Chapter 1

Propositional Logic

Mathematical logic studies correct thinking, correct deductions of statements from other statements. Let us make it more precise. A fundamental property of a statement is that it may be true or false. Whether a statement is true or false is called its truth value. Logic is a systematic study of how statements can be related in ways that capture their respective truth values and how from statements (assumptions) correctly deduce other statements. Knowing which deductions are "logically" correct, we can determine the truth value not only by looking at the words but by looking at the relationship to other statements.

We do not define what a statement is (we also did not define a set, a point etc.). For the reader's convenience, we only describe, in an intuitive way, what we mean by a statement: By a statement we will understand something which is said about the world, and something which has a truth value.

From elementary statements more complicated ones are built, and the truth value of these statements is then determined by the basic ones. To form more complicated statements we use the following logical connectives:

- it is not the case that; we denote it by $\neg$, and call it the *negation*;
- and; we denote it by $\wedge$, and call it the *conjunction*;
- or; we denote it by $\vee$, and call it the *disjunction*;
- if . . . then; we denote it by $\Rightarrow$, and call it the *implication*;
- if and only if; we denote it by $\Leftrightarrow$, and call it the *equivalence.*

1.1 Formal Syntax of Propositional Logic

1.1.1 Definition of a Formula. Given a non-empty set A of *logical variables* (we also call them *elementary statements*, or *propositional variables*). A finite sequence of elements of the set A, of logical connectives and parentheses is called a *propositional formula* (or shortly a *formula*), if it is formed by the following rules:

1. Every logical variable (elementary statement) $a \in A$ is a propositional formula.
2. If α, β are propositional formulas, then so are $(\neg\alpha)$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \Rightarrow \beta)$, and $(\alpha \Leftrightarrow \beta)$.
3. Only sequences that were formed by using finitely many applications of rules 1 and 2, are propositional formulas.

The set of all propositional formulas, that were formed from the logical variables from the set A is denoted by $\mathcal{P}(\mathcal{A})$. □

1.1.2 Remark and Notation. The connective $\neg$ is called *unary*, since it forms a new formula from one formula. The other connectives are called *binary*, since they need two formulas to form a new one.

In what follows, we will always denote logical (propositional) variables by small letters: e.g. a, b, c, $\dots$, x, y, z, $\dots$. Propositional formulas will be denoted by small Greek letters: e.g. α, β, γ, $\dots$, φ, ψ, $\dots$

1.1.3 Convention. We will use two rules about usage of parenthesis:

1. We omit the outward parenthesis. For example, we will write $(\alpha \Rightarrow \beta) \Rightarrow \beta$ instead of $((\alpha \Rightarrow \beta) \Rightarrow \beta)$.
2. We assume that the unary connective $\neg$ "is stronger than" each of the binary ones. Hence, if α and β are formulas then we write $\neg\alpha \Rightarrow \beta$ instead of $(\neg\alpha) \Rightarrow \beta$, $\neg\alpha \vee \beta$ instead of $(\neg\alpha) \vee \beta$, etc. After all, you know such situation in arithmetic. For example, $-2 + 3$ is interpreted as $(-2) + 3$, and not as $-(2 + 3)$.

1.1.4 Syntactic Tree of a Formula. A *syntactic tree* of a formula φ captures its structure; it is a rooted tree where each vertex which is not a leave is labeled by a logical connective and has either one son if the connective is $\neg$, or two sons if the connective is $\wedge$, $\vee$, $\Rightarrow$, or $\Leftrightarrow$. The leaves are labeled by logical variables.

A syntactic tree is also called *derivation tree*.

The *depth of a formula* is defined as the height of the syntactic tree of the formula.

1.1.5 Subformulas of a Given Formula. A *subformula* of a formula α is any substring of α that is a formula itself. □

We can also say that a subformula of α is any string which corresponds to a subtree of the syntactic tree of α.

1.2 Semantics in Propositional Logic

Now we will be interested in the fact whether a correctly formed formula is either true or false. For this we will use the notion of a *truth valuation*.

1.2.1 Truth Valuations of Formulas.

Definition. Given a nonempty set of logical variables A. A mapping $u\colon \mathcal{P}(A) \to \{0, 1\}$ is called a *truth valuation*, if it satisfies the following rules

(1) $u(\neg\alpha) = 1$ if and only if $u(\alpha) = 0$;

(2) $u(\alpha \wedge \beta) = 1$ if and only if $u(\alpha) = u(\beta) = 1$;

(3) $u(\alpha \vee \beta) = 0$ if and only if $u(\alpha) = u(\beta) = 0$;

(4) $u(\alpha \Rightarrow \beta) = 0$ if and only if $u(\alpha) = 1$ and $u(\beta) = 0$;

(5) $u(\alpha \Leftrightarrow \beta) = 1$ if and only if $u(\alpha) = u(\beta)$.

□

Here $u(\alpha) = 1$ means that the formula α is true; and $u(\alpha) = 0$ means that the formula α is false.

1.2.2 Truth Tables. The properties that any truth valuation must have, can also be expressed in terms of the truth tables of the logical connectives. These are:

α	$\neg\alpha$
0	1
1	0

α	β	$\alpha \wedge \beta$	$\alpha \vee \beta$	$\alpha \Rightarrow \beta$	$\alpha \Leftrightarrow \beta$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

1.2.3 How Many Different Truth Valuation There Are? The answer depends on the number of logical variables. We shall show that if A has n logical variables then there are 2^n distinct truth valuations. For this the following proposition is the key.

1.2.4 Proposition. Every mapping $u_0 : A \to \{0, 1\}$ can be uniquely extended to a truth valuation. It means that there is a unique truth valuation $u : \mathcal{P}(A) \to \{0, 1\}$ such that $u_0(a) = u(a)$ for all $a \in A$.

Moreover, two truth valuations $u, v : \mathcal{P}(A) \to \{0, 1\}$ coincide if and only if $u(x) = v(x)$ for every logical variable $x \in A$. □

Justification. Consider any formula α and its syntactic tree. Evaluate all logical variables by their values in u_0 (i.e. x has the value $u(x) = u_0(x)$). Each vertex in the syntactic tree has the value given by the connective and the value/s of its son/s. Now, the value of the root of the syntactic tree is the truth value of the whole formula.

Notice that the above justification may be turned to an exact proof if mathematical induction is used on the depth of the formula α.

1.2.5 Corollary. Let A contain n logical variables. Then there exist 2^n distinct truth valuations. □

Justification. The above proposition 1.2.4 tells us that the number of distinct truth valuations is the number of distinct mappings $u_0 : A \to \{0, 1\}$. And there are 2^n of them.

Remark. Similarly as we formed truth tables for logical connectives we can also form truth tables for any formula. From the above corollary we know that such a truth table will have 2^n rows provided the formula has n logical variables.

1.2.6 A Tautology, a Contradiction, a Satisfiable Formula. Now we can divide formulas into different groups according to their truth values in all valuations.

Definition.

1. A formula is called a *tautology* provided it is true for all truth valuations.
2. A formula is called a *contradiction* provided it is false for all truth valuations.
3. A formula is *satisfiable* provided there is at least one truth valuation for which the formula is true.

□

Remark. It is evident that a negation of any tautology is a contradiction, and conversely, a negation of a contradiction is always a tautology.

For instance, $a \vee \neg a$, $a \Rightarrow a$ are tautologies, whereas $a \wedge \neg a$ is a contradiction. Every tautology is a satisfiable formula, but there are satisfiable formulas that are not tautologies. Indeed, $a \Rightarrow \neg a$ is such an example.

1.2.7 Tautological Equivalence. Formulas of propositional logic are defined as strings of symbols (see 1.1.1), so two formulas are the same if and only if they are the same as strings. Hence the equality of formulas is a very strict notion; indeed, formulas $a \wedge b$ and $b \wedge a$ are different whereas everybody feels that meaning of the conjunction of two formulas does not depend on their order. So we need a new notion for ”equality” of formulas; and the notion is the tautological equivalence. There is the formal definition:

Definition. We say that formulas φ and ψ are *tautologically equivalent* (also *semantically equivalent*), if they have the same value in every truth valuation, i.e. if $u(\varphi) = u(\psi)$ for every truth valuation u.

The fact that φ and ψ are tautologically equivalent is denoted by $\varphi \models\!\!\!\dashv \psi$. □

1.2.8 Examples. It is very easy to verify that the following tautological equivalences are valid (indeed, it suffices to form corresponding truth tables):

1. $\alpha \wedge \alpha \models\!\!\!\dashv \alpha$, $\alpha \vee \alpha \models\!\!\!\dashv \alpha$;
2. $\alpha \wedge \beta \models\!\!\!\dashv \beta \wedge \alpha$, $\alpha \vee \beta \models\!\!\!\dashv \beta \vee \alpha$ (commutativity of $\wedge$ and $\vee$);
3. $\alpha \wedge (\beta \wedge \gamma) \models\!\!\!\dashv (\alpha \wedge \beta) \wedge \gamma$, $\alpha \vee (\beta \vee \gamma) \models\!\!\!\dashv (\alpha \vee \beta) \vee \gamma$ (associativity of $\wedge$ and $\vee$);
4. $\alpha \wedge (\beta \vee \alpha) \models\!\!\!\dashv \alpha$, $\alpha \vee (\beta \wedge \alpha) \models\!\!\!\dashv \alpha$ (absorption of $\wedge$ and $\vee$);
5. $\neg\neg\alpha \models\!\!\!\dashv \alpha$ (double negation);
6. $\neg(\alpha \wedge \beta) \models\!\!\!\dashv (\neg\alpha \vee \neg\beta)$, $\neg(\alpha \vee \beta) \models\!\!\!\dashv (\neg\alpha \wedge \neg\beta)$ (de Morgan's laws);
7. $\alpha \wedge (\beta \vee \gamma) \models\!\!\!\dashv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$, $\alpha \vee (\beta \wedge \gamma) \models\!\!\!\dashv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ (distributivity laws).

If moreover **T** is any tautology and **F** is any contradiction, then

8. $\mathbf{T} \wedge \alpha \models\!\!\!\dashv \alpha$, $\mathbf{T} \vee \alpha \models\!\!\!\dashv \mathbf{T}$, $\mathbf{F} \wedge \alpha \models\!\!\!\dashv \mathbf{F}$, $\mathbf{F} \vee \alpha \models\!\!\!\dashv \alpha$;
9. $\alpha \wedge \neg\alpha \models\!\!\!\dashv \mathbf{F}$, $\alpha \vee \neg\alpha \models\!\!\!\dashv \mathbf{T}$.

□

You may notice that some of the above facts are analogous to the properties of the set operations union, intersection, and complement. It is not surprising since these set operations correspond to logical connectives disjunction, conjunction, and negation.

1.2.9 Properties of Tautological Equivalence. There are other properties that the tautological equivalence has and that are useful if we are looking for a simple formula which is tautologically equivalent to a given one. For this the following two propositions play a crucial role.

Proposition. Tautological equivalence satisfies the following properties: For every formulas α, β and γ

1. $\alpha \models\!\!\!\dashv \alpha$;
2. if $\alpha \models\!\!\!\dashv \beta$ then $\beta \models\!\!\!\dashv \alpha$;
3. if $\alpha \models\!\!\!\dashv \beta$ and $\beta \models\!\!\!\dashv \gamma$ then $\alpha \models\!\!\!\dashv \gamma$.

□

Theorem. Let α, β, γ, and δ be formulas satisfying $\alpha \models\!\!\!\dashv \beta$ and $\gamma \models\!\!\!\dashv \delta$. Then

1. $\neg\alpha \models\!\!\!\dashv \neg\beta$;
2. $(\alpha \wedge \gamma) \models\!\!\!\dashv (\beta \wedge \delta)$, $(\alpha \vee \gamma) \models\!\!\!\dashv (\beta \vee \delta)$, $(\alpha \Rightarrow \gamma) \models\!\!\!\dashv (\beta \Rightarrow \delta)$, $(\alpha \Leftrightarrow \gamma) \models\!\!\!\dashv (\beta \Leftrightarrow \delta)$.

□

Justification of the above Proposition and Theorem is straightforward and is left as an exercise.

1.2.10 Remark. A formula was defined, see 1.1.1, as a correctly formed string of logical variables, logical connectives ($\neg$, $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$), and paranthesis. We could started with only four connectives; indeed, any formula of the form $\alpha \Leftrightarrow \beta$ can be rewritten using the following tautological equivalence

$$\alpha \Leftrightarrow \beta \models\!\!\!\dashv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$$

and we get a tautologically equivalent formula that contains only $\neg$, $\vee$, $\wedge$ and $\Rightarrow$.

Similarly, we can introduce **F** as a new symbol, representing a formula that is false in any truth valuation, so it represents a contradiction. Hence, for example $x \Rightarrow \mathbf{F}$ is a well formed formula, moreover $x \Rightarrow \mathbf{F} \models\!\!\!\dashv \neg x$.

1.3 Semantical Consequence

Our main aim of this section is to give a more precise meaning to the concept of ”correct reasoning”, i.e. how to correctly deduce statements/formulas from a given set of formulas/assumptions. What we mean by it: Given a set of *assumptions*, represented by a set of formulas S, we will be looking for formulas that can be *deduced* from S.

Before doing this we specify what we mean by a *set of formulas is true in a truth valuation.*

1.3.1 Definition. Given a truth valuation u and a set of formulas S. We say that S is *true* in u, (or that S is satisfied in u), if every formula from S is true in u. In other words, $u(\varphi) = 1$ for all $\varphi \in S$.

A set of formulas is said to be *satisfiable* if it is true in at least one truth valuation. Otherwise, it is called *unsatisfiable*. □

We will write $u(S) = 1$ whenever S is a set of formulas, u a truth valuation such that S is true in u.

Example. For instance, the set $\{a \Rightarrow b, \neg b\}$ is true for u where $u(a) = 0 = u(b)$, so it is satisfiable. On the other hand, the set $\{a, a \Rightarrow b, \neg b\}$ is unsatisfiable.

1.3.2 Semantical Consequence.

Definition. We say that a formula φ is a *semantical consequence* of a set of formulas S, (or also that φ is an *entailment* of the set S, or that φ *semantically follows* from the set S), provided φ is true for every truth valuation u for which the set S is true.

The fact that formula φ is a semantical consequence of the set S is denoted by $S \models \varphi$. □

Convention. If S is a one element set, e.g. $S = \{\alpha\}$, and $S \models \varphi$, then we write $\alpha \models \varphi$ instead of $\{\alpha\} \models \varphi$.

If S is the empty set $\emptyset$, and $S \models \varphi$, then we write $\models \varphi$ instead of $\emptyset \models \varphi$.

