	propositions A_1 is		propositions A_2 is	
If the formal system K is …	true	false	true	false
■ correct and complete, then …	✔	○	✔	○
■ korrekt und negationsvollständig, so ist	✔	○	✔	○
■ widerspruchsfrei und vollständig, so ist	✔	○	✔	○
■ widerspruchsfrei und negationsvollständig, so ist	○	✔	○	✔

Exercise 4.5
✍✍✍

Gödel's paramount paper from 1931 contains a technical part that defines 45 primitive-recursive functions and relations. It starts with two relations and three functions:

$$
\begin{aligned}
P_1(x,y) &:\Leftrightarrow \exists z\,(z \le x \wedge x = y \cdot z)\\
P_2(x) &:\Leftrightarrow \neg\exists (z \le x)\,(z \ne 1 \wedge z \ne x \wedge P_1(x,z)) \wedge x > 1\\
f_3(0,x) &:= 0\\
f_3(n+1,x) &:= \min\{y \le x \mid P_2(y) \wedge P_1(x,y) \wedge y > f_3(n,x)\}\\
f_4(0) &:= 1\\
f_4(n+1) &:= (n+1) \cdot f_4(n)\\
f_5(0) &:= 0\\
f_5(n+1) &:= \min\{y \le f_4(f_5(n)) + 1 \mid P_2(y) \wedge y > f_5(n)\}
\end{aligned}
$$

Find out the substantive meaning of the defined relations and functions.

$P_1(x,y) \Leftrightarrow x$ is divisible by y
$P_2(x) \Leftrightarrow x$ is a prime number
$f_3(n,x) = n$-th prime number contained in x
$f_4(n) = n!$
$f_5(n) = n$-te Primzahl

Exercise 4.6

Consider the following line of reasoning:

a) Every true statement in number theory is a logical consequence of the Peano axioms.

b) Peano arithmetic formalizes the Peano axioms. As a first-order theory, it is subject to Gödel's completeness theorem, which states that first-order theories can prove all logical conclusions.

c) It follows from a) and b) that every true statement of number theory is provable within Peano arithmetic.

The result contradicts Gödel's first incompleteness theorem. Where does the error hide?

If we formulate the Peano axioms as a first-order theory, then a) is an incorrect statement. Due to the weakening of the induction axiom, not every true statement in number theory is a logical consequence of the axioms.

The correct version of a) is:

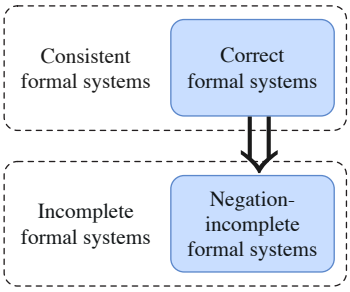
a) Every true statement of number theory is a logical consequence of the Peano axioms *second stage*.

Then a) would be correct, but the completeness theorem no longer applies, since it only applies to first-order theories.

Exercise 4.7

In Section 4.2, you became familiar with the semantic and syntactic variants of Gödel's first incompleteness theorem. Another variant is this one:

"Any correct formal system expressive enough to formalize Peano arithmetic is negation-incomplete."



The formulation is stronger than the semantic variant but weaker than the syntactic variant. Can it still be derived from the semantic variant with little effort?

Yes. If we have the semantic variant in our hands, the variant formulated here can be derived easily. All we need to know is that every correct formal system that is incomplete is always negation-incomplete, too.

Exercise 4.8

On page 212, we claimed that it easy to derive from

$$z = diag(y) \Rightarrow \vdash \forall \mathsf{z}\,(\mathsf{Diag}(\bar{y},\mathsf{z}) \leftrightarrow \mathsf{z} = \bar{z}) \tag{4.1}$$
$$(x,y) \in B \Rightarrow \vdash \mathsf{B}(\bar{x},\bar{y}) \tag{4.2}$$
$$(x,y) \notin B \Rightarrow \vdash \neg\mathsf{B}(\bar{x},\bar{y}) \tag{4.3}$$

the following relationships:

$$x \text{ encodes a proof for the formula } \varphi_y(\bar{y}) \Rightarrow \vdash \psi_{\mathrm{Gdl}}(\bar{x},\bar{y})$$
$$x \text{ does not encode a proof for the formula } \varphi_y(\bar{y}) \Rightarrow \vdash \neg\psi_{\mathrm{Gdl}}(\bar{x},\bar{y})$$

Outline the missing proof.

Proof: Case 1: x encodes a proof of the formula $\varphi_y(\bar{y})$.

Let z be the Gödel number of $\varphi_y(\bar{y})$. Then

$$z = diag(y) \tag{4.4}$$
$$(x,z) \in B \tag{4.5}$$

Then we can construct the proof as follows:

1. $\vdash \forall \mathsf{z}\,(\mathsf{z} = \bar{z} \rightarrow \mathsf{Diag}(\bar{y},\mathsf{z}))$ (from (4.4) and (4.1))
2. $\vdash \mathsf{B}(\bar{x},\bar{z})$ (from (4.5) and (4.2))

   ...
3. $\vdash \mathsf{Diag}(\bar{y},\bar{z})$ (from 1)
4. $\vdash \mathsf{Diag}(\bar{y},\bar{z}) \wedge \mathsf{B}(\bar{x},\bar{z})$ (from 3 and 2)

   ...
5. $\vdash \exists \mathsf{z}\,(\mathsf{Diag}(\bar{y},\mathsf{z}) \wedge \mathsf{B}(\bar{x},\mathsf{z}))$ (from 4)
6. $\vdash \psi(\bar{x},\bar{y})$ (Def)

Case 2: x does not encode a proof of the formula $\varphi_y(\bar{y})$.

Let z be the Gödel number of $\varphi_y(\bar{y})$. Then

$$z = diag(y) \tag{4.6}$$
$$(x,z) \notin B \tag{4.7}$$

Then we can construct the proof as follows:

1. $\vdash \forall \mathsf{z}\,(\mathsf{Diag}(\bar{y},\mathsf{z}) \rightarrow \mathsf{z} = \bar{z})$ (from (4.4) and (4.1))
2. $\vdash \neg\mathsf{B}(\bar{x},\bar{z})$ (from (4.7) and (4.3))

...

3. $\vdash \mathsf{z} = \bar{z} \to \neg \mathsf{B}(\bar{x}, \mathsf{z})$ (from 2)

...

4. $\vdash \mathsf{Diag}(\bar{y}, \mathsf{z}) \to \neg \mathsf{B}(\bar{x}, \mathsf{z})$ (from 1 and 3)

...

5. $\vdash \neg(\mathsf{Diag}(\bar{y}, \mathsf{z}) \wedge \mathsf{B}(\bar{x}, \mathsf{z}))$ (from 4)
6. $\vdash \forall \mathsf{z}\, \neg(\mathsf{Diag}(\bar{y}, \mathsf{z}) \wedge \mathsf{B}(\bar{x}, \mathsf{z}))$ (G, 5)

...

7. $\vdash \neg\exists \mathsf{z}\, (\mathsf{Diag}(\bar{y}, \mathsf{z}) \wedge \mathsf{B}(\bar{x}, \mathsf{z}))$ (from 6)
8. $\vdash \neg\psi(\bar{x}, \bar{y})$ (Def)

Exercise 4.9

Section 2.6 introduced second-order predicate logic and highlighted that the expressive power of PL1 increases with the addition of predicate and function variables. We also mentioned the considerable price to pay. Unlike first-order predicate logic, PL2 is no longer complete if this term relates to the standard semantics; no calculus can derive precisely those formulas of PL2 that are universally valid.

a) Construct a PL2 formula φ that is true under exactly those interpretations whose individual domains are isomorphic to the natural numbers:

$$(U,I) \models \varphi \;\Leftrightarrow\; U \cong \mathbb{N} \tag{4.8}$$

PA is the formula that results from the conjunction of the PA axioms and the following substitutions:

- The symbol 0 was replaced by the constant n.
- The successor operation s was replaced by the function variable $\mathfrak{f}$.
- The operators '$+$' and '$\times$' have been replaced by the function variables $\mathfrak{f}$ and $\mathfrak{g}$.

The substitution makes PA a native PL2 formula and the formula

$$\varphi \;:=\; \exists \mathfrak{f}\, \exists \mathfrak{g}\, \exists \mathfrak{f}\, \exists \mathsf{n}\; \mathsf{PA}$$

has the property we are looking for:

$$(U,I) \models \varphi \;\Leftrightarrow\; U \cong \mathbb{N} \tag{4.9}$$

b) Show that the incompleteness of PL2 is a direct consequence of Gödel's first incompleteness theorem.

The Peano axioms, formulated in PL2, are categorical, i.e. they are true if and only if the set of individuals is isomorphic to the natural numbers. This enables us to

> reconstruct Peano arithmetic within PL2 in such a way that every true statement of number theory corresponds to a universally valid PL2 formula. If PL2 were complete, then all generally valid PL2 formulas could be proved and thus every true statement of number theory. However, this is precisely what the first Gödel incompleteness theorem precludes.

c) Explain why no PL1 formula satisfies (4.9).

If there were such a formula, which we will call $\varphi_{\mathbb{N}}$, we could conjunctively combine it with the Peano axioms (formulated in PL1) and thus obtain a categorical characterization of the natural numbers within PL1 (in effect, we would have eliminated all nonstandard models by adding this formula). The same argument as above would then lead to the incompleteness of PL1, contradicting Gödel's completeness theorem.

Exercise 4.10

Section 4.2.8 defined what it means to *name* a natural number within Peano arithmetic.

Which numbers are named by the following formulas?

a) $x + x = (s(s(0)))$

1

b) $x + x = s(s(s(0)))$

2

c) $x \times x = s(s(0))$

none ($\sqrt{2}$ is not a natural number)

d) $x \times x = s(s(s(0)))$

2

Which of the following statements about the formulas of Peano arithmetic are correct?

e) *"Each formula names a natural number."*

The statement is false. The formula from part c) is a counterexample.

f) *"Each natural number is named by a formula."*

The statement is correct. The number n is z. B. named by the formula $x = \overline{n}$.

g) *"There are an infinite number of ways to name a natural number."*

The statement is correct. Each of the (infinitely many) formulas

$$\underbrace{x + \ldots + x}_{m\text{-mal}} = \underbrace{s(\ldots s(s(s(s(0)))) \ldots)}_{n \cdot m\text{-mal}}$$

names the number n.