Remarks. 1. It is easy to notice that $S \models \varphi$ if and only if for every valuation u it holds that $u(S) \leq u(\varphi)$.

2. Let us observe that one can verify semantical consequences using truth tables; indeed, we first form truth tables for all formulas in S and for the formula φ. Then we look at all the rows where all formulas from S have 1. In all these rows the formula φ must have 1 as well.

1.3.3 Examples. Let us give couple of examples; they are entailments that are commonly used in many ”real life deductions”.

For all formulas α, β we have

1. $\{\alpha, \alpha \Rightarrow \beta\} \models \beta$;
2. $\{\alpha \Rightarrow \beta, \neg\beta\} \models \neg\alpha$;
3. $\{\alpha, \neg\alpha\} \models \beta$.

The justification of the above examples is straightforward.

1.3.4 Properties of Semantical Consequence. Let us state several properties that semantical consequences have.

Proposition. For every two formulas φ, ψ we have

- $\varphi \models\!\mid \psi$ if and only if $\varphi \models \psi$ and $\psi \models \varphi$.
- We have $\varphi \models \psi$ if and only if the formula $\varphi \Rightarrow \psi$ is a tautology.

□

Justification. 1. $\varphi \models\!\!\!\dashv \psi$ means that $u(\varphi) = u(\psi)$ for every truth valuation u. So if $u(\varphi) = 1$ then $u(\psi) = 1$, and if $u(\psi) = 1$ then $u(\varphi) = 1$, which proves the first part of the proposition.

2. Assume that $\varphi \models \psi$ and take arbitrary truth valuation u. Then either $u(\varphi) = 1$, or $u(\varphi) = 0$. In the first case, $u(\psi) = 1$ since $\varphi \models \psi$, and hence $u(\varphi \Rightarrow \psi) = 1$. In the latter case, i.e. if $u(\varphi) = 0$, then from the properties of implication $u(\varphi \Rightarrow \psi) = 1$ as well. Hence $\varphi \Rightarrow \psi$ is a tautology.

Assume that $\varphi \Rightarrow \psi$ is a tautology. Then it cannot happen that $u(\varphi) = 1$ and $u(\psi) = 0$ for any truth valuation u. Hence $\varphi \models \psi$.

1.3.5 More Advanced Properties. We state two further properties that are true for semantical consequence. The first one is a base for so called resolution method. The second one is the Deduction Theorem for propositional logic. We will not prove them, the proofs are not difficult and are left to the readers.

Theorem. Let S be a set of formulas and φ a formula. Then

$$S \models \varphi \quad \text{if and only if} \quad S \cup \{\neg\varphi\} \text{ is unsatisfiable.}$$

Deduction Theorem. Let S be a set of formulas, α and β two formulas. Then

$$S \models (\alpha \Rightarrow \beta) \quad \text{if and only if} \quad S \cup \{\alpha\} \models \beta.$$

1.4 Boolean Calculus

Propositional logic has lot of applications, for example in the theory of logical circuits. In many applications it is useful to form new "operations" capturing the behavior of logical connectives conjunction, disjunction, and negation.

1.4.1 Logical Operations. Given a truth valuation u and two logical variables a and b. Denote $x = u(a)$ and $y = u(b)$. Then the following holds:

$$\begin{aligned} u(a \vee b) &= \max\{u(a), u(b)\} = \max\{x, y\}, \\ u(a \wedge b) &= \min\{u(a), u(b)\} = \min\{x, y\}, \\ u(\neg a) &= 1 - u(a) = 1 - x. \end{aligned}$$

It motivates the following definition of *boolean operations* for $x, y \in \{0, 1\}$:

$$\begin{array}{lcll} x \cdot y & = & \min\{x, y\} & \text{(product)}, \\ x + y & = & \max\{x, y\} & \text{(logical sum)}, \\ \overline{x} & = & 1 - x & \text{(complement)}. \end{array}$$

1.4.2 Properties of Logical Operations. Let us reformulate the properties of logical connectives of $\neg$, $\vee$ and $\wedge$ given in 1.2.8 to the properties of boolean operations.

Proposition. For all $x, y, z \in \{0, 1\}$ we have:

1. $x \cdot x = x,\ x + x = x$;
2. $x \cdot y = y \cdot x,\ x + y = y + x$;
3. $x \cdot (y \cdot z) = (x \cdot y) \cdot z,\ x + (y + z) = (x + y) + z$;
4. $x \cdot (y + x) = x,\ x + (y \cdot x) = x$;
5. $x \cdot (y + z) = (x \cdot y) + (x \cdot z),\ x + (y \cdot z) = (x + y) \cdot (x + z)$;

6. $\overline{\overline{x}} = x$;
7. $\overline{x+y} = \overline{x} \cdot \overline{y}$, $\overline{x \cdot y} = \overline{x} + \overline{y}$;
8. $x + \overline{x} = 1$, $x \cdot \overline{x} = 0$;
9. $x \cdot 0 = 0$, $x \cdot 1 = x$;
10. $x + 1 = 1$, $x + 0 = x$.

1.4.3 Boolean Functions. To every formula φ with n logical variables $a_1, \ldots, a_n$ one can assign a function $f\colon \{0,1\}^n \to \{0,1\}$ of n variables $x_1, \ldots, x_n$ defined

$$f(x_1, \ldots, x_n) = u(\varphi) \quad \text{for} \quad u(a_i) = x_i, i = 1, \ldots, n.$$

If two formulas α and β are tautologically equivalent, then their corresponding functions are the same.

Definition. A function $f\colon \{0,1\}^n \to \{0,1\}$ is called a *boolean function of n variables*, where n is a natural number. □

Proposition. To every boolean function $f\colon \{0,1\}^n \to \{0,1\}$ there is a formula α which corresponds to f. □

Notice that the above proposition means that any boolean function can be written as an expression of boolean operations. For example, the boolean function f corresponding to the formula $a \Rightarrow b$ can be written as $f(x,y) = \overline{x} + y$ since $a \Rightarrow b \models\!\!\!\dashv \neg a \vee b$.

Chapter 2

Predicate Logic

Propositional logic is not able to describe all entailments which we consider to be ”logically correct”. This is because logical variables are the simplest formulas in propositional logic and they do not have any inner structure. And the simplest formulas do need inner structure in some entailments. Let us give an example of such deduction.

Consider the following inference:

Peter can play violin.
Everybody who can play violin has a musical ear.
Peter has a musical ear.

If we try to describe the above sentences as logical variables, then the first sentence would be one logical variable, say a. The second sentence resembles an implication of two formulas, say $b \Rightarrow c$. The third sentence is again a logical variable, say d. But $\{a, b \Rightarrow c\} \not\models d$.

We will see that the above inference is correct in predicate logic. Let us start with an informal description of formulas in predicate logic.

2.1 Syntactics of Predicate Logic

2.1.1 An Informal Description of Predicate Logic. For this purpose we will use predicates – properties, and quantifiers.

What do we intuitively need for a description of the sentence “Peter can play violin.”? The statement speaks about “Peter” as an object and a property which Peter has, i.e. the property ”to be able to play violin”. Also the third sentence “Peter has a musical ear.” has a similar structure. The property here is “to have a musical ear”. The middle sentence speaks about “everybody”, where by “everybody” we mean “every object” (every human being). In fact, it has a form of an implication: Every object that has the property “to be able to play violin”, also has the property of “having a musical ear”.

Let us denote by V the property “to be able to play violin” and by E the property “to have a musical ear”. Now, we can write the above inference as:

Peter has V.
Everybody who has V, has E.
Peter has E.

Even this is a rather long description. We will write $V(p)$ instead of “Peter can V.”, where p denotes Peter. Similarly, we shorten the description of the third sentence, to $E(p)$. To shorten also the second sentence, we introduce an abbreviation for “everybody”, in other words, “for all objects”. We will denote it by $\forall$ and call it the *universal quantifier.* Any quantifier must be followed by a variable. The expression $\forall x$ is read as “for every x”. Of course, we do not say only “for every” we must say “for every object” (and explain where are the objects from). Variables will be denoted by $x, y, \ldots$, and they represent objects. Now the second sentence has the following form: $\forall x\,(V(x) \Rightarrow E(x))$. Hence the whole entailment is written as follows:

$$\begin{array}{l} V(p) \\ \forall x\,(V(x) \Rightarrow E(x)) \\ \hline E(p) \end{array}$$

2.1.2 Example. We give another inference which is typical for predicate logic.

$$\begin{array}{l} \text{If a natural number is even, then its successor is odd.} \\ \text{The number 2 is even.} \\ \hline \text{The successor of the number 2 is odd.} \end{array}$$

Properties (predicate symbols) contained in the inference are: “to be an even number”, we will denote it by E, and “to be an odd number”, which will be denoted by O. Furthermore, we have one object (a constant symbol) which is the number 2. Finally, we need to describe “a successor of a natural number”. Here, by a successor we do not understand the property “to be a successor”, we do not ask whether a number is the successor of another one or not. Here we want to work with the successor of a natural number as with a natural number, i.e. with an object. Of course, this object is indirectly described — by means of a number which precedes it. Therefore, we will look at “a successor” as a function. We denote it by f, and it assigns to every natural number its successor, i.e. $f\colon n \mapsto n+1$. The symbol f will be referred as a functional symbol.

Now, we are able to formalize the second and the third sentences: $E(2)$ and $O(f(2))$. The first sentence contains a quantifier. It is not completely clear from the English formulation which quantifier it should be. This ambiguity of the formulation, i.e. whether our statement should refer to “all” objects or only to “some” object, is common for nearly all natural languages. Hence we recommend to the reader to try to reformulate the sentence under consideration so that the quantification will be clearer. In our case, the first sentence has the same meaning as the statement “Whenever a natural number is even, its successor is an odd number.”. Therefore we get: $\forall x\,(E(x) \Rightarrow O(f(x)))$. The whole inference is:

$$\begin{array}{l} \forall x\,(E(x) \Rightarrow O(f(x))) \\ E(2) \\ \hline O(f(2)) \end{array}$$

2.2 Formal Description of Formulas in Predicate Logic.

Let us now give a formal definition of formulas of predicate logic. For this we need first to define a language of predicate logic which will capture ”objects”, ”properties”, ”special objects”, ”mappings”, and ”quantifiers”.

2.2.1 The Language of Predicate Logic.

Definition. A *language of predicate logic* consists of

1. *logical symbols*, i.e.:

   a) infinite countable set Var of individual variables
   b) propositional logical connectives: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
   c) universal quantifier $\forall$ and existential quantifier $\exists$

2. *special symbols*, i.e.:

   a) a set Pred of predicate symbols (it cannot be empty)
   b) a set Cons of constant symbols (it may be empty)
   c) a set Func of functional symbols (it may be empty)

3. auxiliary symbols, such as brackets “[,]”, parentheses “(,)”, and commas “,”.

□

For every predicate or functional symbol a natural number is given n, for a predicate symbol $n \geq 0$, for a functional symbol $n \geq 1$. This number indicates how many objects the predicate symbol concerns, or how many variables the functional symbol has. We call the number the *arity* of a predicate or functional symbol.

Remark. The fact that a predicate symbol P has arity 0 means that "it does not concern any individual variable", so it is non structured and it can be considered as a logical variable. In such a way, propositional logic is contained in predicate logic.

2.2.2 Terms. First, we introduce a notion of a term: Roughly speaking, a term is an object which can be described using variables, constants and functional symbols also in a more complicated way. In the above examples the terms were for instance "Peter", "Peter's father", "a successor of the number 2". In the language of predicate logic, terms play the role of "substantives".

Definition. The set of *terms* is defined by the following rules:

1. Every variable and every constant symbol is a term.
2. If f is a functional symbol of arity n and $t_1, t_2, \dots, t_n$ are terms, then $f(t_1, t_2, \dots, t_n)$ is also a term.
3. Anything that was not created by a finite use of the above rules 1 and 2 is not a term.

□

2.2.3 Atomic formulas. If we know what are terms (i.e. objects with which our statements deal), we can start to form atomic formulas of predicate logic:

Definition. An *atomic formula* is a predicate symbol P applied to as many terms as its arity is. In other words, if the arity of a predicate symbol P is n and $t_1, t_2, \dots, t_n$ is an n-tuple of terms, then $P(t_1, t_2, \dots, t_n)$ is an atomic formula. □

2.2.4 Formulas of Predicate logic. Analogously as in propositional logic we form also more complicated formulas than atomic ones.

Definition. The set of *formulas* is defined by the following rules:

1. Every atomic formula is a formula.
2. If φ and ψ are formulas, then $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \Rightarrow \psi)$, $(\varphi \Leftrightarrow \psi)$ are also formulas.
3. If φ is a formula and x a variable, then $(\forall x\, \varphi)$ and $(\exists x\, \varphi)$ are also formulas.
4. Anything that was not created by a finite use of the above rules 1, 2 and 3 is not a formula.

□

We will use similar convention as in propositional logic and we will not write the outward parenthesis. Also note that again the negation "is stronger than other connectives", so we $\neg\alpha$ and not $(\neg\alpha)$.

2.2.5 Syntactic Tree of a Formula. We form the *syntactic tree* for every formula of predicate logic similarly as we did for formulas of propositional logic. The only differences are:

- Inner vertex can be labeled by one of the two quantifiers ($\forall$ and $\exists$) followed by a variable. Such vertex has one son (i.e. a quantifier is considered as "unary").
- We form also a syntactic tree of any atomic formula: The root of this subtree is labeled by the corresponding predicate symbol and it has as many sons as is its arity. The subtrees of the sons correspond to the syntactic trees of the terms written from the left to the right.

2.2.6 Subformulas A *subformula* of a formula φ is any substring of φ which is itself a correctly formed formula. □

Given a formula $\beta = (\forall x\ (P(x) \vee Q(x,y))) \Rightarrow R(a,x)$. Then β has subformulas $P(x)$, $Q(x,y)$, $R(a,x)$, $P(x) \vee Q(x,y)$, $\forall x\,(P(x) \vee Q(x,y))$, and finally itself $(\forall x\,(P(x) \vee Q(x,y))) \Rightarrow R(a,x)$.

In other words: A subformula of a formula φ is any string that corresponds to a subtree of the syntactic tree of the formula φ, determined by some vertex labeled by a predicate symbol, by a logical connective, or by a quantifier.

2.2.7 Bound and Free Variables. According to the definition of a formula in 2.2.4 the following string $(\forall x\ P(x)) \Rightarrow (\exists y\ R(x,y))$ is a correctly formed formula of predicate logic. This formula was constructed using $\Rightarrow$ from two formulas, $\forall x\ P(x)$ and $\exists y\ R(x,y)$. One can see that the variable x in $\forall x\ P(x)$ has nothing in common with the variable x in $\exists y\ R(x,y)$. To understand better the difference between two different occurrences of x as in the above example we introduce:

Definition. We say that an occurrence of a variable x is *bounded* in a formula φ, provided there is a vertex labeled by a quantifier with this variable x on the path from the leaf labeled by this x to the root (backwards) in the syntactic tree. Otherwise we speak of a *free* occurrence of the variable x. □

2.2.8 Sentence. A formula that does not have free variables is called a *closed formula* or a *sentence*. A formula that does not have bounded variables is called an *open formula*.

Note that there are formulas that are both closed and open; indeed, any formula that does not have variables (only constant symbols) is at the same time closed and open. Obviously, the formula above contains a variable, x, which has both free and bounded occurrence.

2.3 Semantics of Predicate Logic

Similarly as in propositional logic we need to study what does it mean that a predicate logic formula is true or false. In propositional logic it suffices to declare which of the logical variables are true (and which are false) and we easily get the truth value of the whole formula. The truthfulness was then established by a truth valuation, i.e. a mapping from the set of all logical variables into $\{0,1\}$. In predicate logic the situation is considerably more difficult; the crucial notion will be an *interpretation* which will help us to find out which sentence (a closed formula) is true and which is false in the given interpretation.

This section follows the structure of corresponding section about propositional logic.

2.3.1 Informal Description of an Interpretation Let us start with an example. Given a formula

$$\forall x\,(S(x) \Rightarrow S(f(f(x))))$$

where S is a unary predicate symbol and f is a unary functional symbol (obviously, x is a variable).

The above formula can have several meanings. We state two of them.

1. Objects are people. S represents the property "to be alive". The function f assigns to every person his/her father. The formula corresponds to the sentence: "Everybody who is alive has a grandfather from the father side who is also alive.". And, of course, this is a false statement.

2. Objects are natural numbers. S represents the property "to be an even number", f assigns to every natural number its successor. The formula corresponds to the sentence: "For every even natural number the successor of its successor is also an even number." And, of course, it is a true statement.

It is evident from the example above that in order to decide whether a given formula is true or false we need to know the "meaning" of all special symbols. That is the meaning of all predicate symbols, constant symbols, and functional symbols, and what our "word" is, i.e. from what set our objects will be taken. Moreover, we must have only a sentence (e.g. for the following formula $x > 5$ we cannot decide whether it is true or false unless x is bounded by one of the quantifiers).

2.3.2 Formal Definition of an Interpretation. An *interpretation* of predicate logic with the set of predicate symbols Pred, the set of constant symbols Cons, and the set of functional symbols Func is a pair $\langle U, [\![-]\!]\rangle$, where

- U is a non-empty set called universe of domain;
- $[\![-]\!]$ is an assignment which
  1. to every predicate symbol $P \in \mathsf{Pred}$ of arity n it assigns a subset $[\![P]\!]$ of U^n, (we will see later that it is in fact an n-ary relation on the set U),
  2. to every constant symbol $a \in \mathsf{Cons}$ it assigns an element of U; we denote it by $[\![a]\!]$,
  3. to every functional symbol $f \in \mathsf{Func}$ of arity n it assigns a mapping from the set U^n into U; we denote it by $[\![f]\!]$.

2.3.3 Informal Definition of Truth Value of a Sentence in an Interpretation. First, we have to interpret a term in a given interpretation. If we know what should be substituted for a variable, we can then find the value of the term similarly as evaluation of algebraic expressions (values of constants are given by the interpretation as well as the meaning of the functional symbols).

An atomic formula $P(t_1, \ldots, t_n)$ is true if the n-touple $(o_1, \ldots, o_n)$ has the property $[\![P]\!]$, where o_i, $i = 1, \ldots, n$ are values of terms $t_1, \ldots, t_n$.

If we know the truth value of formulas α and β, then the truth value of $\neg\alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \Rightarrow \beta$ and $\alpha \Leftrightarrow \beta$ are defined as in the propositional logic.

If x is a variable and α is a formula with free variable x, then

- $\forall x\, \alpha$ is true if and only if for every $d \in U$ substituting d for x yields a true formula.
- $\exists x\, \alpha$ is true if and only if there exists at least one $d \in U$ such that if we substitute d for x we get a true formula.

2.3.4 A Model of a Sentence Given a sentence φ. Any interpretation in which φ is true is called a *model* of φ. □

So, the second interpretations from 2.3.1 is a model of the sentence $\alpha = \forall\, x\, (S(x) \Rightarrow S(n(n(x)))$, whereas the first one is a model of $\neg\alpha$.

2.3.5 A Tautology, a Contradiction, a Satisfiable Sentence. Similarly as in propositional logic we can define:

Definition. A sentence is called a *tautology* provided it is true in every interpretation; it is called a *contradiction* provided it is false in every interpretation. A sentence is *satisfiable* provided there is at least one interpretation in which the formula is true. □

We can reformulate the above definition as follows: A satisfiable sentence is a sentence that has a model; a contradiction is a sentence that does not have a model; and a tautology is a sentence for which every interpretation is a model.

Note that there is infinitely many different interpretations of an even simple sentence as $\forall x\, P(x)$. Indeed, on any set we can define a subset corresponding to P. Hence, in predicate logic there is nothing like "truth table" for a sentence.

Examples. Given a unary predicate P and a constant a. Then

1. $\alpha = (\forall x\, P(x)) \vee \neg(\forall x\, P(x))$, $\beta = (\forall x\, P(x)) \Rightarrow (\exists x\, P(x))$, $\gamma = \neg(\forall x\, P(x)) \Leftrightarrow (\exists x\, \neg P(x))$, and $\delta = \neg(\exists x\, P(x)) \Leftrightarrow (\forall x\, \neg P(x))$ are tautologies.
2. $\mu = (\exists x\, P(x) \wedge (\forall x\, \neg P(x))$ is a contradiction. Also the negation of any tautology is a contradiction.
3. $\alpha = P(a)$, $\beta = (\exists x\, P(x)) \Rightarrow P(a)$ are satisfiable sentences which are not tautologies.

Note that x is a variable, otherwise the above examples were not syntactically correct formulas.

2.3.6 Tautological Equivalence.

Definition. We say that two sentences φ and ψ are *tautologically equivalent* (also *semantically equivalent*), if they are either both true or both false in every interpretation. □

2.3.7 Examples. Let us give a couple of examples of tautologically equivalent sentences that are typical for predicate logic.

1. $\forall x\, \forall y\, Q(x,y) \models\!\dashv \forall y\, \forall x\, Q(x,y)$,
2. $\exists x\, \exists y\, Q(x,y) \models\!\dashv \exists y\, \exists x\, Q(x,y)$.
3. $\neg(\forall x\, P(x)) \models\!\dashv (\exists x\, \neg P(x))$;
4. $\neg(\exists x\, P(x)) \models\!\dashv (\forall x\, \neg P(x))$;
5. $(\forall x\, P(x)) \vee (\forall y\, Q(y)) \models\!\dashv \forall x\, \forall y\, (P(x) \vee Q(y))$;
6. $(\exists x\, P(x)) \wedge (\exists y\, Q(y)) \models\!\dashv \exists x\, \exists y\, (P(x) \wedge Q(y))$.

2.3.8 Semantical consequence. Similarly as in the propositional logic we will introduce the notion of semantical consequence to capture correct entailments in predicate logic.

First let us say that a set of sentences S is *true* in an interpretation $\langle U, [\![-]\!]\rangle$, if every sentence from S is true in $\langle U, [\![-]\!]\rangle$.

Definition. We say that a sentence φ is a *semantical consequence* of a set of sentences S, (or also an *entailment* of the set S, or that φ *semantically follows* from the set S), provided φ is true in every interpretation $\langle U, [\![-]\!]\rangle$ in which the set S is true. We denote this fact by

$$S \models \varphi.$$

□

In other words, a sentence φ *is not* a semantical consequence of a set of sentences S, provided that there is a model of the set S which is not a model of φ. That is, there exists an interpretation $\langle U, [\![-]\!]\rangle$, in which every sentence of S is true and the sentence φ is false. Hence, the notion is similar to the semantical consequence in propositional logic, the only difference is that in propositional logic we deal with truth valuations, and in predicate logic we have interpretations.

2.3.9 Example. Let us decide whether $N \models \varphi$ holds, where $N = \{\forall x\, (P(x) \Rightarrow Q(x)), P(a)\}$ and $\varphi = \exists x\, Q(x)$.

Solution. Take any interpretation $\langle U, [\![-]\!]\rangle$, in which both the sentences $\forall x\, (P(x) \Rightarrow Q(x))$ and $P(a)$ are true. Since the formula $P(a)$ is true, the element $c = [\![a]\!]$ has the property $[\![P]\!]$. It means that $c \in [\![P]\!]$. Since also the formula $\forall x\, (P(x) \Rightarrow Q(x))$ is true in $\langle U, [\![-]\!]\rangle$ and the element c has the property $[\![P]\!]$, the element c has also the property $[\![Q]\!]$. It means that $c \in [\![Q]\!]$. Therefore $[\![Q]\!] \neq \emptyset$, thus the formula $\exists x\, Q(x)$ is true in $\langle U, [\![-]\!]\rangle$.

We have shown that φ is a semantical consequence of the set N.

Chapter 3

Sets and Relations

3.1 Sets

In this section we first recall some well known facts about sets and operations with them. You probably know them already. The main focus will be on the notion of *cardinality* of a set, which for finite sets coincide with the number of elements. The cardinality will be defined in such a way that it can be used also for infinite sets, and we show that there are infinite sets with “smaller” number of elements, i.e. a smaller cardinality than other infinite sets.

Let us start with the notion of a set. A set is what mathematicians call a collection of objects that can be distinguished from each other and these objects we call elements. For a well known sets we use common notation — $\mathbb{N}$ for the set of all natural numbers, $\mathbb{Z}$ for the set of all integers, $\mathbb{Q}$ for the set of all rational numbers, and $\mathbb{R}$ for the set of all real numbers.

3.1.1 Principle of Extensionality. Two sets S and T are equal (we write $S = T$) if and only if every element of the set S is an element of the set T and conversely, every element of T is an element of S.

3.1.2 A set can be given by listing its elements, or by a property that characterizes its elements. If $p(x)$ is a property which an element has or does not have then the set C consisting of all elements having the property $p(x)$ (and no others) is denoted by

$$C = \{x \mid p(x)\}.$$

For example, the set of all squares of numbers 1, ..., 4 is either $S = \{1, 4, 9, 16\}$, or as

$$S = \{x \mid x = y^2, y \in \mathbb{N}, 0 < y < 5\}.$$

The set of all even natural numbers is $\{m \mid m = 2k, k \in \mathbb{N}\}$. The set of all odd numbers is $\{m \mid m = 2k + 1, k \in \mathbb{N}\}$.

3.1.3 Subsets.

Definition. Given two sets S and T. If every element of the set S is also an element of T, we say that S is a *subset* of T, and we write $S \subseteq T$.

If $S \subseteq T$ and S and T are different sets, we also say that S is a *proper subset* of T. □

Note that equivalently we have: T is *not* a subset of S if and only if there is an element x for which $x \in T$ and $x \notin S$.

3.1.4 Proposition. For all sets S, T we have $S = T$ if and only if $S \subseteq T$ and $T \subseteq S$. □

Justification. $S \subseteq T$ and $T \subseteq S$ mean precisely that S and T have the same elements, so they are equal.

3.1.5 Empty Set.

A very important role among sets is played by so called empty set.

Definition. The *empty set* is a set that contains no element; we denote it $\emptyset$. □

Fact. We have $\emptyset \subseteq A$ for every set A.

Justification. It cannot happen that $\emptyset$ **is not** a subset of a set A. Indeed, $\emptyset \not\subseteq A$ means that there is $x \in \emptyset$ and $x \notin A$. But there is no $x \in \emptyset$. So, $\emptyset \not\subseteq A$ cannot be true.

3.1.6 Set Operations.

We recall the well known set operations.

Given two sets A and B. Their *union* is the set

$$A \cup B = \{x \mid x \in A \ \text{ or } x \in B\};$$

their *intersection* is the set

$$A \cap B = \{x \mid x \in A \ \text{ and } x \in B\}.$$

The *difference* of two sets A and B (in this order) is the set

$$A \setminus B = \{x \mid x \in A \text{ and } x \notin B\}.$$

If there is a fixed "universal" set U and $A \subseteq U$ then by the *complement* of the set A in U we mean the set $U \setminus A$. If the universe U is generally understood, then we write only $\overline{A}$ for the complement of A in U.

The *Cartesian product* of A and B (we denote it by $A \times B$) is

$$A \times B = \{(a,b) \mid a \in A, b \in B\}.$$

If $A = B$ we speak about a *Cartesian power* of the set A and write A^2 instead of $A \times A$, A^3 is the set of all triples of elements of A, More precisely,

$$A^3 = \{(a,b,c) \,|\, a,b,c \in A\}.$$

Similarly, for natural number $n \geq 2$ we have

$$A^n = \{(a_1, a_2, \ldots, a_n) \,|\, a_i \in A\}.$$

Examples. We have $\{1,2,4\} \cup \{1,4,5,6\} = \{1,2,4,5,6\}$ and $\{1,2,4\} \cap \{1,4,5,6\} = \{1,4\}$. $\{1,2,4\} \setminus \{1,4,5,6\} = \{2\}$. $\{1,4,5,6\} \setminus \{1,2,4\} = \{5,6\}$.

For example, $\{1,4\} \times \{1,2,6\} = \{(1,1),(1,2),(1,6),(4,1),(4,2),(4,6)\}$; the second Cartesian power of $A = \{1,4\}$ is $A^2 = \{(1,1),(1,4),(4,1),(4,4)\}$.

3.1.7 Disjoint Sets.

Definition. If $A \cap B = \emptyset$, we say that the sets A and B are *disjoint.* Evidently, the set of even natural numbers is disjoint with the set of odd natural numbers, and their union is the whole set $\mathbb{N}$ of all natural numbers. □

3.1.8 Power Sets.

Definition. Let A be a set. The *power set* $P(A)$ of the set A is the set of all subsets of the set A; or more formally,

$$P(A) = \{B \mid B \subseteq A\}.$$

□

For example, $P(\emptyset) = \{\emptyset\}$ and $P(\{1,2,3\}) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$. Let us point out that a power set is always non-empty, since it contains the empty set $\emptyset$.

3.1.9 Characteristic Function of a Subset. We introduce a new notion – so called a characteristic function of a given subset. You may come across it when representing a set in computer programs. The notion can also be used for counting the number of subsets of a given set (as we see later).