Exercise 5.1

The instruction sets of two Turing machines, M_1 and M_2, are given as follows:

$$I_1 := \{(q_1,S_0,S_1,\mathrm{R},q_2),(q_2,S_0,S_0,\mathrm{R},q_3),(q_3,S_0,S_2,\mathrm{R},q_1)\}$$
$$I_2 := \{(q_1,S_0,S_1,\mathrm{R},q_2),(q_2,S_0,S_0,\mathrm{R},q_3),(q_3,S_0,S_2,\mathrm{R},q_1),(q_4,S_0,S_0,\mathrm{R},q_1)\}$$

a) Create the standard descriptions of M_1 and M_2.

M_1 : 731332531173113353111731113322531
;*DADDCRDAA*;*DAADDRDAAA*;*DAAADDCCRDA*

M_2 : 7313325311731133531117311133225317311111133531
;*DADDCRDAA*;*DAADDRDAAA*;*DAAADDCCRDA*;*DAAAADDRDA*

b) Analyze the behavior of the machines. How do both differ?

Obviously, M_2 has a different set of instructions and thus a different Gödel number than M_1. However, M_1 and M_2 are functionally identical, since the state q_4 is unreachable and the last instruction can therefore never execute. Bottom line: Machines that are functionally identical can still have different Gödel numbers.

Exercise 5.2

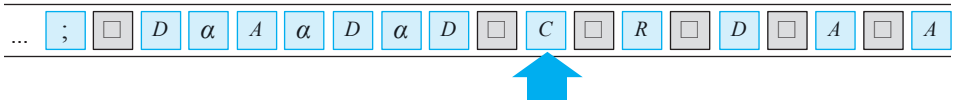
Figure 5.12 contains the instruction table of Turing's historical *universal machine*. Among others, Turing defined the following submachine:

$$\mathfrak{con}(\mathfrak{C},\alpha)\begin{cases} \text{Not } A & R,R & \mathfrak{con}(\mathfrak{C},\alpha) \\ A & L,P\alpha,R & \mathfrak{con}_1(\mathfrak{C},\alpha) \end{cases}$$

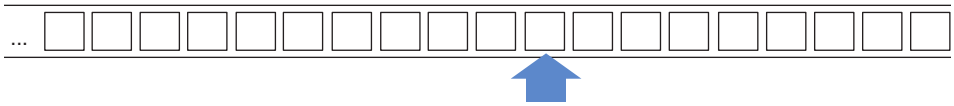
$$\mathfrak{con}_1(\mathfrak{C},\alpha)\begin{cases} A & R,P\alpha,R & \mathfrak{con}_1(\mathfrak{C},\alpha) \\ D & R,P\alpha,R & \mathfrak{con}_2(\mathfrak{C},\alpha) \\ \text{None} & PD,R,P\alpha,R,R,R & \mathfrak{C} \end{cases}$$

$$\mathfrak{con}_2(\mathfrak{C},\alpha)\begin{cases} C & R,P\alpha,R & \mathfrak{con}_2(\mathfrak{C},\alpha) \\ \text{Not } C & R,R & \mathfrak{C} \end{cases}$$

When started in the configuration



the machine terminates after 10 computation steps with the read-write head located at the following tape position:



a) Add the tape content in the illustration above.

b) Which subtask might the machine fulfill in Turing's overall design?

Turing employs them to mark the first two components of an instruction. First, the semicolon is sought with another machine. Then, the machine con is used to mark the components q_i, S_j and S_k of an instruction by writing the symbol α to the left of each character. The fact that there is a free cell next to each tape cell is a basic concept in the way Turing's computing machines work. As this example shows, he uses them to attach temporary marks or to store intermediate results.

Exercise 5.3

A universal Turing machine takes another Turing machine as input in encoded form. This exercise focuses on the encoding proposed in Turing's 1936 paper.

Rate the following statements:	true	false
■ Every natural number is the Gödel number of a Turing machine.	◯	✔
■ The encoding is injective.	✔	◯
■ The encoding is surjective.	◯	✔
■ The encoding is bijektive.	◯	✔

Hints:

1. The first assertion is false, since every Gödel number begins with the digit 7.
2. The second assertion is correct. Every Gödelization is injective.
3. The third assertion is a different formulation of the first and is therefore false.
4. The fourth claim is false, since every bijective function must also be surjective.

Exercise 5.4

A Turing machine or a register machine may operate in two distinct ways: as a transducer or an acceptor. For a given function $f : \mathbb{N} \to \mathbb{N}$, this means the following: As a transducer, the machine takes x as input and produces y as output. As an acceptor, it receives the tuple (x, y) and accepts the input precisely if x and y satisfy the relationship $f(x) = y$.

This exercise aims at establishing a connection between the two concepts.

a) Can a transducer simulate an acceptor?

Yes. How this is achieved can be seen directly from the definition of decidability. We just have to construct a calculating machine that writes a 0 or a 1 as the result. The input word is accepted if the function value is equal to 1. It is rejected if the result value is equal to 0.

b) Can an acceptor simulate a transducer?

This is also possible. Suppose we want to calculate a function $f(x)$, but we only have an acceptor available that accepts those inputs (x, y) for which $y = f(x)$. To calculate the function value $f(x)$, we convert the acceptor into an enumerator, as described in Figure 5.24.

If $f(x) \neq 0$, the tuple (x, y) will be enumerated at some point. In this case, we write y on the tape and halt. If $f(x) = 0$, the machine continues forever. Thus, we have constructed a calculating machine for f.

Exercise 5.5

Section 5.1.2 has introduced a register machine that James P. Jones and Yuri Matiyasevich invented in 1991 [110]. The machine, which works as a transducer, utilizes register R_1 to receive the input x and store the output $f(x)$. The execution log in Figure 5.17 provided insight into how the machine acts for the input $R_1 = 2$. It stopped after 23 steps, leaving the result 1 in R_1.

a) Complete the list below by simulating the machine for additional inputs:

$f(0) =$ 0 , $f(1) =$ 1 , $f(2) =$ 1 , $f(3) =$ 2 , $f(4) =$ 3 , $f(5) =$ 5

b) Which famous number sequence does the machine compute?

The machine calculates the Fibonacci numbers. They are defined recursively:

$$\begin{aligned} F_0 &:= 0 \\ F_1 &:= 1 \\ F_{n+2} &:= F_{n+1} + F_n \end{aligned}$$

Exercise 5.6

All discussed register machines offered three branch instructions:

- L_i : goto L_n
- L_i : if $R_j = 0$ goto L_n
- L_i : if $R_j \neq 0$ goto L_n

a) Do these instructions suffice to implement the following extended branch instruction?

$$L_i : \text{if } R_j = 0 \text{ goto } L_n \text{ else goto } L_m$$

Yes.

L_i : if $R_j = 0$ goto L_n
L_{i+1} : goto L_m

b) Does the expressiveness decrease if we restrict ourselves to the instruction

$$L_i : \text{if } R_j = 0 \text{ goto } L_n ?$$

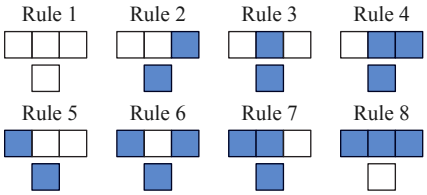
No, the other two can be simulated.

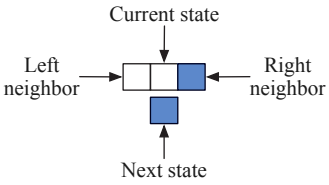
Simulation of L_i : goto L_n
L_i : if $R_j = 0$ goto L_n where R_j is an unused register.

Simulation of L_i : if $R_j \neq 0$ goto L_n
L_i : if $R_j = 0$ goto L_{i+2}
L_{i+1} : goto L_n

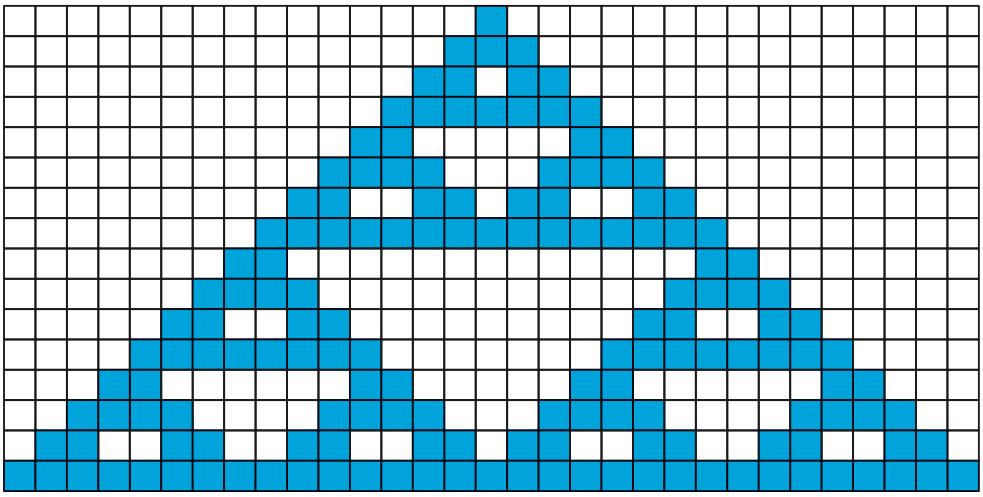
Exercise 5.7

This exercise is about the cellular automaton defined by the following rules:





Which familiar structure is generated by this automaton? To answer the question, complement the following diagram.

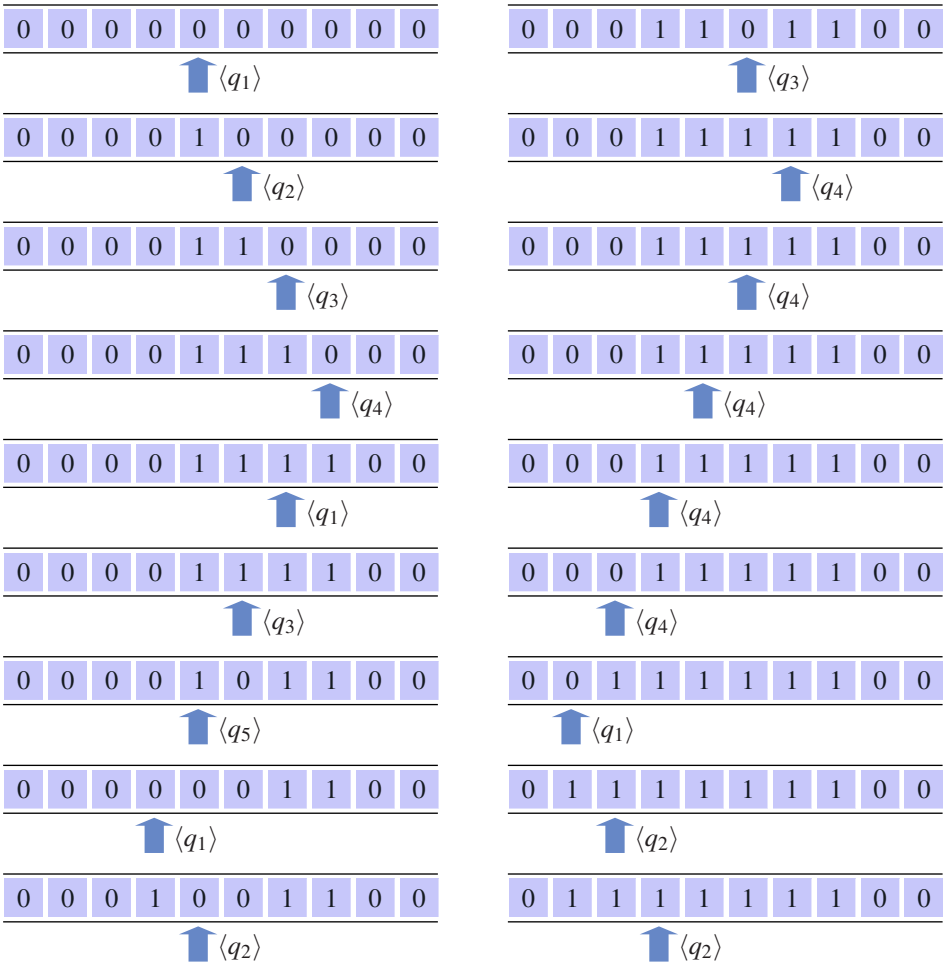


Exercise 5.8

Consider the following Turing machine:

$$Q = \{q_1, q_2, q_3, q_4, q_5, q_6\}$$
$$S = \{0, 1\}$$
$$I = \{(q_1,0,1,\mathrm{R},q_2),(q_2,0,1,\mathrm{R},q_3),(q_3,0,1,\mathrm{R},q_4),(q_4,0,1,\mathrm{L},q_1),(q_5,0,1,\mathrm{R},q_6),$$
$$(q_1,1,1,\mathrm{L},q_3),(q_2,1,1,\mathrm{R},q_2),(q_3,1,0,\mathrm{L},q_5),(q_4,1,1,\mathrm{L},q_4),(q_5,1,0,\mathrm{L},q_1)\}$$

a) Complete the simulation run printed below.



b) Find out which well-known Turing machine we are looking at here. Will the machine terminate?

Exercise 5.9

In Definition 5.7, we agreed that a set N is *enumerable* if and only if a surjective and computable function $f : \mathbb{N} \to N$ exists. What are the consequences of replacing the requirement of surjectivity with the requirement of bijectivity?

This has no consequences from a theoretical point of view, since we can reprogram each enumerator so that it outputs each element only once. To do this, it could log all previously generated output during the calculation and suppress the output if an element has already been output previously.

The fact that most books only require surjectivity has a very practical reason. Many enumeration algorithms are then much easier to formulate because we no longer have to ensure that each element is output exactly once.

Exercise 5.10

Section 5.3.2 presented Rice's theorem. In a sweeping blow, it dashed any hope of algorithmically deciding any non-trivial property of Turing machines. However, is this really so? For example, consider the property of a machine to have precisely five states. For any machine, we can easily decide on this property by briefly inspecting the instruction table. However, this should be impossible according to Rice's theorem, shouldn't it?

It is important to read Rice's result carefully: it destroys the hope of algorithmically deciding any non-trivial *functional* property of Turing machines. A functional property is a property of the function f_T and not a property of T itself. f_T is the function calculated by T.

The property of having 5 states is not a functional property. It is easy to construct two Turing machines, T_1 and T_2, having a different number of states and still satisfying the relation $f_{T_1} = f_{T_2}$.

Exercise 5.11

In this exercise, we take a closer look at the formula $\mathrm{Inst}(q_i, S_j, S_k, \mathrm{L}, q_l)$ from Section 5.4.1. In his 1936 paper, Turing employed it to describe the leftward movement of a tape-restricted Turing machine:

$$\begin{aligned}
&\mathrm{Inst}(q_i, S_j, S_k, \mathrm{L}, q_l) := \\
&\quad \forall \mathsf{t}\, \forall \mathsf{y}\, \forall \mathsf{t}'\, \forall \mathsf{y}'\, ((\mathsf{R}_{\mathsf{S}_j}(\mathsf{t},\mathsf{y}) \wedge \mathsf{I}(\mathsf{t},\mathsf{y}) \wedge \mathsf{K}_{\mathsf{q}_i}(\mathsf{t}) \wedge \mathsf{F}(\mathsf{t},\mathsf{t}') \wedge \mathsf{F}(\mathsf{y}',\mathsf{y})) \\
&\qquad \rightarrow (\mathsf{I}(\mathsf{t}',\mathsf{y}') \wedge \mathsf{R}_{\mathsf{S}_k}(\mathsf{t}',\mathsf{y}) \wedge \mathsf{K}_{\mathsf{q}_l}(\mathsf{t}') \\
&\qquad\quad \wedge \forall \mathsf{z}\, (\mathsf{F}(\mathsf{y}',\mathsf{z}) \vee \bigwedge_{i=0}^{M} (\mathsf{R}_{\mathsf{S}_i}(\mathsf{t},\mathsf{z}) \rightarrow \mathsf{R}_{\mathsf{S}_i}(\mathsf{t}',\mathsf{z})))))
\end{aligned}$$

On page 261, we have defined how such a machine behaves when the read-write head is on the far left:

> *"As the read-write head of a tape-restricted Turing machine can never move beyond the end of the tape, the machine ignores any request to cross the boundary and keeps the read-write head in its current position instead."*

Apparently, the verbal description does not manifest itself anywhere in the formula $\varphi_I(\mathsf{x},\mathsf{y})$.

a) Is Turing's proof perhaps incomplete?

No. It is not necessary to treat the behavior for the case described. In the section on the Turing machine with one-sided restricted tape, it was shown that a Turing machine can be converted so that it never moves the write-read head beyond a special marker symbol to the left. This means that we can assume that the case described will never occur. That is why Turing did not need to explicitly code it into his formula.

b) How difficult would it be to incorporate the verbal description into the formula $\varphi_I(\mathsf{x},\mathsf{y})$?

The following formula serves this purpose:

$$\begin{aligned}
&\mathrm{Inst}(q_i, S_j, S_k, \mathrm{L}, q_l) := \\
&\quad \forall \mathsf{t}\, \forall \mathsf{t}'\, ((\mathsf{R}_{\mathsf{S}_j}(\mathsf{t},0) \wedge \mathsf{I}(\mathsf{t},0) \wedge \mathsf{K}_{\mathsf{q}_i}(\mathsf{t}) \wedge \mathsf{F}(\mathsf{t},\mathsf{t}') \\
&\qquad \rightarrow (\mathsf{I}(\mathsf{t}',0) \wedge \mathsf{R}_{\mathsf{S}_k}(\mathsf{t}',0) \wedge \mathsf{K}_{\mathsf{q}_l}(\mathsf{t}') \\
&\qquad\quad \wedge \forall \mathsf{z}\, (\mathsf{F}(0,\mathsf{z}) \vee \bigwedge_{i=0}^{M} (\mathsf{R}_{\mathsf{S}_i}(\mathsf{t},\mathsf{z}) \rightarrow \mathsf{R}_{\mathsf{S}_i}(\mathsf{t}',\mathsf{z}))))) \wedge \\
&\quad \forall \mathsf{t}\, \forall \mathsf{y}\, \forall \mathsf{t}'\, \forall \mathsf{y}'\, ((\mathsf{R}_{\mathsf{S}_j}(\mathsf{t},\mathsf{y}) \wedge \mathsf{I}(\mathsf{t},\mathsf{y}) \wedge \mathsf{K}_{\mathsf{q}_i}(\mathsf{t}) \wedge \mathsf{F}(\mathsf{t},\mathsf{t}') \wedge \mathsf{F}(\mathsf{y}',\mathsf{y})) \\
&\qquad \rightarrow (\mathsf{I}(\mathsf{t}',\mathsf{y}') \wedge \mathsf{R}_{\mathsf{S}_k}(\mathsf{t}',\mathsf{y}) \wedge \mathsf{K}_{\mathsf{q}_l}(\mathsf{t}') \\
&\qquad\quad \wedge \forall \mathsf{z}\, (\mathsf{F}(\mathsf{y}',\mathsf{z}) \vee \bigwedge_{i=0}^{M} (\mathsf{R}_{\mathsf{S}_i}(\mathsf{t},\mathsf{z}) \rightarrow \mathsf{R}_{\mathsf{S}_i}(\mathsf{t}',\mathsf{z})))))
\end{aligned}$$