Definition. A *characteristic function* χ_A of a subset $A \subseteq U$ is the mapping $\chi_A \colon U \to \{0,1\}$ defined by

$$\chi_A(x) = \begin{cases} 1 & \text{for } x \in A; \\ 0 & \text{for } x \in U \setminus A. \end{cases}$$

Note that $\chi_A(x)$ can be viewed as the answer to the question: Does x belong to A? Here $\chi_A(x) = 1$ means yes, and $\chi_A(x) = 0$ means no.

Let A and B be two different subsets of the set U. Then their characteristic functions are different. Indeed, if $A \neq B$ then there is an element x which is in one of the sets and does not belong to the other set. For the sake of simplicity, let us assume that $x \in A$ and $x \notin B$. Then $\chi_A(x) = 1 \neq 0 = \chi_B(x)$ and therefore $\chi_A \neq \chi_B$.

Every mapping $\chi \colon U \to \{0,1\}$ is a characteristic function of a suitable subset of U. Indeed, define a subset C of U by $C = \{x \mid \chi(x) = 1\}$. Then $\chi = \chi_C$.

Thus subsets of a given set U and characteristic functions from U into $\{0,1\}$ are only two equivalent descriptions of the same ideal reality. A simple consequence of this fact is the following fact: *Let U be any finite set with n elements. Then there are exactly 2^n subsets of U.* (In other words, for a finite set U with n elements the power set $P(U)$ has 2^n elements.)

3.2 Cardinality of Sets

In this section we present a formal approach to the intuitive notion of the size of a set, and when two sets have the same size. The reader is familiar with *finite sets*. A set is infinite if it is not a finite one. We will show that among infinite sets we can speak of sets with “the same size” and that there are infinite sets of “different sizes”. At first we must recall what a bijection, i.e. a bijective mapping means.

3.2.1 Bijections. Let us recall that a mapping f from a set A into a set B is *bijective* if is injective and surjective. A mapping f is *injective* (or *one-to-one*) provided for every two different elements $x, y \in A$ their images $f(x)$, $f(y)$ are also different. A mapping is *surjective* (or *onto*) provided for every element $y \in B$ there exists an element $x \in A$ such that $y = f(x)$.

3.2.2 Sets with Same Cardinality.

Definition. Two sets A and B are said to have the *same cardinality* if there exists a bijective mapping from A onto B. This fact is denoted by $|A| = |B|$. □

Example. The set S of all even natural numbers has the same cardinality as the set T of all odd natural numbers. Indeed, the mapping

$$f \colon S \to T \text{ defined by } 2n \mapsto 2n+1$$

is a bijective mapping from S onto T.

3.2.3 Countable Sets. The “smallest” cardinality of an infinite set is the cardinality of the set of all natural numbers $\mathbb{N}$. Countable sets are in fact sets that have the same cardinality as $\mathbb{N}$.

Definition. We say that a set A is *countable* provided it has the same cardinality as the set of all natural numbers $\mathbb{N}$. If a set A is infinite and not countable then we say that it is *uncountable*. □

The following proposition helps us to easily decide whether an infinite set is countable or not.

3.2.4 Proposition. A set A is countable if and only if it can be arranged in an injective infinite sequence, (i.e. a sequence which is infinite and where no two elements are equal). □

Justification. Assume that we are able to arrange the set A in an injective infinite sequence, say

$$A = \{a_0, a_1, \ldots, a_n, \ldots\}.$$

Define a mapping $f\colon \mathbb{N} \to A$ by $f(i) = a_i$ for every $i \in \mathbb{N}$. Then f is a desired injective mapping from the set $\mathbb{N}$ onto the set A.

Assume we have a bijective mapping f from $\mathbb{N}$ onto A. Then the set A can be written as follows: $A = \{f(0), f(1), \ldots, f(n), \ldots\}$. In other words we have arranged the set A into an injective infinite sequence.

3.2.5 Example of a Countable Set. The set of all integers $\mathbb{Z}$ is countable.

Indeed, $\mathbb{Z}$ can be arranged into an injective infinite sequence. We do it as follows

$$0, 1, -1, 2, -2, 3, -3, 4, -4, \ldots, n, -n, \ldots.$$

More precisely, the number 0 will be in the 0-th place (i.e. 0 is the element a_0), the number 1 in the first place (the element a_1), the number -1 in the second place (the element a_2), the number 2 in the place $2 \cdot 2 - 1 = 3$ (the element a_3), the number -2 in the place $2 \cdot 2 = 4$ (the element a_4), etc. Generally, a positive integer n will be in the place $2n - 1$ (the element a_{2n-1}) and the number $-n$ in the place $2n$ (the element a_{2n}).

Further, we give three propositions about countable sets. The first one in fact shows that a subset of countable set cannot be uncountable; the next two show set operations that maintain countability.

3.2.6 Proposition. Any infinite subset of a countable set is again countable. □

Justification. If we can arrange the set A into an injective infinite sequence $\{a_0, a_1, \ldots\}$ then each of its infinite subsets $B \subseteq A$ can be obtained as a subsequence of the sequence $\{a_0, a_1, \ldots\}$ which is again injective and infinite.

3.2.7 Proposition. If two sets are countable so is their union. □

Justification. Let $A = \{a_0, a_1, a_2, \ldots\}$ and $B = \{b_0, b_1, b_2, \ldots\}$ be two countable sets. Their union $C = A \cup B$ may be written as follows

$$A \cup B = \{a_0, b_0, a_1, b_1, a_2, b_2, \ldots\}.$$

More formally, $A \cup B = \{c_0, c_1, \ldots\}$, where $c_{2n} = a_n$ and $c_{2n+1} = b_n$. This sequence does not need to be injective. Indeed, the elements that are in A and simultaneously in B are in the above sequence twice. But by omitting the second appearance of them we get an injective sequence. Therefore the set $A \cup B$ is countable.

3.2.8 Proposition. The Cartesian product of two countable sets is a countable set. □

Justification. We will show that the set $C = A \times B$, for A and B countable sets, can be arranged into an injective infinite sequence. Let $A = \{a_0, a_1, a_2, \ldots\}$ and $B = \{b_0, b_1, b_2, \ldots\}$ be injective sequences. We will make use of the following scheme.

$$\begin{tikzcd}
(a_0, b_0) & (a_0, b_1) \arrow[dl] & (a_0, b_2) \arrow[dl] & \cdots \\
(a_1, b_0) & (a_1, b_1) \arrow[dl] & (a_1, b_2) \arrow[dl] & \cdots \\
(a_2, b_0) & (a_2, b_1) \arrow[dl] & (a_2, b_2) \arrow[dl] & \cdots \\
\vdots & \vdots & \vdots &
\end{tikzcd}$$

The arrangement of C is seen from the scheme. Indeed,

$$C = \{(a_0, b_0), (a_0, b_1), (a_1, b_0), (a_0, b_2), (a_1, b_1), (a_2, b_0), (a_0, b_3), \ldots\}.$$

Here is the precise description of the sequence: The pair (a_i, b_j) will be in the place $k = i + \frac{(i+j)(i+j+1)}{2}$ of the sequence. Notice that in this case the constructed sequence is injective.

3.2.9 We will use the above proposition to show that well known sets are countable.

Example. The set $\mathbb{Q}$ of all rational numbers is countable.

Justification. Indeed, every rational number can be represented as a fraction $\frac{p}{q}$ where q is a non-zero natural number and p is an integer. Fractions can be viewed as ordered pairs (p, q) where p is the numerator and q the denominator of the fraction $\frac{p}{q}$. Since the set of all integers is countable as well as the set of all non-zero natural numbers, according to 3.2.8 we get that the set M of all ordered pairs (p, q) is countable. The set $\mathbb{Q}$ is now the subset of the set M which contains only those pair (p, q) for which p and q are relatively prime. Since the set $\mathbb{Q}$ is infinite, it is countable by the proposition 3.2.6.

3.2.10 Proposition. The union of a countable system of finite disjoint sets is again countable.

In other words, if $A_0, A_1, \ldots A_n, \ldots$ are finite disjoint sets then their union $A_0 \cup A_1 \cup \ldots = \bigcup_{i \in \mathbb{N}} A_i$ is countable as well. □

Remark. The above proposition can be generalized to countable union of countable sets, but the proof needs axiom of choice. This is beyond the scope of our course.

3.2.11 Uncountable Sets. Recall that an infinite set is said to be uncountable if it is not countable.

In the next paragraph we will show that the set of all subsets of natural number is uncountable. To prove this we will prove the following:

3.2.12 Cantor Diagonal Method. The set of all infinite sequences of 0's and 1's is uncountable. □

Justification. By contradiction. Assume that the set of all infinite sequences of zeros and ones is countable. It means that we are able to arrange it into an infinite sequence. We list all the sequences in the following scheme — in the first row we have the sequence s_0, in the second row the sequence s_1, in the third one s_2, etc.

$$\begin{array}{ccccccccc}
s_0 & = & \boxed{s_0(0),} & s_0(1), & s_0(2), & s_0(3), & s_0(4), & s_0(5), & \ldots \\
s_1 & = & s_1(0), & \boxed{s_1(1),} & s_1(2), & s_1(3), & s_1(4), & s_1(5), & \ldots \\
s_2 & = & s_2(0), & s_2(1), & \boxed{s_2(2),} & s_2(3), & s_2(4), & s_2(5), & \ldots \\
s_3 & = & s_3(0), & s_3(1), & s_3(2), & \boxed{s_3(3),} & s_3(4), & s_3(5), & \ldots \\
s_4 & = & s_4(0), & s_4(1), & s_4(2), & s_4(3), & \boxed{s_4(4),} & s_4(5), & \ldots \\
s_5 & = & s_5(0), & s_5(1), & s_5(2), & s_5(3), & s_5(4), & \boxed{s_5(5),} & \ldots \\
\vdots & & & & & & & &
\end{array}$$

We will construct a new sequence of zeros and ones and we will show that this new sequence is not included into the list above. It is the sequence $\overline{s}$ defined as follows: The sequence $\overline{s}$ begins with 0 if in the first frame above there is 1, and begins with 1 if in the first frame there is 0. In other words, $\overline{s}(0) = 1$ provided $s_0(0) = 0$ and $\overline{s}(0) = 0$ provided $s_0(0) = 1$. Further we proceed analogously. If $s_1(1) = 0$, then $\overline{s}(1) = 1$, and if $s_1(1) = 1$, then $\overline{s}(1) = 0$ (notice that in the above scheme the value $s_1(1)$ is put into a frame). The value of $\overline{s}(2)$ is the number $1 - s_2(2)$, etc.

More formally: $\overline{s} = \{\overline{s}(0), \overline{s}(1), \overline{s}(2), \ldots, \overline{s}(n), \ldots\}$, where $\overline{s}(n) = 1 - s_n(n)$ for all $n \in \mathbb{N}$.

The sequence $\overline{s}$ is not equal to any of the sequences $s_0, s_1, s_2, \ldots, s_n, \ldots$. Indeed, it differs from s_0 in the zeroth place ($\overline{s}(0) = 1 - s_0(0)$), it differs from s_1 in the first place ($\overline{s}(1) = 1 - s_1(1)$), ... it differs from s_n in the n-th place ($\overline{s}(n) = 1 - s_n(n)$). This means that we have not listed *all* sequences (there is at least one missing — $\overline{s}$), a contradiction.

Therefore the set of all infinite sequences of zeros and ones is not countable.

3.2.13 Theorem. The set of all subsets of natural numbers $\mathbb{N}$ is uncountable. □

Justification. Since each characteristic function of a $A \subseteq \mathbb{N}$ is an infinite sequence of 0's and 1's, the theorem follows from the Cantor diagonal method 3.2.12.

3.3 Binary relations

In mathematics, as in everyday situations, we often speak about a relationship between objects, which means an idea of two objects being related or associated one to another in some way. The notion of a binary relation makes this precise. Let us start with some examples.

1. To be a grandfather. Objects of our consideration are people; a person a is associated with a person b if a is a grandfather of b.

2. To be of the same length. Objects of our consideration are sticks; a stick a is associated with another stick b if both sticks have the same length.

3. To be a subset. Objects of our consideration are subsets of a given set U; a subset X is related to a subset Y if X is a subset of Y.

4. To be greater or equal. Objects of our consideration are numbers; a number n is related to a number m if n is greater than or equal to m.

5. To be a student of a study group. Objects of our consideration are first year students and study groups; a student a is related to a study group number K if student a belongs to study group K.

6. The sine function. Consider real numbers; a number x is related to a number y if $y = \sin x$.

3.3.1 Definition. A *relation* (more precisely a *binary relation*) *from a set A into a set B* is any set of ordered pairs $R \subseteq A \times B$. If $A = B$ we speak about a *relation on a set A*. □

We can construct new relations from others. Since a relation is a set of ordered pairs, we can use set operations for construction of new relations. But there are also specific operations – inverse relation and composition of relations. First we start with set operations.

3.3.2 Set Operations with Relations.

Definition. We say that a relation R is a *sub relation* of a relation S if $R \subseteq S$; i.e. if $a\,R\,b$, then also $a\,S\,b$. □

Definition. Let R and S be two relations from a set A into a set B. The *intersection* of relations R and S is the relation $R \cap S$; the *union* of R and S is the relation $R \cup S$; the *complement* of R is the relation $\overline{R} = (A \times B) \setminus R$. □

For example, let T be equality on the set of all real numbers $\mathbb{R}$, and S be the relation "smaller than" on $\mathbb{R}$. Then $T \cap S = \emptyset$ and $T \cup S$ is the relation to be smaller than or equal to. The complement of the relation T is non-equality on $\mathbb{R}$; i.e. the relation $\overline{T} = \{(a, b) \mid a, b \in \mathbb{R}, a \neq b\}$.

3.3.3 Inverse Relation.

Definition. Let R be a relation from a set A into a set B. Then the *inverse relation* of the relation R is the relation R^{-1} from set B into set A, defined by:

$$x\,R^{-1}y \quad \text{if and only if} \quad y\,R\,x.$$

□

Notice that the inverse relation R^{-1} to R always exists. So if R is a function then the relation R^{-1} exists; on the other hand, R^{-1} does not need to be a function.

3.3.4 Composition of Relations.

Definition. Let R be a relation from a set A into a set B and S be a relation from the set B into a set C. Then the *composition of the relations* (sometimes also called the *product*), $R \circ S$, is the relation from the set A into the set C defined by:

$$a\,(R \circ S)\,c \quad \text{if and only if there is an element } b \in B \text{ such that } \ a\,R\,b \ \text{ and } \ b\,S\,c.$$

□

3.3.5 Properties of Composition of Relations. We will show some properties of composition of relation. First, we prove that a composition of relations is associative, then that it is not commutative. (Roughly speaking, we do not need to use parentheses, but we cannot change the order.)