One problem remains. The 0 is not available to us and must be eliminated. We achieve this by replacing all occurrences of 0 with a variable n and demanding that it represents the number that has no predecessor. We then obtain the following result:

$$
\begin{aligned}
&\mathrm{Inst}(q_i, S_j, S_k, \mathrm{L}, q_l) :=\\
&\exists \mathsf{n}\ (\forall \mathsf{z}\ \neg \mathsf{F}(\mathsf{z},\mathsf{n}) \wedge\\
&\quad \forall \mathsf{t}\ \forall \mathsf{t}'\ ((\mathsf{R}_{\mathsf{S}_j}(\mathsf{t},\mathsf{n}) \wedge \mathsf{I}(\mathsf{t},\mathsf{n}) \wedge \mathsf{K}_{\mathsf{q}_i}(\mathsf{t}) \wedge \mathsf{F}(\mathsf{t},\mathsf{t}')\\
&\qquad \rightarrow (\mathsf{I}(\mathsf{t}',\mathsf{n}) \wedge \mathsf{R}_{\mathsf{S}_k}(\mathsf{t}',\mathsf{n}) \wedge \mathsf{K}_{\mathsf{q}_l}(\mathsf{t}')\\
&\qquad\quad \wedge \forall \mathsf{z}\ (\mathsf{F}(\mathsf{n},\mathsf{z}) \vee \bigwedge_{i=0}^{M} (\mathsf{R}_{\mathsf{S}_i}(\mathsf{t},\mathsf{z}) \rightarrow \mathsf{R}_{\mathsf{S}_i}(\mathsf{t}',\mathsf{z}))))) \wedge\\
&\quad \forall \mathsf{t}\ \forall \mathsf{y}\ \forall \mathsf{t}'\ \forall \mathsf{y}'\ ((\mathsf{R}_{\mathsf{S}_j}(\mathsf{t},\mathsf{y}) \wedge \mathsf{I}(\mathsf{t},\mathsf{y}) \wedge \mathsf{K}_{\mathsf{q}_i}(\mathsf{t}) \wedge \mathsf{F}(\mathsf{t},\mathsf{t}') \wedge \mathsf{F}(\mathsf{y}',\mathsf{y}))\\
&\qquad \rightarrow (\mathsf{I}(\mathsf{t}',\mathsf{y}') \wedge \mathsf{R}_{\mathsf{S}_k}(\mathsf{t}',\mathsf{y}) \wedge \mathsf{K}_{\mathsf{q}_l}(\mathsf{t}')\\
&\qquad\quad \wedge \forall \mathsf{z}\ (\mathsf{F}(\mathsf{y}',\mathsf{z}) \vee \bigwedge_{i=0}^{M} (\mathsf{R}_{\mathsf{S}_i}(\mathsf{t},\mathsf{z}) \rightarrow \mathsf{R}_{\mathsf{S}_i}(\mathsf{t}',\mathsf{z})))))
\end{aligned}
$$

Exercise 5.12

Section 5.4.2 dealt with the arithmetization of Turing machines. In this context, we introduced the PA formula $\varphi_I(x,y)$, describing the transition from one configuration x to another configuration y. For the left movement, it reads as follows:

$$\begin{aligned}
\varphi_I(x,y) \;:=\; & \exists h_1\, \exists h_2\, \exists n_1\, \exists n_2 (\\
& L(x) = n_1 \wedge K(x) = \bar{i} \wedge I(x) = h_1 \wedge R_{h_1}(x) = \bar{j} \wedge \\
& L(y) = n_2 \wedge K(y) = \bar{l} \wedge I(y) = h_2 \wedge \\
& (h_1 \neq 0 \rightarrow (\\
& \quad n_1 = n_2 \wedge h_1 = h_2 + \bar{1} \wedge R_{h_1}(y) = \bar{k} \wedge \\
& \quad \forall (h < n_1)\, (h \neq h_1 \rightarrow \exists s\, (R_h(x) = s \wedge R_h(y) = s)))) \wedge \\
& (h_1 = 0 \rightarrow (\\
& \quad n_1 + \bar{1} = n_2 \wedge R_0(y) = 0 \wedge h_1 = h_2 \wedge R_1(y) = \bar{k} \wedge \\
& \quad \forall (h < n_1)\, (h \neq 0 \rightarrow \exists s\, (R_h(x) = s \wedge R_{h+1}(y) = s)))))
\end{aligned}$$

We could have simplified the proof by restricting ourselves to tape-restricted Turing machines. How would the formula read in that case?

$$\begin{aligned}
\varphi_I(x,y) \;:=\; & \exists h_1\, \exists h_2\, \exists n\, (\\
& L(x) = n \wedge K(x) = \bar{i} \wedge I(x) = h_1 \wedge R_{h_1}(x) = \bar{j} \wedge \\
& L(y) = n \wedge K(y) = \bar{l} \wedge I(y) = h_2 \wedge R_{h_1}(y) = \bar{k} \wedge \\
& h_1 \neq 0 \rightarrow h_1 = h_2 + \bar{1} \wedge \\
& h_1 = 0 \rightarrow h_1 = h_2 \wedge \\
& \forall (h < n)\, (h \neq h_1 \rightarrow \exists s\, (R_h(x) = s \wedge R_h(y) = s))))
\end{aligned}$$

Exercise 5.13

By considering the example

$$(x+1)^3+(y+1)^3=(z+1)^3,$$

we argued in Section 5.4.3 that it makes a difference whether we seek the solutions of a Diophantine equation in the integers or the natural numbers.

a) Demonstrate that the equation has an infinite number of solutions in the integers.

For example, let $y=-1$ and $x=z$.

b) Why is the equation unsolvable in the natural numbers?

The unsolvability is a consequence of Fermat's theorem.

Exercise 5.14

Section 5.4.3 worked out how to combine Diophantine equations conjunctively or disjunctively.

a) Which relations do the following two equations represent?

$$a+x+1-b=0 \lor b+y+1-a=0$$

$$\begin{aligned} a+x+1-b=0 \lor b+y+1-a=0 \\ \Leftrightarrow\ a<b \lor b<a \\ \Leftrightarrow\ a\neq b \end{aligned}$$

$$a+x-b=0 \land b+y-a=0$$

$$\begin{aligned} a+x-b=0 \land b+y-a=0 \\ \Leftrightarrow\ a\leq b \land b\leq a \\ \Leftrightarrow\ a=b \end{aligned}$$

b) Translate the expressions into ordinary Diophantine equations.

$$\begin{aligned} a+x+1-b=0 \lor b+y+1-a=0 \\ \Leftrightarrow\ (a+x+1-b)\cdot(b+y+1-a)=0 \\ \Leftrightarrow\ ab+ay+a-a^2+bx+xy+x-ax+b+y+1-a-b^2-by-b+ab=0 \\ \Leftrightarrow\ -a^2+2ab-b^2+ay-ax+bx-by+xy+x+y+1=0 \end{aligned}$$

$$\begin{aligned} a+x-b=0 \land b+y-a=0 \\ \Leftrightarrow\ (a+x-b)^2+(b+y-a)^2=0 \\ \Leftrightarrow\ ((a-b)+x)^2+(y-(a-b))^2=0 \\ \Leftrightarrow\ (a-b)^2+2x(a-b)+x^2+y^2-2y(a-b)+(a-b)^2=0 \\ \Leftrightarrow\ 2(a-b)^2+2x(a-b)+x^2+y^2-2y(a-b)=0 \\ \Leftrightarrow\ 2a^2-4ab+2b^2+2ax-2bx+x^2+y^2-2ay-2by=0 \end{aligned}$$

Exercise 5.15 Let R and S be two diophantically representable relations.

a) Does the relation $R \cup S$ have a Diophantine representation?

Yes. If R is represented by $r = 0$ and S by $s = 0$,

then $R \cup S$ is represented by $r = 0 \vee q = 0$.

b) Does the relation $R \cap S$ have a Diophantine representation?

Yes. If R is represented by $r = 0$ and S by $s = 0$,

then $R \cap S$ is represented by $r = 0 \wedge q = 0$.

Exercise 5.16

Figure 5.41 features a Diophantine equation with 26 unknowns, having a solution in the positive natural numbers if and only if $k+2$ is a prime number.

a) Can this equation be utilized to enumerate all prime numbers?

Yes. Enter all combinations one after the other and solve the equation. If the combination is a solution, then output the number $k+2$.

b) Does the set of all prime twins also have a Diophantine representation?

Yes:

$$\text{prime}(x) \wedge \text{prime}(y) \wedge x+2-y=0$$

Exercise 5.17 ✍✍✍

The register machine program below originates from the repeatedly cited work by Jones and Matiyasevich from 1984 [109]:

L_0	$R_2 \leftarrow R_2 + 1$		$R_2 \leftarrow R_2 - 1$	L_{11}	if $R_2 < R_1$ goto L_{10}
L_1	$R_2 \leftarrow R_2 + 1$	L_6	if $0 < R_2$ goto L_5	L_{12}	$R_1 \leftarrow R_1 - 1$
L_2	if $R_3 = 0$ goto L_5	L_7	$R_2 \leftarrow R_2 + 1$		$R_2 \leftarrow R_2 - 1$
L_3	$R_3 \leftarrow R_3 - 1$		$R_4 \leftarrow R_4 - 1$		$R_3 \leftarrow R_3 - 1$
L_4	goto L_2	L_8	if $0 < R_4$ goto L_7	L_{13}	if $0 < R_1$ goto L_{12}
L_5	$R_3 \leftarrow R_3 + 1$	L_9	if $R_3 < R_1$ goto L_5	L_{14}	stop
	$R_4 \leftarrow R_4 + 1$	L_{10}	if $R_1 < R_3$ goto L_1		

a) With $R_i < R_j$, the machine utilizes an operation unsupported by the original register machine model. Show that the native language elements are sufficiently expressive to simulate the operation.

The query $R_i < R_j$ can be simulated by first copying R_i and R_j into free registers R'_i and R'_j. Then both are decremented by 1 step by step. If R'_i reaches the value 0 first, then $R_i < R_j$.

b) Manually execute the program with input $R_1 = 2$ and create an execution log similar to the one in Figure 5.17. Note that the program starts in line L_0 and not, as before, in line L_1.

	Line	Instruction	R_1	R_2	R_3	R_4
1	L_0	$R_2 \leftarrow R_2 + 1$	2	0	0	0
2	L_1	$R_2 \leftarrow R_2 + 1$	2	1	0	0
3	L_2	if $R_3 = 0$ goto L_5	2	2	0	0
4	L_5	$R_3 \leftarrow R_3 + 1$ $R_4 \leftarrow R_4 + 1$ $R_2 \leftarrow R_2 - 1$	2	2	0	0
5	L_6	if $0 < R_2$ goto L_5	2	1	1	1
6	L_5	$R_3 \leftarrow R_3 + 1$ $R_4 \leftarrow R_4 + 1$ $R_2 \leftarrow R_2 - 1$	2	1	1	1
7	L_6	if $0 < R_2$ goto L_5	2	0	2	2
8	L_7	$R_2 \leftarrow R_2 + 1$ $R_4 \leftarrow R_4 - 1$	2	0	2	2
9	L_8	if $0 < R_4$ goto L_7	2	1	2	1
10	L_7	$R_2 \leftarrow R_2 + 1$ $R_4 \leftarrow R_4 - 1$	2	1	2	1
11	L_8	if $0 < R_4$ goto L_7	2	2	2	0
12	L_9	if $R_3 < R_1$ goto L_5	2	2	2	0
13	L_{10}	if $R_1 < R_3$ goto L_1	2	2	2	0
14	L_{11}	if $R_2 < R_1$ goto L_{10}	2	2	2	0
15	L_{12}	$R_1 \leftarrow R_1 - 1$ $R_2 \leftarrow R_2 - 1$ $R_3 \leftarrow R_3 - 1$	2	2	2	0
16	L_{13}	if $0 < R_1$ goto L_{12}	1	1	1	0
17	L_{12}	$R_1 \leftarrow R_1 - 1$ $R_2 \leftarrow R_2 - 1$ $R_3 \leftarrow R_3 - 1$	1	1	1	0
18	L_{13}	if $0 < R_1$ goto L_{12}	0	0	0	0
19	L_{14}	stop	0	0	0	0

c) Set up the data and control flow matrix for this computation sequence. You already know what such a matrix looks like from Figure 5.43.

18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	t
0	0	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	R_1
0	0	1	1	2	2	2	2	2	1	1	0	0	1	1	2	2	1	0	R_2
0	0	1	1	2	2	2	2	2	2	2	2	2	1	1	0	0	0	0	R_3
0	0	0	0	0	0	0	0	0	1	1	2	2	1	1	0	0	0	0	R_4

18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	t
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	L_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	L_1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	L_2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	L_3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	L_4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	L_5
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	L_6
0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	L_7
0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	L_8
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	L_9
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	L_{10}
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	L_{11}
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	L_{12}
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	L_{13}
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	L_{14}

d) By starting the register machine with different input values, you will notice that it halts in some cases and keeps calculating forever in others. Try to establish a connection between the input value and the termination property.

The program terminates exactly when the input in R_1 is a prime number.

Exercise 5.18

Section 5.4.3.2 demonstrated how to encode register machines diophantically. Among others, the constructed equation contained the subexpression

$$L_1 \preccurlyeq I \ \wedge \ \ldots \ \wedge \ L_l \preccurlyeq I \tag{5.10}$$

to ensure that $L_1, \ldots, L_l$ only consists of the digits 0 and 1. In addition, we utilized the subexpression

$$I = \sum_{i=1}^{l} L_i \tag{5.11}$$

to guarantee that there is at most one '1' in each column of the control flow matrix. At first glance, we can do without (5.10), as it seems to follow from (5.11). Prove or disprove this assertion.

The assumption is false!

Let $l := 2$, $Q := 4$, $L_1 := 02_Q$ and $L_2 := 03_Q$.

For the chosen values of L_1 and L_2, (5.10) is false. On the other hand, $L_1 + L_2 = 11_Q$ and therefore (5.11) is true. So we actually have to include both partial expressions in the Diophantine equation.

Exercise 6.1

Let r_i denote the i-th digit of a random binary sequence $(i \geq 0)$. In which of the following cases is the sequence $s_0, s_1, s_2, \ldots$ also random?

a) $s_i = \begin{cases} 1 & \text{if } i < 10 \\ r_i & \text{otherwise} \end{cases}$

Random

b) $s_i = \begin{cases} r_{r_i} & \text{if } i \text{ is even} \\ r_i & \text{otherwise} \end{cases}$

Random

c) $s_i = \begin{cases} r_i & \text{if } i < 10 \\ 1 & \text{otherwise} \end{cases}$

Not random

d) $s_i = \begin{cases} r_i & \text{if } i \text{ is a prime number} \\ 1 & \text{otherwise} \end{cases}$

Not random

e) $s_i = \begin{cases} r_i & \text{if } i \text{ is even} \\ 1 & \text{otherwise} \end{cases}$

Not random

f) $s_i = \begin{cases} r_i & \text{if } i \text{ is a square number} \\ 1 & \text{otherwise} \end{cases}$

Not random

Exercise 6.2

Name, if possible, an example of a *finitely* long binary sequence that is

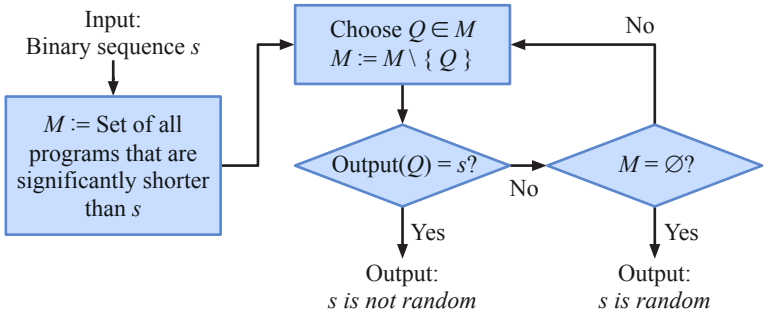
- computable and compressible: 1111111111...1
- computable but uncompressible: any random sequence
- uncomputable but compressible: impossible
- uncomputable and uncompressible: impossible

Name, if possible, an example of an *infinitely* long binary sequence that is

- computable and compressible: π
- computable but uncompressible: impossible
- uncomputable but compressible: Halting sequence H
- uncomputable and uncompressible: Chaitin's constant Ω

Exercise 6.3

This exercise puts the content of Theorem 6.2 to the test. The theorem states that no systematic procedure can always correctly decide whether a given binary sequence s is random. Nevertheless, the following program seems to do the trick:



The design of the algorithm is motivated by the following observations: Only a finite number of programs are shorter than the given binary sequence s. If s were not random, it would be the output of one of them. Therefore, it suffices to iterate over the (finitely many) programs significantly shorter than s and compare their output against s. If both match, s is not a random sequence. However, if s is not output by these programs, it must be random. Needless to say, the result contradicts Theorem 6.2, but where exactly is the reasoning flawed?

The apparent contradiction can be quickly resolved. The problem is that we can only determine the output of the program Q by executing Q. Since not every program Q terminates, our program will inevitably end up in an endless loop.

If we could decide in advance which programs terminate and which do not, algorithmic complexity could actually be calculated in the outlined way. We have thus uncovered an important relationship. It follows directly from the undecidability of $\kappa(s)$ that we cannot decide the termination of Turing machines algorithmically. The consideration shows how strong the content of Theorem 6.2 really is. It yields Turing's famous theorem on the undecidability of the halting problem as a corollary.

Exercise 6.4

Chapter 1 discussed the twin prime conjecture:

> *"There are infinitely many numbers n with the property that n and $n+2$ are prime numbers."*

This exercise assumes the halting probability Ω_n is known for any given n. On page 336, we explained how to exploit this knowledge to prove or disprove statements such as Goldbach's conjecture. Could we use the same method to decide the conjecture about the existence of infinitely many prime twins?

For Goldbach's conjecture, the method described worked like this:

- A program is written that searches for a counterexample. The program terminates as soon as a counterexample is found. If there are no counterexamples, the program continues forever.
- If we knew the holding probabilities Ω_n, we could decide whether the program holds (Goldbach's conjecture would be false) or does not hold (Goldbach's conjecture would be true).

The method cannot be applied one-to-one to the twin conjecture, because we cannot write a program that searches for a counterexample in the same way. This is due to the fact that we have an existential proposition here (there are infinitely many...). A found pair of prime numbers is of no use in making a statement, nor is a pair of prime numbers that is not a pair of twins.

Exercise 6.5

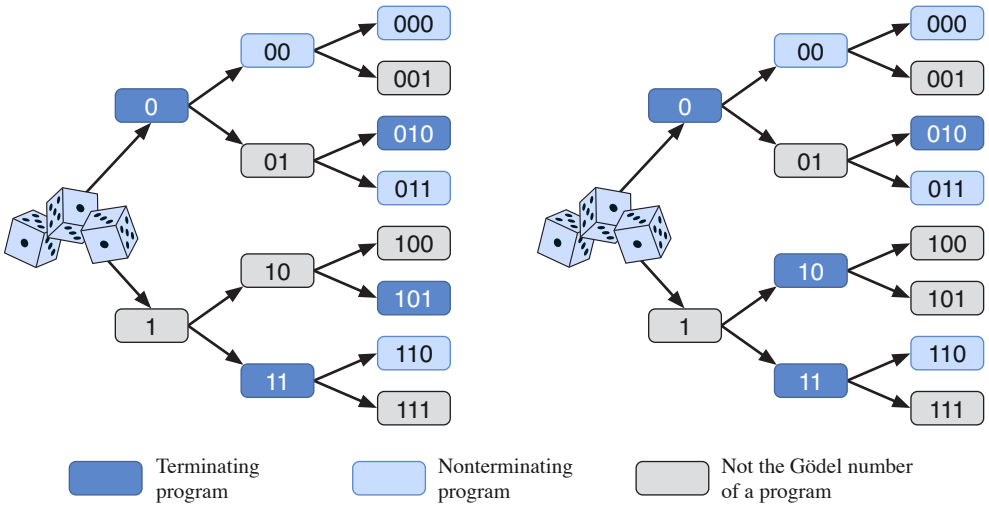
Section 6.2 revealed that the bit sequence of Chaitin's constants is random. Therefore, whether 0 or 1 occurs at a specific bit position of Ω is independent of the bits at other positions. Can we nevertheless make a statement about *how many* ones and zeros an initial piece $\Omega[1 \ldots n]$ contains for larger values of n?

Yes. It follows directly from proposition 6.4 that the bit sequence of the Chaitin constant must contain as many zeros as ones.