Proposition. The composition of relations is associative. More precisely, if R is a relation from A to B, S is a relation from B to C, and T is a relation from C to D then

$$R \circ (S \circ T) \,=\, (R \circ S) \circ T\,.$$

□

Justification: It is not difficult to show hat for all elements $a \in A$, $d \in D$ it holds: $a\,R \circ (S \circ T)\,d$ if and only if there exist $b \in B$, $c \in C$ such that $a\,R\,b$, $b\,S\,c$ and $c\,T\,d$. And this means that $a\,(R \circ S) \circ T\,d$.

Proposition. The composition of relations is not commutative. It is not the case that $R \circ S \,=\, S \circ R$ holds for all relations R and S. □

Justification. To show the above proposition it suffices to find two relations S and T for which $R \circ S \,=\, S \circ R$ does not hold despite of the fact that both compositions exist.

Here is an example: Let A be the set of all people in the Czech Republic. Consider the following two relations R, S defined on A:

$$\begin{array}{ll} a\,R\,b & \text{if and only if } a \text{ is a brother or a sister of } b \text{ and } a \neq b \\ c\,S\,d & \text{if and only if } c \text{ is a child of } d. \end{array}$$

To show that $R \circ S \,\neq\, S \circ R$ it suffices to find two people x, y such that $x\,R \circ S\,y$ holds and $x\,S \circ R\,y$ does not hold. Consider any pair of a nephew a and an uncle b. We have $a\,S \circ R\,b$ since a parent of a is a brother or a sister of the uncle b. On the other hand, $a\,R \circ S\,b$ does not hold. Indeed, it would mean that one of the brothers or sisters of a were a parent of uncle b.

3.3.6 Relations on a Set. In applications an important role play relations $S \subseteq A \times B$ where $A = B$. Recall that such relations are called *relations on A*.

3.3.7 Different Types of Relations on A. Relations on a set A may have different properties. We will be mainly interested in four of them: reflexivity, symmetry, antisymmetry and transitivity. Here are the definitions:

Definition. We say that relation R on set A is

1. *reflexive* if for every $a \in A$ we have $a\,R\,a$;
2. *symmetric* if for every $a, b \in A$ it holds that: $a\,R\,b$ implies $b\,R\,a$;
3. *antisymmetric*, if for every $a, b \in A$ it holds that: $a\,R\,b$ and $b\,R\,a$ imply $a = b$;
4. *transitive*, if for every $a, b, c \in A$ it holds that: if $a\,R\,b$ and $b\,R\,c$ then $a\,R\,c$.

□

Examples. Consider the relation of non-equality R on the set of all natural numbers $\mathbb{N}$; (i.e. $n\,R\,m$ if and only if n and m are different natural numbers). This relation is not reflexive because for no $n \in \mathbb{N}$ do we have $n \neq n$. It is symmetric: If $n \neq m$ then also $m \neq n$. Relation R is not antisymmetric because e.g. $2 \neq 3$, $3 \neq 2$, and 2 and 3 are different numbers. (That is $2\,R\,3$ and $3\,R\,2$ and at the same time $2 \neq 3$.) This relation is not transitive because for example we have $2 \neq 3$ and $3 \neq 2$, and at the same time $2 = 2$ (i.e. $2\,R\,3$ and $3\,R\,2$ and it is not true $2\,R\,2$).

Relation "to be smaller than or equal to" $\leq$ on set $\mathbb{R}$ is reflexive, since $a \leq a$ for every a. It is also antisymmetric, since whenever for two numbers a, b we have $a \leq b$ and $b \leq a$, then $a = b$. It is also transitive, since if $a \leq b$ and $b \leq c$, then also $a \leq c$.

3.3.8 Equivalence Relations. One of the most important type of relations on A is so called equivalence relation. Let us recall the tautological equivalence of propositional formulas. It is one example of equivalence relation on the set of all propositional formulas. Have in mind that an "equivalence relation" on A is some sort of "generalized equality" of elements of A.

Definition. A relation R on a set A is called an *equivalence*, if it is reflexive, symmetric and transitive. □

Example. The following relation R on the set of all integers $\mathbb{Z}$, defined by:

$$m\,R\,n \quad \text{if and only if} \quad m - n \text{ is divisible by } 12, \ (m, n \in \mathbb{Z}),$$

is an equivalence relation.

Justification. Relation R is reflexive. Indeed, for every $m \in \mathbb{Z}$ we have $m - m = 0$, and zero is divisible by 12. Hence $m\,R\,m$.

Relation R is also symmetric. Indeed, if $m\,R\,n$, i.e., $m - n = 12k$ for some k, then also $n - m$ is divisible by 12 ($n - m = -12k$). Therefore $n\,R\,m$.

Moreover, R is transitive: Take any numbers $m, n, p \in \mathbb{Z}$ such that $m\,R\,n$ and $n\,R\,p$. This means $m - n = 12k$ and $n - p = 12l$ for some k and l. Then $m - p = (m - n) + (n - p) = 12k + 12l = 12(k + l)$. Hence we have $m\,R\,p$.

3.3.9 Equivalence Classes. Every equivalence relation R on A "divides" A into the sets of equivalent elements. These sets are called equivalence classes. We will see the importance of equivalence classes later when we introduce so called residue classes.

Definition. Let R be an equivalence relation on a set A. An *equivalence class* of R corresponding to the element $a \in A$ is the set $R\,[a] = \{b \in A \mid a\,R\,b\}$. □

Example: Given the equivalence relation from 3.3.8. There are 12 different equivalence classes of R; namely

$$R\,[i] = \{j \mid j = i + 12k, k \in \mathbb{Z}\}, \quad \text{for } i = 0, \ldots, 11.$$

Definition. Let R be an equivalence relation on A. The set

$$\{R[a] \mid a \in A\}$$

is called the *quotient set* and denoted by A/R.

3.3.10 Properties of the Set of Equivalence Classes. The next proposition gives properties that sets of equivalence classes have.

Proposition. Let R be an equivalence relation on a set A. The set $\{\, R\,[a] \mid a \in A\}$ has the following properties:

1. Every element $a \in A$ belongs to $R\,[a]$ and hence $\bigcup\{\, R\,[a] \mid a \in A\} = A$.
2. Equivalence classes $R\,[a]$ are pairwise disjoint. That is, if $R\,[a] \cap R\,[b] \neq \emptyset$, then $R\,[a] = R\,[b]$.

□

Justification. Since every element $a \in A$ is related to itself (reflexivity), we get $a \in R\,[a]$. Thus $A \subseteq \bigcup\{\, R\,[a] \mid a \in A\}$. Moreover, each equivalence class is a subset of A, and therefore $\bigcup\{\, R\,[a] \mid a \in A\} \subseteq A$. We have shown the first property.

Let us verify the second property. Assume that there are two classes with non-empty intersection. We will show that they coincide. Take an element $z \in R\,[a] \cap R\,[b]$. Then $a\,R\,z$ and $b\,R\,z$. Since R is symmetric, we have $z\,R\,b$. Furthermore, since $a\,R\,z$ and $z\,R\,b$, it follows from transitivity of R that $a\,R\,b$. We have shown: If two classes $R\,[a]$, $R\,[b]$ have non-empty intersection, then the elements a and b are equivalent. Now, take any element $c \in R\,[a]$. Then $c\,R\,a$. From transitivity of R and from $a\,R\,b$ we get that $c\,R\,b$. Hence $c \in R\,[b]$. Analogously one can show that every element $c \in R\,[b]$ also belongs to $R\,[a]$. Therefore $R\,[a] = R\,[b]$.

3.3.11 Partition. The properties stated in the above proposition characterize another mathematics notion – a partition of a set. Here is the formal definition.

Definition. Let A be a non-empty set. A set $\mathcal{S}$ of non-empty subsets of A is called a *partition* of set A provided the following conditions hold:

1. Every element $a \in A$ belongs to some member of $\mathcal{S}$, i.e. $\bigcup \mathcal{S} = A$.

2. Elements of the set $\mathcal{S}$ are pairwise disjoint. In other words, if $X \cap Y \neq \emptyset$ then $X = Y$ for all $X, Y \in \mathcal{S}$.

□

In the above proposition we have shown that the quotient set modulo an equivalence relation forms a partition of the underlying set. On the other hand, we can associate an equivalence relation to any partition.

3.3.12 We already know that for every equivalence relation its set of equivalence classes forms a partition of the underlying set. On the other hand, every partition creates an equivalence relation. This is what the next proposition states.

Proposition. Let $\mathcal{S}$ be a partition of a set A. Then the relation $R_{\mathcal{S}}$ defined by:

$$a\,R_{\mathcal{S}} b \quad \text{if and only if} \quad a, b \in X \text{ for some } X \in \mathcal{S}$$

is an equivalence relation on set A. □

Justification. We need to show that relation $R_{\mathcal{S}}$ is reflexive, symmetric, and transitive. Since every element $a \in A$ belongs to some set $X \in \mathcal{S}$, we have $a\,R_{\mathcal{S}} a$, and relation $R_{\mathcal{S}}$ is reflexive. The symmetry of $R_{\mathcal{S}}$ is clear. Whenever we have $a\,R_{\mathcal{S}} b$ then also $b\,R_{\mathcal{S}} a$, and relation $R_{\mathcal{S}}$ is symmetric.

We show the transitivity: Let $a\,R_{\mathcal{S}} b$ and $b\,R_{\mathcal{S}} c$, i.e., let $a, b \in X$, $b, c \in Y$ for some $X, Y \in \mathcal{S}$. This means that $b \in X \cap Y$. Therefore the sets X and Y have a non-empty intersection. Hence they coincide. Thus we get $a, c \in X$, and $a\,R_{\mathcal{S}} c$ follows.

3.3.13 Remark. Notice that if we start with an equivalence relation R, then we form a corresponding partition into classes of R, and finally we make the equivalence relation that corresponds to the partition (according to the above proposition), we get precisely the equivalence relation R, with which we have started. Similarly, if we start with a partition, then form its corresponding equivalence relation, and finish with the partition into classes of the equivalence relation, we get the original partition.

3.3.14 Partial Order, a Poset. Apart from equivalence relations there is another type of relations that plays a special role in mathematics. And it is a so called partial order, or a partial ordered set, shortly a poset.

Definition. A relation R on a set A is called an *order (partial order)*, if it is reflexive, antisymmetric and transitive. A set A together with a partial order is often called a *poset.* □

3.3.15 Examples of Posets.

1. The well-known ordering of real numbers is a partial order in the above sense. Hence, $(\mathbb{R}, \leq)$ is a poset. Indeed, for all real numbers $a, b, c \in \mathbb{R}$ we have: $a \leq a$; if $a \leq b$ and $b \leq a$ then necessarily $a = b$; if $a \leq b$ and $b \leq c$ then also $a \leq c$.

2. Denote by A the set of all subsets of the set U. Then the relation $\subseteq$, "to be a subset", is a partial order on A. Hence, $(P(U), \subseteq)$ is a poset. Verification of reflexivity, antisymmetry and transitivity is left to the reader.

3. Let $A = \mathbb{N}$, where $\mathbb{N}$ is the set of all natural numbers. The the relation of divisibility defined by $m \,|\, n$ if and only if m is a divisor of n (i.e. if $n = k \cdot m$ for some $k \in \mathbb{N}$) is a partial order. Hence $(\mathbb{N}, |)$ is a poset. Indeed, for all natural numbers m, n, k we have $m \,|\, m$; if $m \,|\, n$ and $n \,|\, m$ then $m = n$; if $m \,|\, n$ and $n \,|\, k$ then also $m \,|\, k$.

Chapter 4

Integers

4.1 Integers and Their Properties

Integers are well known numbers. They play a crucial role in mathematics, primarily in the discrete mathematics and its applications. We will use them in the sequel to introduce ”new numbers”, the residual classes of integers modulo a positive integer n.

First, let us recall some well known facts about division of integers. They are: integer division with remainder, a common divisor, and the greatest common divisor. We present the Euclid’s Algorithm for finding the greatest common divisor and its applications, namely for solving Diophantic equations — equations in which only integer solutions are sought.

4.1.1 The Division Theorem. Let a, b, $b > 0$, be two integers. Then there exist unique integers q, r such that

$$a = q\,b + r, \quad 0 \le r < b.$$

□

We will prove later only the uniqueness part of the theorem, the existence of q and r follows from the well known way how to divide two integers.

4.1.2 Remark. 1. The number q is called the *quotient*, and r the *remainder* when we divide a by b.

2. We formulated the division theorem 4.1.1 not only for natural numbers a and b, but also for a negative integer a. In that case, we have to be a little more careful. Assume that a is negative. Divide the absolute value $|a|$ by b. Then $|a| = q'b + r'$ for $0 \le r' < b$, $q' \le 0$, and $a = -q'b - r'$. If $r' = 0$ then $a = -q'b$, and we have $q = -q'$, $r = 0$. Assume $0 < r' < b$, then $a = -q'b - r' = -(q'+1)b + (b - r')$. Moreover, $0 < b - r' < b$, and hence $q = -(q'+1)$ and $r = b - r'$.

We show the procedure on the following example: Let $a = -7$, $b = 3$. We have $7 = 2\cdot3+1$, hence $-7 = -2\cdot3 - 1 = -3\cdot3 + (3-1)$. Therefore, $q = -3$ and $r = 2$.

Let us prove the uniqueness of the quotient and the remainder.

4.1.3 Justification of Uniqueness. Assume that there exist two pairs q and r from 4.1.1, say q_1, r_1 and q_2, r_2, where $0 \le r_1, r_2 < b$. We have

$$a = q_1\,b + r_1, \text{ and } a = q_2\,b + r_2.$$

Then

$$q_1\,b + r_1 = q_2\,b + r_2, \quad \text{i.e.} \quad (q_1 - q_2)\,b = r_2 - r_1.$$

Because $|r_2 - r_1| < b$ and it is a multiple of b, the number $q_1 - q_2$ must be 0 (indeed, otherwise $|(q_1 - q_2)b| \ge b$) . And this means that $q_1 = q_2$ and $r_1 = r_2$. We have shown that the quotient and the remainder are unique. □

4.1.4 Divisibility.

Let us recall other well known notions.

Definition. Given two integers a, b. We say that b *divides* a if $a = k\,b$ for some integer k. We also say that a is a *multiple* of b. This fact is denoted by $b \,|\, a$.

A positive integer p, $p > 1$, is said to be a *prime* if it satisfies:

$$a \,|\, p,\ a \geq 0, \quad \text{implies} \quad a = 1 \ \text{ or } a = p.$$

A number $n > 1$ is *composite* if it is not a prime, or equivalently, if there exist $r, s \in \mathbb{Z}$ such that $n = r \cdot s$ and $r > 1$ and $s > 1$. □

Notice, that 0 divides 0; indeed, e.g. $0 = 1 \cdot 0$. If $b \neq 0$ then $b \,|\, a$ if and only if the remainder when dividing a by b equals 0. Also, note that 1 has a special role, it is (by definition) neither a composite number nor a prime.

4.1.5 A Common Divisor and the Greatest Common Divisor.

Let us recall the definition of a common divisor and the greatest common divisor.

Definition. Let a and b be two integers. A *common divisor* of a and b is any integer e for which $e \,|\, a$ and $e \,|\, b$.

The *greatest common divisor* of a, b is the integer c such that

1. $c \geq 0$
2. c is a common divisor of a and b, i.e. $c \,|\, a$ and $c \,|\, b$,
3. and if e is any common divisor of a and b then $e \,|\, c$.

The greatest common divisor of a and b is denoted by $\gcd(a, b)$. Integers a and b are called *relatively prime* (or *coprime*) if $\gcd(a, b) = 1$. □

4.1.6 Remarks.

1. For every natural number a we have $a = \gcd(a, 0)$.
2. If for natural numbers a, b we have $a \,|\, b$ then $\gcd(a, b) = a$.
3. For every integers a, b it holds that $\gcd(a, b)$ is always non-negative and $\gcd(a, b) = \gcd(-a, b) = \gcd(a, -b) = \gcd(-a, -b)$.

4.1.7 You know from school mathematics that the greatest common divisor of a and b can be found using a factorization of a and b into products of primes. Unfortunately, finding such factorization for big a (or b) is a very difficult task. (There is not known a tractable algorithm for finding it.) The following fast algorithm, due to Euclid, is based on the division theorem.

4.1.8 Euclid's Algorithm.

Input: Positive natural numbers a and b
Output: $c = \gcd(a, b)$.

```
1. (Initialization.)
   $u := a$, $t := b$;

2. (Divide $u$ by $t$.)
   repeat
          do $u = q \cdot t + r$;
                 $u := t$, $t := r$.
   until $t = 0$.
3. (The greatest common divisor)
   return $c := u$.
```

4.1.9 Correctness of the Euclid's Algorithm. Notice that the above algorithm will always terminate; indeed, the number t in the next execution of the step 2 is an integer that is always strictly smaller than the previous one. So after a finite number of executions of step 2, we get $t = 0$ and the algorithm terminates.

The fact that the algorithm returns $\gcd(a, b)$ is proved in the following proposition.

Proposition. The pairs of numbers u, t and t, r from the Euclid's algorithm 4.1.8 have the same common divisors. Hence $\gcd(u, t) = \gcd(t, r) = \gcd(a, b)$. □

Justification. Since $r = u - q \cdot t$ for an integer q, any common divisor of u and t is also a divisor of t, r. Indeed, if $u = d \cdot u'$ and $t = d \cdot t'$, then also $r = d \cdot u' - q \cdot d \cdot t' = d(u' - qt')$.

On the other hand, $u = q \cdot t + r$ so any common divisor of t, r is a divisor of u as well. Indeed, if $t = d \cdot t'$ and $r = d \cdot r'$, then also $u = q \cdot d \cdot t' + d \cdot r' = d(qt' + r')$. □

4.1.10 Euclid's Algorithm can be extended in such a way that it finds not only $\gcd(a, b)$ but also **integers** x, y that solve the following equation

$$a\,x + b\,y = \gcd(a, b).$$

Such equations (considered as equations over integers) will play a crucial role when investigating properties of residual classes modulo n.

4.1.11 Bezout's Theorem. Let a and b be two natural numbers. Denote $c = \gcd(a, b)$. Then there exist integers x, y such that

$$a\,x + b\,y = c.$$

□

The proof of the Bezout's theorem will be given by the extended Euclid's algorithm, because the extended Euclid's algorithm not only proves the existence of integers x and y, but it finds them together with the greatest common divisor of a and b.

4.1.12 Extended Euclid's Algorithm.

Input: natural numbers a and b.

Output: $c = \gcd(a, b)$ together with $x, y \in \mathbb{Z}$ for which $a\,x + b\,y = c$.

```
1. (Initialization.)
     $u := a$, $x_u := 1$, $y_u := 0$, $t := b$, $x_t := 0$, $y_t := 1$;
2. (Division.)
     repeat
            do $u = q \cdot t + r$, $x_r := x_u - q\,x_t$, $y_r := y_u - q\,y_t$;
                  $u := t$, $x_u := x_t$, $y_u := y_t$
                  $t := r$, $x_t := x_r$, $y_t := y_r$.
     until $t = 0$
3. (Greatest common divisor and $x$, $y$)
     return $c := u$, $x := x_u$, $y := y_u$.
```

Justification of the above algorithm is similar to 4.1.8.

1. $a = 1 \cdot a + 0 \cdot b$ and $b = 0 \cdot a + 1 \cdot b$. So, the step 1 correctly sets x_u, y_u and x_t, y_t.
2. Assume that $u = a\,x_u + b\,y_u$ and $t = a\,x_t + b\,y_t$. Then

$$r = u - q\,t = a\,x_u + b\,y_u - q\,(a\,x_t + b\,y_t) = a\,(x_u - q\,x_t) + b\,(y_u - q\,y_t).$$

Hence, it is clear that the numbers x_r and y_r are correctly defined.

□

The Bezout's theorem has couple of important corollaries; some of them you have used in school mathematics without justification.

4.1.13 Corollary.

1. Let a and b be two relatively prime numbers. If a divides a product $b \cdot c$ then a divides c.
2. If a prime number p divides a product $a \cdot b$ then it divides at least one of the numbers a, b.

□

Justification. We prove the first part of the corollary; the second one is an easy consequence of the first one.

Assume that numbers a and b are relatively prime. By the Bezout's theorem there exist integers x, y such that

$$1 = a\,x + b\,y.$$

Multiplying the equation by c we get

$$c = a\,c\,x + b\,c\,y.$$

Number a divides $a\,c$ and it also divides the product $b\,c$, hence a divides c. □

4.1.14 Prime Factorization. Let us recall another known fact – a factorization of a natural number different from 1 into a product of primes.

Theorem. Every natural number n, $n > 1$, factors into a product of primes, i.e.

$$n = p_1^{i_1} \cdot p_2^{i_2} \cdot \ldots \cdot p_k^{i_k},$$

where $p_1, \ldots, p_k$ are distinct primes, and $i_1, \ldots, i_k$ positive natural numbers.

If moreover $p_1 < p_2 < \ldots < p_k$ then the factorization is unique. □

Justification. The existence of a prime factorization is shown using mathematical induction (more precisely, the principle of strong mathematical induction).

To justify the uniqueness one can use the above corollary. Assume that

$$p_1^{i_1} \cdot p_2^{i_2} \cdot \ldots \cdot p_k^{i_k} = q_1^{j_1} \cdot q_2^{j_2} \cdot \ldots \cdot q_m^{j_m}$$

and $p_1 < p_2 < \ldots < p_k$, $q_1 < q_2 < \ldots < q_m$ then p_1 divides $q_1^{j_1} \cdot q_2^{j_2} \cdot \ldots \cdot q_m^{j_m}$ so $p_1 = q_1$. (Indeed, a prime number p divides a prime number q then $p = q$. Hence, p_1 must be equal to the smallest prime among q_j and it is q_1.)

If we divide the equality by p_1 and repeat the argument we get that $i_1 = j_1$. Analogously (after dividing by $p_1^{i_1}$) we get $p_2 = q_2$, $i_2 = j_2$, etc. $k = m$ and $p_k = q_k$, $i_k = j_k$. □

4.1.15 There is a Countably Many Primes. Using the prime factorization theorem one can easily prove that there is an infinite number of primes – see the following theorem. Since every prime is an integer, it means that there is countably many of them.

Theorem. There are infinitely (countably) many primes. □

Justification. Assume that there were only finitely many primes, say $p_1, p_2, \ldots, p_N$ were the only primes. Then the number $n = p_1 \cdot p_2 \cdot \ldots \cdot p_N + 1$ is a product of primes; namely is divisible by some prime p. But p cannot be among $p_1, \ldots, p_N$, since n is not divisible by any p_i – a contradiction. □

4.1.16 Diophantic Equations. The Bezout's theorem 4.1.11 helps us to solve other linear equations where we are looking for integer solutions – so called *Diophantic equations*.

Definition. Given three integers a, b, c. Find all integers $x, y \in \mathbb{Z}$ which are solutions of the following equation

$$ax + by = c. \tag{4.1}$$

□

4.1.17 When a Diophantic Equation Has Got a Solution. The following proposition characterizes all Diophantic equations that have got at least one solution.

Proposition. Equation 4.1 has got at least one solution if and only if c is divisible by the greatest common divisor of a and b. □

Justification. Denote $d = \gcd(a, b)$. If c is a multiple of d, say $c = k\,d$, then it suffices to find integers x', y' from the Bezout's Theorem for which

$$d = a\,x' + b\,y' \quad \text{and} \quad c = k\,d = a\,k\,x' + b\,k\,y'.$$

Now $x := k\,x'$ and $y := k\,y'$ is one solution of the equation 4.1.

Assume that there exist integers x, y such that

$$c = a\,x + b\,y.$$

Then every common divisor of a, b divides c as well. Hence the greatest common divisor is one of them and $\gcd(a, b)$ divides c. □

4.1.18 Homogeneous Diophantic Equations. A Diophantic equation is said to be *homogeneous* if the right hand side is 0, i.e. $c = 0$ in 4.1. A homogeneous Diophantic equation always has got countably many solutions, see the following proposition.

Proposition. If $a \neq 0 \neq b$ then the equation $ax + by = 0$ has always got infinitely many solutions, more precisely, $x = -k \cdot b_1$, $y = k \cdot a_1$ for any $k \in \mathbb{Z}$, where $a_1 = \frac{a}{\gcd(a,b)}$ and $b_1 = \frac{b}{\gcd(a,b)}$ are all integer solutions of it. □

Justification. Divide the equation $ax + by = 0$ by $\gcd(a, b)$. We get $a_1 x + b_1 y = 0$ for $a_1 = \frac{a}{\gcd(a,b)}$ and $b_1 = \frac{b}{\gcd(a,b)}$. Moreover, a_1 and b_1 are relatively prime.

From $a_1 x = -b_1 y$ we get that a_1 divides y, see the corollary 4.1.13. Hence there is $k \in \mathbb{Z}$ for which $y = ka_1$. Substituting it to the equation we get

$$a_1 x = -b_1(ka_1) \quad \text{and} \quad x = -kb_1.$$

□

4.1.19 Equations 4.1 are linear equations with two variables. Analogously as in linear algebra it is easy to see that a general solution of $ax + by = c$ is a sum of **one** solution of $ax + by = c$ plus a general solution of the corresponding homogeneous equation $ax + by = 0$. Hence, we get the following proposition.

Proposition. If c is a multiple of $\gcd(a, b)$ then any solution of 4.1 is of the form

$$x = x_0 + k \cdot b_1, \;\; y = y_0 - k \cdot a_1,$$

where x_0, y_0 is a solution of the equation 4.1, $a_1 = \frac{a}{\gcd(a,b)}$, $b_1 = \frac{b}{\gcd(a,b)}$ and $k \in \mathbb{Z}$. □

4.1.20 A Procedure How to Solve Diophantic Equations. We can summarize the above propositions to the following instructions how to solve equations 4.1.

1. Using the extended Euclid's algorithm we find integers x_0 and y_0 satisfying 4.1 or find out that the equation does not have a solution.

2. If there is at least one integer solution of 4.1 we find a general integer solution of the equation $a\,x + b\,y = 0$ as follows.

   First, we divide the equation by $\gcd(a, b)$ and obtain an equation $a_1\,x + b_1\,y = 0$ where a_1 and b_1 are relatively prime. The general solution is now $x = b_1\,k$, $y = -a_1\,k$ where $k \in \mathbb{Z}$.

3. The general solution of 4.1 is

$$x = x_0 + b_1\,k, \;\; y = y_0 - a_1\,k, \quad k \in \mathbb{Z}.$$

The correctness of the above method follows from the proposition above.

4.2 Congruence relation modulo n

We have introduced the extended Euclid's algorithm which helped us to solve Diophantic equations; linear equations with two unknowns where we look only for integer solutions. The material of the last lecture will be used for introduction new "numbers", residue classes, and operations with them.

4.2.1 The Relation Modulo n. First of all we introduce an equivalence relation *modulo n* for a natural number $n > 1$. You have already come across it; indeed, consider the imaginary unit i. We have

$$i^2 = -1, \ \ i^3 = -i, \ \ i^4 = 1, \ \text{ and } \ i^5 = i.$$

Therefore, it is easy to calculate powers of the imaginary unit i; indeed for example $i^{651} = i^{4\cdot 162+3} = 1^{162} \cdot i^3 = -i$. More generally, to calculate i^k it suffices to know the remainder r when k is divided by 4, and then we have $i^k = i^r$.

Definition. Given two integers a, b and a natural number $n > 1$. We say that *a is congruent to b modulo n* and write $a \equiv b \pmod{n}$ if $a - b$ is divisible by n. □

4.2.2 Equivalent Characterizations of Modulo n. We could introduce the relation modulo n in other two ways.

Proposition. Let a and b be two integers. Then the following is equivalent:

1. $a \equiv b \pmod{n}$,
2. $a = b + k\,n$ for some integer k,
3. a and b have the same remainders when divided by n.

□

Justification. It is clear that conditions 1. and 2. are equivalent; indeed the fact that $a - b$ is divisible by n means that $a - b = k\,n$ for some integer $k \in \mathbb{Z}$; and this is the same as $a = b + k\,n$.

We show that $a \equiv b \pmod{n}$ if and only if a and b have the same remainder when divided by n. Assume that $a = q_1\,n + r_1$, $b = q_2\,n + r_2$ and $0 \le r_1, r_2 < n$.

If $r_1 = r_2$, then $a - b = (q_1 - q_2)\,n$ and $a \equiv b \pmod{n}$ holds.

If $r_1 \neq r_2$, then from the uniqueness of the division theorem, $a - b$ is not divisible by n, so $a \equiv b \pmod{n}$ does not hold. □

4.2.3 The Relation Modulo n is an Equivalence Relation on $\mathbb{Z}$.

Proposition. Let a, b, and c be integers. Then

1. $a \equiv a \pmod{n}$ (modulo n is reflexive);
2. if $a \equiv b \pmod{n}$, then also $b \equiv a \pmod{n}$ (modulo n is symmetric);
3. if $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$, then $a \equiv c \pmod{n}$ (modulo n is transitive).