Exercise 6.6

In the investigations on algorithmic complexity, we have assumed that the applied Gödelizations must be prefix-free; that is, the Gödel number of a program never starts with the Gödel number of another program. This exercise will elucidate why we need this assumption.

Consider the following two decision trees:



- Explain why the respective Gödelizations are not prefix-free.

  For example, both 00 and 000 are the Gödel number of a program. However, 000 starts with 00.

- For each decision tree, calculate the halting probability Ω_3.
  - Left:
    7 of the sequences 000, 001, 010, 011, 100, 101, 110, 111 start with the Gödel number of a terminating program (000, 001, 010, 011, 101, 110, 111).
    Result: $\Omega_3 = 7/8$
  - Right:
    8 of the sequences 000, 001, 010, 011, 100, 101, 110, 111 start with the Gödel number of a terminating program (000, 001, 010, 011, 100, 101, 110, 111).
    Result: $\Omega_3 = 8/8 = 1$

- Repeat the calculation with the following formula from Section 6.2:

$$\Omega_n = \sum_{\substack{P \text{ halts,} \\ |P| \leq n}} \frac{1}{2^{|P|}}$$

  Explain your findings.

Left: $\Omega_3 = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} = 1$
Right: $\Omega_3 = \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} = 1\frac{1}{8}$

Observation:

1. Although there are the same number of terminating programs on the left and on the right, the formula yields a different result. Therefore, we cannot calculate back from Ω_n how many terminating programs actually exist.
2. The value of Ω_n can even become greater than 1. Obviously, Ω_n is no longer a probability.

Conclusion: the prefix-freeness is essential. Without this assumption, the derived formulas for Ω_n and for Ω do not make sense.

Exercise 7.1

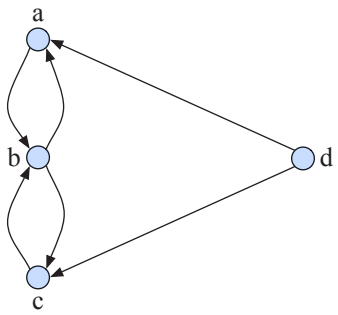
In this exercise, we utilize first-order predicate logic to describe directed graphs. We select the set of individuals U of an interpretation (U,I) as the set of graph nodes and let the binary predicate symbol E determine whether there is an edge between two nodes, x and y. The following example illustrates the relationship between interpretations and graphs:

- Interpretation (U,I)

$$U := \{\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{d}\}$$

$$I(\mathsf{E}) := \{(\mathsf{a},\mathsf{b}),\ (\mathsf{b},\mathsf{a}),\ (\mathsf{b},\mathsf{c}),\ (\mathsf{c},\mathsf{b}),\ (\mathsf{d},\mathsf{a}),\ (\mathsf{d},\mathsf{c})\}$$

- Graph G



Furthermore, the following list of first-order predicate-logic formulas is given:

$$\begin{aligned}
\varphi_1 &:= \neg\mathsf{E}(\mathsf{a},\mathsf{b}) \\
\varphi_2 &:= \neg\exists \mathsf{z}_1\ (\mathsf{E}(\mathsf{a},\mathsf{z}_1) \wedge \mathsf{E}(\mathsf{z}_1,\mathsf{b})) \\
\varphi_3 &:= \neg\exists \mathsf{z}_1\ \exists \mathsf{z}_2\ (\mathsf{E}(\mathsf{a},\mathsf{z}_1) \wedge \mathsf{E}(\mathsf{z}_1,\mathsf{z}_2) \wedge \mathsf{E}(\mathsf{z}_2,\mathsf{b})) \\
\ldots &:= \ldots
\end{aligned}$$

a and b are constant symbols.

a) What is the intuitive meaning of the formula φ_n?

From a to b there is no path of length n.

b) The property of being a connected graph is not definable within first-order predicate logic. Prove this assertion by showing that no PL1 formula φ with the following property exists:

$$(U,I) \models \varphi \ \Leftrightarrow\ (U,I) \text{ describes a connected graph}$$

Remember: A graph is connected if there is a path between two arbitrary nodes, x and y. A path from x to y is a finite sequence of edges connecting x and y in the desired direction.

We assume that the formula φ exists and consider the set

$$M := \{\varphi, \varphi_1, \varphi_2, \ldots\}$$

Every finite subset of M is satisfiable. Finally, we can construct a graph consisting of $n+2$ nodes that connects nodes a and b only via a path of length $n+1$. In this, n is the largest index that occurs in the formulas φ_n.

According to the compactness theorem, M should then also be solvable, which is impossible. At the same time, there should be a connected graph that does not have a path of length 1, a path of length 2, or a path of length 3, etc. between any two nodes.

Exercise 7.2

Let K be a formal system capable of deriving arithmetical statements. Suppose K fulfills the following property:

$$\vdash \varphi \;\Leftrightarrow\; \not\models \varphi$$

- Theorems of K are, for example, $1+1=3$, $\exists x \forall y\, x > y$, $\neg\exists y\, 1+y=2, \ldots$
- No theorems of K are, for example, $1+1=2$, $\forall x \exists y\, x \leq y$, $\exists y\, 1+y=2, \ldots$

K is the perfect liar as its axioms allow us to derive precisely those arithmetic formulas being false in the standard model of Peano arithmetic.

a) Can the symbols ‘s’, ‘$=$’, ‘$+$’, and ‘$\times$’ be reinterpreted such that K has a model?

Yes. The argument is consistent (there is no formula for which both φ and $\neg\varphi$ are derivable, since one of these two formulas must be true). From the model existence theorem, it follows that K has a model.

b) Can a formal system with the postulated property exist at all?

No. K cannot exist. We could transform K into K' with

$$K \vdash \varphi \;\Leftrightarrow\; K' \vdash \neg\varphi$$

K' would then be a complete and correct calculus for Peano arithmetic, which cannot exist according to Gödel’s incompleteness theorem.

Exercise 7.3

Let φ be a formula of first-order predicate logic with equality. Are the following statements true or false? Justify your answers.

a) If φ has a finite model, it also has an infinite model.

The statement is false. For example, the formula

$$\exists x \, \forall y \, (x \doteq y)$$

is true if and only if the domain contains a single element.

b) If φ has an infinite model, it also has a finite model.

The statement is false. For example, the formula

$$\forall x \, \forall y \, \forall z \, (P(x,y) \wedge P(y,z) \rightarrow P(x,z)) \wedge \forall x \, (\neg P(x,x) \wedge \exists y \, P(x,y))$$

is true if and only if P is interpreted as a transitive relation in which each element is related to another, but never to itself. However, this is only possible if the domain is infinite. Thus, φ has an infinite, but no finite model. Note that we did not actually need the equal sign in the constructed formula. The same argument also applies in predicate logic without equality.

c) Can a) or b) be answered with the help of the Löwenheim-Skolem-Tarski theorem?

For part a): No. The condition of the Löwenheim-Skolem-Tarski theorem that an infinite model exists is not fulfilled.

For part b): No. The Löwenheim-Skolem-Tarski theorem states that models of any *transfinite* cardinality exist. However, it does not say anything about finite models.

Exercise 7.4

Euclid's theorem postulates the existence of infinitely many prime numbers, which implies that no natural number x is dividable by every prime number. Peano arithmetic can express this fact as follows:

$$\varphi := \neg\exists \mathsf{x}\, \forall \mathsf{y}\, (\mathrm{prime}(\mathsf{y}) \rightarrow \mathsf{y} \mid \mathsf{x})$$

The formula φ is apparently a true statement in the standard model of Peano arithmetic. As usual, we assume Peano arithmetic is free of contradictions.

- Show that φ is unprovable within Peano arithmetic.

  The formula is unprovable because PA has a nonstandard model in which φ is a false proposition. We deduce the existence of this model from the compactness theorem. Let

  $$\varphi_y := \exists \mathsf{z}\, \mathsf{z} \times \bar{y} = \mathsf{c}$$

  The formula expresses that y is a divisor of c. Now we consider the set

  $$\{\varphi_p \mid p \text{ ist eine Primzahl}\}$$

  Every finite subset has a model. The constant c is assigned the product of all prime numbers occurring in the subset. It follows that the set itself also has a model. In this model, the constant c is assigned an element that is divided by all prime numbers.

- Is the unprovability of φ a consequence of Gödel's incompleteness theorem?

  No. The undecidability arises solely from the fact that the formula φ is not true in all models in which the PA axioms are true. This is due to the fact that PA is not categorical. However, this has nothing to do with Gödel's incompleteness theorem. The theorem also applies, for example, in categorical theories such as Peano arithmetic, formulated as a second-order theory.

Exercise 7.5

In our wording, the model existence theorem also applies to predicate logic with equality. If we restrict ourselves to PL1 without equality, we can even tighten it a bit:

Theorem 7.1 (Model Existence, PL1 Without Equality)

Let K be a first-order theory. Then:

$$K \text{ has a denumerable model} \Leftrightarrow K \text{ is consistent}$$

The difference, though subtle, is significant: In predicate logic without equality, consistency does not only imply the existence of a model but the existence of a *denumerable* model

a) What are the consequences regarding the Löwenheim-Skolem-Tarski theorem?

This sentence can be strengthened, too. Every formula with a model then has an infinite one. The sentence then reads as follows:

b) Show that the generalization does not apply to predicate logic with equality.

The formula

$$\exists x \forall y\, x = y$$

is true precisely among those interpretations that have a one-element domain. The formula therefore has a model, but not a countable one.

Let $M := \{N \subseteq \mathbb{N} \mid 1 \in N\}$

Exercise 7.6

a) Which of the following statements are true? Which are false?

	True	False
■ M is a filter.	✔	◯
■ M ist ein maximaler Filter.	✔	◯
■ M ist ein Ultrafilter.	✔	◯
■ M ist ein freier Ultrafilter.	◯	✔

b) Let F be an ultrafilter. What can we say about its elements if it contains the set $\{1\}$?