□

The justification is easy, especially if we use 4.2.2.

4.2.4 Properties of the Equivalence Modulo n. The equivalence modulo n also "maintain" operations addition and multiplication of integers. More precisely:

Proposition. Assume that for integers a, b, c, and d it holds that $a \equiv b \pmod{n}$ and $c \equiv d \pmod{n}$. Then

$$(a + c) \equiv (b + d) \pmod{n} \quad \text{a} \quad (a \cdot c) \equiv (b \cdot d) \pmod{n}.$$

□

Justification. Assume that $a \equiv b \pmod{n}$ and $c \equiv d \pmod{n}$. Then $a = b + kn$ and $c = d + rn$ for some $k, r \in \mathbb{Z}$. Therefore, $a + c = b + d + (k + r)n$ and $a \cdot c = (b + kn)(d + rn) = bd + (br + dk + krn)n$. And this is equivalent to $(a + c) \equiv (b + d) \pmod{n}$ and $(a \cdot c) \equiv (b \cdot d) \pmod{n}$. □

4.2.5 The 4.2.4 has two special cases. We state them as corollaries.

Corollary. Given two integers a, b such that $a \equiv b \pmod{n}$. Then

1. $ra \equiv rb \pmod{n}$ for every integer r;
2. $a^k \equiv b^k \pmod{n}$ for every natural number k.
3. Moreover, if $a_i \equiv b_i \pmod{n}$ for every $i = 0, \dots, k$, a $r_0, \dots, r_k$ are arbitrary integers, then
$$(r_0\, a_0 + \ldots + r_k\, a_k) \equiv (r_0\, b_0 + \ldots + r_k\, b_k) \pmod{n}.$$

□

Justification. 1. To prove the first part it suffices to use the above proposition 4.2.4 for the pair $r \equiv r \pmod{n}$ a $a \equiv b \pmod{n}$.

2. From the above proposition we know that $a^2 \equiv b^2 \pmod{n}$ (we have used $a \equiv b \pmod{n}$ and $a \equiv b \pmod{n}$). Now, from $a \equiv b \pmod{n}$ and $a^2 \equiv b^2 \pmod{n}$ we get $a^3 \equiv b^3 \pmod{n}$, $a^4 \equiv b^4 \pmod{n}$, etc.

To make the argument more accurate we can use mathematical induction over k. □

4.2.6 We can ask whether the first part of the corollary is still valid if we reverse the implication. More precisely, if from $ra \equiv rb \bmod n$ it follows that $a \equiv b \bmod n$. A simple example shows that this is not the case. Indeed, we have $6 \equiv 10 \bmod 4$, but $3 \not\equiv 5 \bmod 4$. The following proposition states what can be deduced form $r\,a \equiv r\,b \bmod n$.

Proposition. Let r, a, b be integers and n a natural number $n > 1$ such that $ra \equiv rb \pmod{n}$. Then
$$a \equiv b \left(\bmod \frac{n}{\gcd(n, r)}\right). \tag{4.2}$$

□

Justification. We know that $ra - rb = kn$ for an integer $k \in \mathbb{Z}$. Hence $r(a - b) = kn$. Denote $d = \gcd(r, n)$. Then $r = s \cdot d$, $n = m \cdot d$, and the integers s and m are relatively prime. Substituting into $r(a - b) = kn$ and get
$$s\,d\,(a - b) = k\,m\,d, \quad \text{and} \quad s\,(a - b) = k\,m.$$

Since the numbers s and m are relatively prime, and s divides the product $k\,m$, the number s must divide k. Therefore, $s\,(a - b) = s\,j\,m$ and $a - b = j\,m$. We have shown that $a \equiv b \pmod{m}$, in other words $a \equiv b$ $(\bmod \frac{n}{\gcd(n,r)})$. □

4.2.7 Solving $(a + x) \equiv b$ mod n. Given integers a, b and a natural number $n > 1$. Find all integers x for which
$$(a + x) \equiv b \pmod{n}. \tag{4.3}$$

This problem has got always a solution which is any $x \in \mathbb{Z}$ for which $x \equiv (b - a) \pmod{n}$.

4.2.8 Solving $(a \cdot x) \equiv b$ mod n. Given two integers a, b and a natural number $n > 1$. Find all integers x for which
$$a\,x \equiv b \pmod{n}. \tag{4.4}$$

Such x does not always exist. For example, there is no integer x for which $2x \equiv 3 \pmod{4}$. We will use Diophantic equations and their solutions to find a necessary and sufficient condition on a, b, and n for which $x \in Z$ satisfying the relation 4.4 exists.

4.2.9 Proposition. Equation 4.4 has got a solution if and only if the number b is a multiple of $\gcd(a,n)$.

In this case all integers x satisfying 4.4 are solutions of the following Diophantic equation

$$a\,x + n\,y = b.$$

□

Justification. We know that $a\,x \equiv b \pmod{n}$ means $a\,x - b = k\,n$ for an integer $k \in \mathbb{Z}$, and this is equivalent to $a\,x - k\,n = b$. If we substitute $y := -k$, we get the Diophantic equation 4.2 which has a solution if and only if b is divisible by $\gcd(a,n)$. □

4.2.10 Let us mention another property that the equivalences modulo have got.

Proposition. Let $n > 1$, $m > 1$ be two relatively prime natural number. And let for some $a, b \in \mathbb{Z}$ it holds that $a \equiv b \pmod{n}$ and $a \equiv b \pmod{m}$

Then also $a \equiv b \pmod{nm}$. □

Justification. We know that $a - b = kn$ and $a - b = jm$ for some $k, j \in \mathbb{Z}$. Hence $kn = jm$. Since n and m are relatively prime and n divides the product jm, we know that n divides j. So $a - b = jm = r\,nm$ for some $r \in \mathbb{Z}$. We have shown that $a \equiv b \pmod{nm}$. □

4.2.11 Remark. A stronger proposition can be proved than 4.2.10, namely: Assume that $a \equiv b \pmod{n}$ and $a \equiv b \pmod{m}$. Let $n_1 = \frac{n}{\gcd(n,m)}$ and $m_1 = \frac{m}{\gcd(n,m)}$. Then

$$a \equiv b \pmod{n_1 m_1}.$$

The justification is analogous to 4.2.10; indeed, the equation $kn = jm$ must be first divided by $\gcd(n,m)$.

4.2.12 Small Fermat Theorem. We will end this part concerning the equivalence modulo n by the small Fermat theorem which is a basis of the RSA public-key cryptosystem. (In literature, the Small Fermat Theorem is sometimes called Fermat Little Theorem.)

Theorem. Let p be a prime and a an integer relatively prime to p. Then

$$a^{p-1} \equiv 1 \pmod{p}.$$

□

Justification. One of the proofs of the small Fermat theorem uses basic properties of groups and we will give it later. There is also a proof which uses only elementary mathematics. In fact, we will first show that for every integer a it holds that $a^p \equiv a \bmod p$ by mathematical induction on a

1. Basic step: Let $a = 0$ or $a = 1$. Then $a^p = a$, hence $a^p \equiv a \pmod{p}$.

2. Induction step: Assume that $a^p \equiv a \pmod{p}$, and calculate $(a+1)^p - (a+1)$. By the binomial theorem we have

$$(a+1)^p - (a+1) = a^p + \binom{p}{1}a^{p-1} + \binom{p}{2}a^{p-2} + \ldots + \binom{p}{p-1}a + 1 - (a+1) =$$

$$= a^p - a + \binom{p}{1}a^{p-1} + \ldots + \binom{p}{p-1}a.$$

We know by the induction hypothesis that $a^p - a$ is divisible by p. Hence, if we show that $\binom{p}{i}$ is divisible by p for every i, $0 < i < p$, we will know that so is $(a+1)^p - (a+1)$. And this means that $(a+1)^p \equiv (a+1) \bmod p$.

We know that

$$\binom{p}{i} = \frac{p!}{i!\,(p-i)!}.$$

Hence

$$i!\,(p-i)!\binom{p}{i}=p!.$$

Moreover, p divides neither $i!$ $(i<p)$ nor $(p-i)!$ $(0<i)$. Hence, p must divide $\binom{p}{i}$.

Now, assume that a and p are relatively prime. We have $a^p-a=k\,p$ for some $k\in\mathbb{Z}$. Thus $a\,(a^{p-1}-1)=k\,p$. Since a and p are relatively prime, a divides k and $a^{p-1}-1=j\,p$ which proves that $a^{p-1}\equiv 1\pmod p$. □

4.3 Residue Classes Modulo n

We know that the relation modulo n is an equivalence relation on the set $\mathbb{Z}$, see 4.2.3. An equivalence class of the equivalence modulo n containing a number $i\in\mathbb{Z}$ is called the *residue class containing* i and is denoted by $[i]_n$. We know that

$$[i]_n=\{j\,|\,j=i+kn \text{ for some } k\in\mathbb{Z}\}. \tag{4.5}$$

The name residue classes comes from the fact that an integer j belongs to $[i]_n$ if and only if i and j have the same remainders when divided by n.

4.3.1 The Set $\mathbb{Z}_n$. There are n distinct residue classes modulo n; indeed, they are the residue classes corresponding to the numbers (remainders) $0,1,\ldots,n-1$. The set of all residue classes is denoted by $\mathbb{Z}_n$, so

$$\mathbb{Z}_n=\{[0]_n,[1]_n,\ldots,[n-1]_n\}.$$

4.3.2 Calculations in $\mathbb{Z}_n$. It is clear from the proposition 4.2.4 that the equivalence modulo n is compatible with operations $+$ and $\cdot$. Indeed, if $i\equiv j\pmod n$ and $k\equiv l\pmod n$ then $i+k\equiv j+l\pmod n$ and $i\cdot k\equiv j\cdot l\pmod n$. These properties can be reformulate as follows:

If we choose any $a\in[i]_n$ and any $b\in[j]_n$ then the number $a+b$ belongs to $[i+j]_n$, and the number $a\cdot b$ belongs to $[i\cdot j]_n$. This allows us to define operations addition $\oplus$ and multiplication $\odot$ on the set $\mathbb{Z}_n$ as follows:

$$[i]_n\oplus[j]_n=[i+j]_n,\qquad [i]_n\odot[j]_n=[i\cdot j]_n. \tag{4.6}$$

4.3.3 Properties of the Operation $\oplus$.

- $\oplus$ is associative, i.e. for any three integers i,j,k we have:

$$([i]_n\oplus[j]_n)\oplus[k]_n=[i]_n\oplus([j]_n\oplus[k]_n).$$

- $\oplus$ is commutative, i.e. for any two integers i,j we have:

$$[i]_n\oplus[j]_n=[j]_n\oplus[i]_n.$$

- The class $[0]_n$ plays the role of “zero”, more precisely, for any integer i we have:

$$[0]_n\oplus[i]_n=[i]_n.$$

- We can ”subtract”, more precisely for any integer $[i]_n$ there exists class $-[i]_n$ such that

$$[i]_n\oplus(-[i]_n)=[0]_n.$$

□

Justification. Verification of the above properties is straightforward and it is left to the reader.

4.3.4 Properties of the Operation $\odot$.

- $\odot$ is associative, i.e for any three integers i, j, k we have:
$$([i]_n \odot [j]_n) \odot [k]_n = [i]_n \odot ([j]_n \odot [k]_n).$$
- $\odot$ is commutative, i.e. for any two integers i, j we have:
$$[i]_n \odot [j]_n = [j]_n \odot [i]_n.$$
- The class $[1]_n$ plays the role of "identity", More precisely, for any integer i we have:
$$[1]_n \odot [i]_n = [i]_n.$$

□

Justification. Verification of the above properties is straightforward and it is left to the reader.

4.3.5 Remark. In the above properties there is no one which means something as "cancellation" or "division" for $\odot$. More precisely, we have not stated any general condition under which for a given integer i there exists an integer j such that $[i]_n \odot [j]_n = [1]_n$. The reason is that no every equation of the form $[i]_n \odot [x]_n = [1]_n$ has a solution. The following proposition characterizes i and n for which such x exists.

4.3.6 Properties of the Operation $\odot$.

- $\odot$ is associative, i.e for any three integers i, j, k we have:
$$([i]_n \odot [j]_n) \odot [k]_n = [i]_n \odot ([j]_n \odot [k]_n).$$
- $\odot$ is commutative, i.e. for any two integers i, j we have:
$$[i]_n \odot [j]_n = [j]_n \odot [i]_n.$$
- The class $[1]_n$ plays the role of "identity", More precisely, for any integer i we have:
$$[1]_n \odot [i]_n = [i]_n.$$

□

Justification. $[i]_n \odot [x]_n = [j]_n$ can be rewritten as $[i \cdot x]_n = [j]_n$ and hence
$$i \cdot x \equiv j \bmod n.$$
And this leads to
$$i\,x - j = k\,n, \text{ so we have } i\,x - k\,n = j.$$
And the last equation is in fact a Diophantic equation $i\,x + n\,y = j$ which has a solution if and only if j is a multiple of $\gcd(i, n)$.

From **??** we know that all integers $x \in \mathbb{Z}$ satisfying $i\,x + n\,y = j$ are of the form $x_0 + k\,n_1$ where x_0 is one solution of the non-homogeneous equation, and $i_1 = \frac{i}{d}$ and $n_1 = \frac{n}{d}$. It can be shown that for $x_k = x_0 + k\,n_1$ it holds that $[x_k]_n$ are distinct elements of $\mathbb{Z}_n$ for which **??** holds. □

4.3.7 A special case of **??** is the following:

Corollary. For a residue class $[i]_n$ there is a residue class $[x]_n$ such that
$$[i]_n \odot [x]_n = [1]_n \tag{4.7}$$
if and only if the numbers i and n are relatively prime. □

The class $[x]_n$ satisfying 4.7 is called the *inverse* of $[i]_n$ and we denote it $[i]_n^{-1}$.

4.3.8 Distributivity Law for $\oplus$ and $\odot$. For any three integers i, j, k it holds that

$$[i]_n \odot ([j]_n \oplus [k]_n) = ([i]_n \odot [j]_n) \oplus ([i]_n \odot [k]_n).$$

4.3.9 Remark. If p is a prime number then the set $\mathbb{Z}_p$ satisfies all the properties that addition and multiplication of real numbers have got.

If n is a composite number (not a prime) then the situation is different. For example if $n = r \cdot s$, $0 < r < n$ and $0 < s < n$, then $[r]_n \odot [s]_n = [0]_n$ even though the classes $[r]_n$ and $[s]_n$ are non-zero. (It means that we cannot "divide" by such elements.)