F is a filter, so all supersets also belong to F:

$$\{N \subseteq \mathbb{N} \mid 1 \in N\} \subseteq F$$

Does it include additional sets? The answer is no! We assume there is such a set and call it M. Obviously, M cannot contain the number 1. But then the complement $\overline{M}$ would already be in F. Since a set and its complement cannot be contained in the same filter, M cannot be included. Thus:

$$F = \{N \subseteq \mathbb{N} \mid 1 \in N\}$$

Exercise 7.7

In this exercise, let E_1 and E_2 be two enumerators with:

$$E_1 := \{5,8,9,18,19,25,33\}$$
$$E_2 := \{8,12,19,23,24,25,31,32,39,49,60\}$$

a) Create a matrix representation of E_1 and E_2:

	0	1	2	3	4	5	x
0	0	1	3	6	10	15	…
1	2	4	7	11	16	22	…
2	5	8	12	17	23	30	…
3	9	13	18	24	31	39	…
4	14	19	25	32	40	49	…
5	20	26	33	41	50	60	…
y	↑	↑	↑	↑	↑	↑	⋱
	$\{2,3\}$	$\{2,4\}$	$\{3,4,5\}$	$\emptyset$	$\emptyset$	$\emptyset$	

	0	1	2	3	4	5	x
0	0	1	3	6	10	15	…
1	2	4	7	11	16	22	…
2	5	8	12	17	23	30	…
3	9	13	18	24	31	39	…
4	14	19	25	32	40	49	…
5	20	26	33	41	50	60	…
y	↑	↑	↑	↑	↑	↑	⋱
	$\emptyset$	$\{2,4\}$	$\{2,4\}$	$\{3,4\}$	$\{2,3\}$	$\{3,4,5\}$	

b) For which of the following sets is E_1 or E_2 an enumerator?

$$M_1 := \{\emptyset,\{2,3\},\{2,4\}\} \qquad M_3 := \{\{2,3\},\{2,4\}\}$$
$$M_2 := \{\emptyset,\{2,3\},\{2,4\},\{3,4\},\{3,4,5\}\} \qquad M_4 := \{\{2,3\},\{2,4\},\{3,4\},\{3,4,5\}\}$$

E_1 is an enumerator for M_1 and M_3.
E_2 is an enumerator for M_1, M_2, M_3, and M_4.

Exercise 7.8

From Section 7.3, we know that an enumerator of a set M is, by definition, also an enumerator for all subsets of M.

Suppose for the moment that we had dispensed with the subset rule. Then, a set $E \in \mathcal{P}(\mathbb{N})$ would be an enumerator for a set $M \subset \mathcal{P}(\mathbb{N})$ if M contains precisely those sets that occur in the matrix representation of E. The set

$$E = \{0, 4, 12, 24, 40, 60\}$$

would, therefore, only be an enumerator for the set

$$M = \{\{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}.$$

	0	1	2	3	4	5	x
0	0	1	3	6	10	15	…
1	2	4	7	11	16	22	…
2	5	8	12	17	23	30	…
3	9	13	18	24	31	39	…
4	14	19	25	32	40	49	…
5	20	26	33	41	50	60	…
y	↑	↑	↑	↑	↑	↑	⋱
	{0}	{1}	{2}	{3}	{4}	{5}	

At first glance, this definition better captures our intuitive idea of an enumerator than the original, yet we have deliberately refrained from using it. What might be the reasons?

The empty set would present a problem. We have no simple way of distinguishing in matrix form between the empty set (all the entries in a column would be free) and a column that has been left empty (all the entries in the column would then also be free). This means that we have no way of telling whether an enumerator describes a set that contains the empty set or not. The subset rule is an elegant way of getting around this problem.

Exercise 7.9

✍✍✍

a) Does the triple $(\mathcal{P}(\emptyset), \cap, \cup)$ constitute a Boolean algebra?

Yes, it is the smallest possible Boolean algebra that exists. It has exactly one element.

b) Prove the existence of a finite Boolean algebra with an odd number of elements.

First, we derive two important intermediate results:

(1) If $|V| \geq 2$, then $1 \neq 0$.
Proof: Suppose that $1 = 0$. If $|V| \geq 2$, then there exists an x with $x \neq 0$. For this x we have $x = x \wedge 1 = x \wedge 0 = 0$. Contradiction! □

(2) Ist $|V| \geq 2$, so ist immer $\neg x \neq x$.
Proof. $\neg x = x$ implies $x = x \vee x = x \vee \neg x = 1$ and at the same time $x = x \wedge x = x \wedge \neg x = 0$.
Thus, we would have $1 = 0$. Contradiction. □

Now we define the relation '$\sim$' on V as follows:

$$x \sim y \Leftrightarrow y = x \text{ oder } y = \neg x$$

It is easy to verify that '$\sim$' is an equivalence relation ('$\sim$' is reflexive, symmetrical, and transitive). This means that '$\sim$' partitions the set V. Specifically, '$\sim$' groups together every $x \in V$ with its inverse element into an equivalence class:

$$[x]_{\sim} = \{x, \neg x\}$$

If $|V| \geq 2$, then each equivalence class contains exactly 2 elements because of (2). This means that the number of equivalence classes, multiplied by 2, exactly yields the number of elements of V. For $|V| \geq 2$, $|V|$ is therefore always an even number.

Exercise 7.10

The following list includes several laws that hold in every Boolean algebra. Derive each law from Huntington's four axioms.

- Associative laws

$$a \vee (b \vee c) = (a \vee b) \vee c$$
$$a \wedge (b \wedge c) = (a \wedge b) \wedge c$$

- Idempotence laws

$$a \vee a = a$$
$$a \wedge a = a$$

- Absorption laws

$$a \vee (a \wedge b) = a$$
$$a \wedge (a \vee b) = a$$

- De Morgan's laws

$$\neg(a \vee b) = \neg a \wedge \neg b$$
$$\neg(a \wedge b) = \neg a \vee \neg b$$

- Annulment laws

$$a \vee 1 = 1$$
$$a \wedge 0 = 0$$

- Double negation law

$$\neg\neg a = a$$

Derivation of $x \vee x = x$

$$\begin{aligned} & x \vee x \\ = {} & (x \vee x) \wedge 1 \\ = {} & (x \vee x) \wedge (x \vee \neg x) \\ = {} & x \vee (x \wedge \neg x) \\ = {} & x \vee 0 \\ = {} & x \end{aligned}$$

Derivation of $x \vee 1 = 1$

$$\begin{aligned} & x \vee 1 \\ = {} & (x \vee 1) \wedge 1 \\ = {} & (x \vee 1) \wedge (x \vee \neg x) \\ = {} & x \vee (1 \wedge \neg x) \\ = {} & x \vee \neg x \\ = {} & 1 \end{aligned}$$

Derivation of $x \vee (x \wedge y) = x$

$$
\begin{aligned}
& x \vee (x \wedge y) \\
= {} & (x \wedge 1) \vee (x \wedge y) \\
= {} & x \wedge (1 \vee y) \\
= {} & x \wedge (y \vee 1) \\
= {} & x \wedge 1 \\
= {} & x
\end{aligned}
$$

Derivation of $x \vee (y \vee z) = (x \vee y) \vee z$

$$
\begin{aligned}
& x \vee (y \vee z) \\
= {} & (x \vee (y \vee z)) \wedge 1 \\
= {} & (x \vee (y \vee z)) \wedge (x \vee \neg x) \\
= {} & [(x \vee (y \vee z)) \wedge x] \vee [(x \vee (y \vee z)) \wedge \neg x] \\
= {} & [x] \vee [(x \vee (y \vee z)) \wedge \neg x] \\
= {} & [x \vee (x \wedge z)] \vee [(x \vee (y \vee z)) \wedge \neg x] \\
= {} & [(x \wedge (x \vee y)) \vee (x \wedge z)] \vee [(x \vee (y \vee z)) \wedge \neg x] \\
= {} & [x \wedge ((x \vee y) \vee z)] \vee [(x \vee (y \vee z)) \wedge \neg x] \\
= {} & [((x \vee y) \vee z) \wedge x] \vee [(x \vee (y \vee z)) \wedge \neg x] \\
= {} & [((x \vee y) \vee z) \wedge x] \vee [\neg x \wedge (x \vee (y \vee z))] \\
= {} & [((x \vee y) \vee z) \wedge x] \vee [(\neg x \wedge x) \vee (\neg x \wedge (y \vee z))] \\
= {} & [((x \vee y) \vee z) \wedge x] \vee [0 \vee (\neg x \wedge (y \vee z))] \\
= {} & [((x \vee y) \vee z) \wedge x] \vee [\neg x \wedge (y \vee z)] \\
= {} & [((x \vee y) \vee z) \wedge x] \vee [(\neg x \wedge y) \vee (\neg x \wedge z)] \\
= {} & [((x \vee y) \vee z) \wedge x] \vee [(0 \vee (\neg x \wedge y)) \vee (\neg x \wedge z)] \\
= {} & [((x \vee y) \vee z) \wedge x] \vee [((\neg x \wedge x) \vee (\neg x \wedge y)) \vee (\neg x \wedge z)] \\
= {} & [((x \vee y) \vee z) \wedge x] \vee [(\neg x \wedge (x \vee y)) \vee (\neg x \wedge z)] \\
= {} & [((x \vee y) \vee z) \wedge x] \vee [\neg x \wedge ((x \vee y) \vee z)] \\
= {} & [((x \vee y) \vee z) \wedge x] \vee [((x \vee y) \vee z) \wedge \neg x] \\
= {} & ((x \vee y) \vee z) \wedge [x \vee \neg x] \\
= {} & ((x \vee y) \vee z) \wedge 1 \\
= {} & (x \vee y) \vee z
\end{aligned}
$$

Derivation of $\neg\neg x = x$

$$\neg\neg x$$

$$
\begin{aligned}
&= \neg\neg x \wedge 1 \\
&= \neg\neg x \wedge (x \vee \neg x) \\
&= (\neg\neg x \wedge x) \vee (\neg\neg x \wedge \neg x) \\
&= (\neg\neg x \wedge x) \vee 0 \\
&= (\neg\neg x \wedge x) \vee (x \wedge \neg x) \\
&= (\neg\neg x \wedge x) \vee (\neg x \wedge x) \\
&= (\neg\neg x \vee \neg x) \wedge x \\
&= (\neg x \vee \neg\neg x) \wedge x \\
&= 1 \wedge x \\
&= x
\end{aligned}
$$

Derivation of $\neg(x \vee y) = \neg x \wedge \neg y$

$$
\begin{aligned}
&\neg(x \vee y) \\
&= \neg((x \vee y) \vee 0) \\
&= \neg((x \vee y) \vee ((\neg x \wedge \neg y) \wedge \neg(\neg x \wedge \neg y))) \\
&= \neg(((x \vee y) \vee (\neg x \wedge \neg y)) \wedge ((x \vee y) \vee \neg(\neg x \wedge \neg y))) \\
&= \neg(((x \vee y \vee \neg x) \wedge (x \vee y \vee \neg y)) \wedge ((x \vee y) \vee \neg(\neg x \wedge \neg y))) \\
&= \neg((((x \vee \neg x) \vee y) \wedge (x \vee (y \vee \neg y))) \wedge ((x \vee y) \vee \neg(\neg x \wedge \neg y))) \\
&= \neg(((1 \vee y) \wedge (x \vee 1)) \wedge ((x \vee y) \vee \neg(\neg x \wedge \neg y))) \\
&= \neg((1 \wedge 1) \wedge ((x \vee y) \vee \neg(\neg x \wedge \neg y))) \\
&= \neg(1 \wedge ((x \vee y) \vee \neg(\neg x \wedge \neg y))) \\
&= \neg((\neg(\neg x \wedge \neg y) \vee (\neg x \wedge \neg y)) \wedge ((x \vee y) \vee \neg(\neg x \wedge \neg y))) \\
&= \neg(\neg(\neg x \wedge \neg y) \vee ((\neg x \wedge \neg y) \wedge (x \vee y))) \\
&= \neg(\neg(\neg x \wedge \neg y) \vee ((\neg x \wedge \neg y \wedge x) \vee (\neg x \wedge \neg y \wedge y))) \\
&= \neg(\neg(\neg x \wedge \neg y) \vee (((\neg x \wedge x) \wedge \neg y) \vee (\neg x \wedge (\neg y \wedge y)))) \\
&= \neg(\neg(\neg x \wedge \neg y) \vee ((0 \wedge \neg y) \vee (\neg x \wedge 0))) \\
&= \neg(\neg(\neg x \wedge \neg y) \vee (0 \vee 0)) \\
&= \neg(\neg(\neg x \wedge \neg y) \vee 0) \\
&= \neg\neg(\neg x \wedge \neg y) \\
&= \neg x \wedge \neg y
\end{aligned}
$$

The complementary rules are derived in the same way.

Exercise 7.11

In Section 7.4.1, we introduced the '$\leq$' relation as an order on the elements of a Boolean algebra. Show the following equivalences:

$$a \leq b \;\Leftrightarrow\; a \vee b = b \;\Leftrightarrow\; a \to b = 1$$

The relationship $a \leq b$ is equivalent to $a \wedge b = a$.

1. We show: $a \wedge b = a \Rightarrow a \vee b = b$

$$\begin{aligned} a \wedge b = a \Rightarrow a \vee b &= (a \wedge b) \vee b \\ &= b \end{aligned}$$

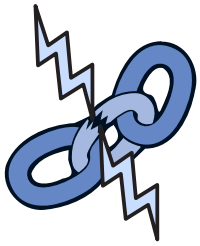
2. We show: $a \vee b = b \Rightarrow a \to b = 1$

$$\begin{aligned} a \vee b = b \Rightarrow a \to b &= \neg a \vee b \\ &= \neg a \vee (a \vee b) \\ &= (\neg a \vee a) \vee b \\ &= 1 \end{aligned}$$

3. We show: $a \to b = 1 \Rightarrow a \wedge b = a$

$$\begin{aligned} a \to b = 1 \Rightarrow a \wedge b &= (a \wedge \neg a) \vee (a \wedge b) \\ &= a \wedge (\neg a \vee b) \\ &= a \wedge (a \to b) \\ &= a \wedge 1 \\ &= a \end{aligned}$$

Exercise 7.12



In Section 7.5, we have utilized the forcing technique to extend the ground model $\mathcal{M}$ to a model $\mathcal{M}[G]$ that contains a bijection between $\mathbb{R}$ and $\aleph_2$. Applying the forcing machinery to $\mathcal{M}[G]$ a second time generates another model with a bijection between $\mathbb{R}$ and $\aleph_3$, for instance. Consequently, this model would also contain a bijection between $\aleph_2$ and $\aleph_3$, which is impossible.

Where is this argument flawed?

It was missed that both the concept of a power set and the concept of a cardinal number are relative. This means that the sets that take on the role of the real numbers or the role of the cardinal numbers $\aleph_2$ or $\aleph_3$ vary from model to model. Consequently, the set that establishes a bijection between $\mathbb{R}$ and $\aleph_2$ in $\mathcal{M}[G]$ is also present in the third model, which we will call $\mathcal{M}[GG]$. However, in that model it is merely a bijection between sets that no longer necessarily represent the real numbers or the cardinal number $\aleph_2$. Thus, we only have the following relationship:

$$\mathcal{M}[GG] \models |\aleph_2^{\mathcal{M}[G]}| = |\aleph_3^{\mathcal{M}[G]}| \tag{7.12}$$

Symbolically, the contradiction suggested in the text means

$$\mathcal{M}[GG] \models |\aleph_2^{\mathcal{M}[GG]}| = |\aleph_3^{\mathcal{M}[GG]}|,$$

,which is different from (7.12).

In summary, the following applies: in the description of the exercise, the principle of absoluteness was assumed for quantities whose meaning is relative.

Exercise 7.13

In this exercise, let us try to apply Theorem 7.13 to the half-order $(P,\supseteq)$ given by:

$$P := \{f : \aleph_1^{\mathcal{M}} \to \mathbb{R}^{\mathcal{M}} \mid f \in \mathcal{M} \text{ and } \mathrm{dom}(f) \text{ is countable in } \mathcal{M}\}$$

Expressed in plain words, P comprises all partial functions from $\mathcal{M}$ that are defined on a countable subset of $\aleph_1^{\mathcal{M}}$ and map to sets that belong to the real numbers in $\mathcal{M}$. The superset relation orders the elements of P.

Next, let us define the following sets:

$$D_\alpha := \{p \in P \mid \alpha \in \mathrm{dom}(p)\} \tag{7.13}$$
$$D_x := \{p \in P \mid x \in \mathrm{ran}(p)\} \tag{7.14}$$

Verbally put, set D_α contains those functions from P defined at $\alpha \in \aleph_1^{\mathcal{M}}$, and set D_x contains all functions from P that map at least one element to the real number $x \in^{\mathcal{M}}$.

a) Show that the two sets are dense in P for all $\alpha \in \aleph_1^{\mathcal{M}}$ and all $x \in \mathbb{R}^{\mathcal{M}}$.

1. We show: For all $q \in P$ there exists $p \in D_\alpha$ with $p \leq q$.
   If $\alpha \in \mathrm{dom}(q)$, then we set $p := q$ and we are done. If $\alpha \notin \mathrm{dom}(q)$, then we set $p := q \cup \{(\alpha, 0)\}$. Then $p \in D_\alpha$ and $p \leq q$ applies, which had to be shown.
2. We show: For all $q \in P$ there exists $p \in D_x$ with $p \leq q$.
   If $x \in \mathrm{ran}(q)$, then we set $p := q$ and we are done. If $x \notin \mathrm{ran}(q)$, then we choose an $\alpha \in \aleph_1^{\mathcal{M}}$ with $\alpha \notin \mathrm{dom}(q)$ and set $p := q \cup (\alpha, x)$. Then $p \in D_x$ and $p \leq q$, which was to be shown. That such an α always exists follows from the property of P that it only contains functions whose domain in $\mathcal{M}$ is at most denumerable. The domain of such functions cannot completely cover the set $\aleph_1^{\mathcal{M}}$.

b) Let G be a generic filter on P. Show that the set $\bigcup G$ is a total and surjective function from $\aleph_1^{\mathcal{M}}$ to $\mathbb{R}^{\mathcal{M}}$.

1. We show: f is a partial function from $\aleph_1^{\mathcal{M}}$ to $\mathbb{R}$.
   We have to show that the elements in G are pairwise compatible functions, i.e., that any two functions $p, q \in G$ can always be extended to a common function. Such an extension is possible if and only if p and q assign the same value to every α lying in the intersection of their domains.
   The compatibility of p, q follows from the filter property of G. From $p, q \in G$ it follows that an element r is contained in G with $r \leq p$ and $r \leq q$. However, the element r can only be contained in P if it is a function, and this means that the function values of p and q must match in the intersection of their domains.
2. We show: f is total and surjective.
   Both properties of f follow from the property of G of being generic. This property means that G intersects all dense subsets and thus in particular the sets D_α and D_x, for all $\alpha \in \aleph_1^{\mathcal{M}}$ and all $x \in \mathbb{R}$. On the one hand, this shows that the function f must be defined everywhere (totality), and on the other hand, that its image covers the real numbers completely (surjectivity).

c) Part b) implies $\mathcal{M}[G] \models |\mathbb{R}^{\mathcal{M}}| \leq |\aleph_1^{\mathcal{M}}|$. Does this mean that the continuum hypothesis is true in $\mathcal{M}[G]$?

The continuum hypothesis is indeed true in $\mathcal{M}[G]$, but we cannot derive this result directly from part b). The point of this task is to see why this is so. The relation

$$\mathcal{M}[G] \models |\mathbb{R}^{\mathcal{M}}| \leq \aleph_1^{\mathcal{M}} \tag{7.15}$$

is not what we need. It makes a statement about the sets $\mathbb{R}^{\mathcal{M}}$ and $\aleph_1^{\mathcal{M}}$, that is, about the sets that take on the role of the real numbers and the role of the cardinal number $\aleph_1$ in $\mathcal{M}$. However, we cannot rely on these sets being the same in $\mathcal{M}[G]$. What we need is a statement about the sets $\mathbb{R}^{\mathcal{M}[G]}$ and $\aleph_1^{\mathcal{M}[G]}$ and not about the sets $\mathbb{R}^{\mathcal{M}}$ and $\aleph_1^{\mathcal{M}}$.
To complete the last step of the proof, a property of $(P, \leq)$ is needed that is called σ-completeness and lies outside of what we can discuss in this book.