4.3.10 Convention. Later on, when there is not fear of misunderstanding we will write $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ instead of $\mathbb{Z}_n = \{[0]_n, [1]_n, \ldots, [n-1]_n\}$ and the operations $\oplus$, $\odot$ will be denoted by an "ordinary signs", i.e. simply by $+$ and $\cdot$.

Note that we can write that in $\mathbb{Z}_n$ for every $i, j \in \mathbb{Z}_n$

$$i + j = k, \ \text{ where } k \text{ is the remainder when } i + j \text{ is divided by } n;$$

$$i \cdot j = l, \ \text{ where } l \text{ is the remainder when } i\,j \text{ is divided by } n.$$

Chapter 5

Binary Operations

In the last lecture, we introduced the residue classes $\mathbb{Z}_n$ together with their addition and multiplication. We have also shown some properties that these two operations have. Today, we will study sets together with one operation in general and try to derive some properties that can be used regardless how an operation is defined and what elements a set has. We will define item-by-item groupids (the most general case), semigroups (groupoids that satisfy the associativity law), monoids (semigroups with a neutral element), and groups (monoids where every element is invertible).

5.1 Groupoids, Semigroups, Monoids

5.1.1 Groupoids. The most general notion of this section is the notion of a groupoid.

Definition. A *binary operation on a set* S is any mapping from the set of all pairs $S \times S$ into the set S.

A pair $(S, \circ)$ where S is a set and $\circ$ is a binary operation on S is called a *groupoid*. □

Note that the only condition for a binary operation on S is that **for every** pair of elements of S their result must be defined and must be an element in S.

A binary operation is usually denoted by $\cdot$, or $+$, $\circ$, $\star$ etc. (A binary operation $\circ$ assigns to elements x, y the element $x \circ y$.)

Examples of groupoids. The following are groupoids.

1) $(\mathbb{R}, +)$ where $+$ is addition on the set of all real numbers.
2) $(\mathbb{Z}, +)$ where $+$ is addition on the set of all integers.
3) $(\mathbb{N}, +)$ where $+$ is addition on the set of all natural numbers.
4) $(\mathbb{R}, \cdot)$ where $\cdot$ is multiplication on the set of all real numbers.
5) $(\mathbb{Z}, \cdot)$ where $\cdot$ is multiplication on the set of all integers.
6) $(M_n, \cdot)$ where M_n is the set of all square matrices of order n, and $\cdot$ is multiplication of matrices.
7) $(\mathbb{Z}_n, \oplus)$ for any $n > 1$.
8) $(\mathbb{Z}_n, \odot)$ for any $n > 1$.
9) $(\mathbb{Z}, -)$, where $-$ is subtraction on the set of all integers.

Examples which are not groupoids.

- $(\mathbb{N}, -)$ is not a groupoid because subtraction is not a binary operation on $\mathbb{N}$. Indeed, $3 - 4$ is not a natural number.
- $(\mathbb{Q}, :)$, where $:$ is the division, because $1 : 0$ is not defined.

5.1.2 Semigroups. General groupoids are structures where it is rather difficult to "calculate". Indeed, if we want to "multiply" four elements we must know in which order to do it. It means whether it is $a \circ ((b \circ c) \circ d)$, or $a \circ ((b \circ c) \circ d)$, or one of the other two possibilities. First, we will be interested in groupoids where we do not need to use brackets, these will be groupoids where the associative law holds.

Definition. Given a groupoid $(S, \circ)$. If for every $x, y, z \in S$ we have

$$x \circ (y \circ z) = (x \circ y) \circ z \tag{5.1}$$

$(S, \circ)$ is called a *semigroup.* □

The property 5.1 is called the *associative law.*

Examples of semigroups. The following groupoids are semigroups:

1) $(\mathbb{R}, +)$, $(\mathbb{Z}, +)$, $(\mathbb{N}, +)$.
2) $(\mathbb{R}, \cdot)$, $(\mathbb{Z}, \cdot)$, $(\mathbb{N}, \cdot)$.
3) $(\mathbb{Z}_n, \oplus)$, $(\mathbb{Z}_n, \odot)$.
4) $(M_n, +)$, $(M_n, \cdot)$, where M_n is the set of square real matrices of order n and $+$ and $\cdot$ is addition and multiplication, respectively, of matrices.
5) $(A, \circ)$ where A is the set of all mappings $f\colon X \to X$ for a set X, and $\circ$ is the composition of mappings.

Examples of groupoids which are not semigroups.

- $(\mathbb{Z}, -)$, i.e. the set of all integers with subtraction. Indeed, $2 - (3 - 4) = 3$ but $(2 - 3) - 4 = -5$.
- $(\mathbb{R} \setminus \{0\}, :)$, i.e. the set of nonzero real numbers together with the division :. Indeed, $4 : (2 : 4) = 8$, but $(4 : 2) : 4 = \frac{1}{2}$.

5.1.3 Neutral (Identity) Element. A groupoid $(S, \circ)$ may or may not have an element that "does not change" anything if it is used. The precise definition is given bellow.

Definition. Given a groupoid $(S, \circ)$. An element $e \in S$ is called a *neutral* (also *identity*) element if

$$e \circ x = x = x \circ e \quad \text{for every } x \in S. \tag{5.2}$$

□

If the operation is denoted by $\cdot$ then we usually use the term "identity element" instead of a neutral element.

Examples of neutral elements.

1) For $(\mathbb{R}, +)$ the number 0 is its neutral element, the same holds for $(\mathbb{Z}, +)$.
2) For $(\mathbb{R}, \cdot)$ the number 1 is its neutral (identity) element, the same holds for $(\mathbb{Z}, \cdot)$, and $(\mathbb{N}, \cdot)$.
3) For $(M_n, \cdot)$ where $\cdot$ is the multiplication of square matrices of order n the identity matrix is its neutral (identity) element.
4) $(\mathbb{Z}_n, \oplus)$ has the class $[0]_n$ as its neutral element.
5) $(\mathbb{Z}_n, \odot)$ has the class $[1]_n$ as its neutral (identity) element.

Example of a groupoid that does not have a neutral element. The groupoid $(\mathbb{N} \setminus \{0\}, +)$ does not have a neutral element. Indeed, there is not a positive number e for which $n + e = n = e + n$ for every positive $n \in \mathbb{N}$

5.1.4 Uniqueness of the Neutral Element. The following proposition shows that if a groupoid $(S, \circ)$ has its neutral element then it is unique.

Proposition. Given a groupoid $(S, \circ)$. If there exist elements e and f such that for every $x \in S$ we have $e \circ x = x$ and $x \circ f = x$, then $e = f$ is the neutral element of $(S, \circ)$. □

Justification. Consider the product $e \circ f$. From the property of e we have $e \circ f = f$ (indeed, take $x = f$); from the property of f we have $e \circ f = e$ (indeed, take $x = e$). Hence $e = f$, and in this case e is the neutral element. □

5.1.5 Monoids.

We will be mainly interested in semigroups which have the neutral element; they will be called monoids.

Definition. If in a semigroup $(S, \circ)$ there exists a neutral element then we call $(S, \circ)$ a *monoid*. □

In the paragraph above, we gave couple of examples of monoids and also an example of a semigroup which is not a monoid.

Convention. In the following text, the fact that $(S, \circ)$ is a monoid with the neutral element e will be shortened to $(S, \circ, e)$.

5.1.6 Powers in a Monoid.

Similarly as powers are defined in $(\mathbb{R}, \circ, 1)$ we can introduce powers in an arbitrary monoid.

Definition. Given a monoid $(S, \circ, e)$ and its element $a \in S$. The *powers* of a are defined by:

$$a^0 = e, \;\; a^{i+1} = a^i \circ a \;\text{ for every } i \geq 0.$$

□

Note that if the operation is $+$ with neutral element 0 then we write $0\,a = 0$ instead of a^0 and $k\,a$ instead of a^k.

5.1.7 Invertible Elements.

In many examples given above, we can somehow "reverse" the operation. For instance, in $(\mathbb{R}, +, 0)$ we can subtract; in $(\mathbb{R}, \cdot, 1)$ we can divide by any nonzero number; in $(M_n, \cdot, E)$ where M_n is the set of all square matrices of order n, and E is the identity matrix, we can cancel all the regular matrices (this means multiplying by the inverse matrix to a given regular one). In this paragraph, roughly speaking, we characterize those elements of a monoid that not only permit "cancellation" but "help solving equations". More precisely:

Definition. Given a monoid $(S, \circ, e)$. We say that an element $a \in S$ is *invertible* if there exists an element $y \in S$ such that

$$a \circ y = e = y \circ a. \tag{5.3}$$

□

Let us show that if y from 5.3 exists then it is unique.

Proposition. Given a monoid $(S, \circ, e)$. Assume that there are elements $a, x, y \in S$ such that

$$x \circ a = e \;\text{ and }\; a \circ y = e,$$

then $x = y$. □

Justification. Consider the product $x \circ a \circ y$. Since we are in a semigroup it holds that

$$y = e \circ y = (x \circ a) \circ y = x \circ (a \circ y) = x \circ e = x.$$

□

5.1.8 The Inverse Element.

Since y from 5.3 is unique we can define:

Definition. Let $(S, \circ, e)$ be a monoid, and $a \in S$ an invertible element. Let $y \in S$ satisfy

$$a \circ y = e = y \circ a.$$

Then y is called the *inverse element to a* and is denoted by a^{-1}. □

Remark. If a binary operation is denoted by $+$ we speak about the *opposite* element (instead of the inverse element) and denote it by $-a$ (instead of a^{-1}). The reason is that we sometimes have two different binary operations defined on the same set, (indeed, on the set $\mathbb{R}$ we have both $+$ and $\cdot$), hence it is convenient to distinguish between "inverses" with respect to $+$ and with respect to $\cdot$.

5.1.9 We know that not every element of a general monoid is invertible. Indeed, consider for example the set of all square matrices together with multiplication and the identity matrix. Then only regular matrices are invertible, and moreover for any regular matrix A it holds that $(A^{-1})^{-1} = A$. The next proposition shows that properties of invertible elements and their inverses are the same in any monoid.

Proposition. Let $(S, \circ, e)$ be a monoid. Then

1. e is invertible and $e^{-1} = e$.
2. If a is invertible then so is a^{-1}, and we have $(a^{-1})^{-1} = a$.
3. If a and b are invertible elements then so is $a \circ b$, and we have $(a \circ b)^{-1} = b^{-1} \circ a^{-1}$.

□

Justification.
1. It suffices to notice that $e \circ e = e$, this immediately means that $e^{-1} = e$.

2. Assume that a is invertible. Then we have $a \circ a^{-1} = e = a^{-1} \circ a$. If we look at the last identities we see that a is the element such that if we multiply by it the element a^{-1} we get e. Hence $a = (a^{-1})^{-1}$.

3. Assume that a^{-1} and b^{-1} exist. Then

$$(a \circ b) \circ (b^{-1} \circ a^{-1}) = a \circ (b \circ b^{-1}) \circ a^{-1} = a \circ e \circ a^{-1} = a \circ a^{-1} = e.$$

Similarly, we get that $(b^{-1} \circ a^{-1}) \circ (a \circ b) = e$. We have shown that $(a \circ b)^{-1} = b^{-1} \circ a^{-1}$. □

Remark. Note that **it is not** always the case that $(a \circ b)^{-1} = a^{-1} \circ b^{-1}$. This holds when the operation $\circ$ is commutative, i.e. $x \circ y = y \circ x$ for every x and y.

5.1.10 An Invertible Element Can Be Canceled.

Proposition. Let $(S, \circ, e)$ be a monoid, and let $a \in S$ is its invertible element. Then

$$a \circ b = a \circ c, \text{ or } b \circ a = c \circ a \quad \text{implies} \quad b = c.$$

□

Justification. Assume that a^{-1} exists and

$$a \circ b = a \circ c. \tag{5.4}$$

Multiply 5.4 by a^{-1} form the left. We get

$$a^{-1} \circ (a \circ b) = a^{-1} \circ (a \circ c), \text{ which gives } (a^{-1} \circ a) \circ b = (a^{-1} \circ a) \circ c \text{ and } b = c.$$

Similarly for $b \circ a = c \circ a$. The only difference is that here we multiply by a^{-1} from the right. (Notice the similarity with matrix operations.) □

5.1.11 Groups. In couple of examples above, every element was invertible; indeed, it holds for $(\mathbb{Z}, +, 0)$, $(\mathbb{R} \setminus \{0\}, \cdot, 1)$, and $(\mathbb{Z}_n, +, 0)$. Such monoids are of great importance and they are called groups.

Definition. A monoid $(S, \circ, e)$ in which every element is invertible is called a *group*. □

Examples of groups. The following monoids are groups:

1) The monoid $(\mathbb{R}, +, 0)$. Indeed, for every $x \in \mathbb{R}$ there exists $-x$ for which $x + (-x) = 0 = (-x) + x$.
2) The monoid $(\mathbb{Z}, +, 0)$. Indeed, for each integer x there exists an integer $-x$ for which $x + (-x) = 0 = (-x) + x$.
3) The monoid $(\mathbb{R}^+, \cdot, 1)$, where $\mathbb{R}^+$ is the set of all positive real numbers. Indeed, for every positive real number x there exists a positive real number $\frac{1}{x}$ for which $x \cdot \frac{1}{x} = 1 = \frac{1}{x} \cdot x$.

4) The monoid $(\mathbb{Z}_n, \oplus, [0]_n)$. Indeed, for a class $[i]_n$ there exists a class $[n-i]_n$ for which $[i]_n \oplus [n-i]_n = [0]_n = [n-i]_n \oplus [i]_n$.
5) Let A be the set of all permutation of the set $\{1, 2, \dots, n\}$, and let $\circ$ be the composition of permutations. Then $(A, \circ)$ is a monoid with the neutral element the identity permutation id. Moreover, for every permutation ϕ there exists its inverse permutation ϕ^{-1} for which $\phi \circ \phi^{-1} = id = \phi^{-1} \circ \phi$.

Examples of monoids that are not groups.

1) The monoid $(\mathbb{Z}, \cdot, 1)$. Indeed, for example 2 is not invertible because there is no **integer** k such that $2 \cdot k = 1$.
2) The monoid $(\mathbb{Z}_n, \odot, [1]_n)$. Indeed, the class $[0]_n$ is not invertible because for any $[i]_n$ we have $[0]_n \odot [i]_n = [0]_n \neq [1]_n$.
3) Let B be the set of all mappings from the set $\{1, 2, \dots, n\}$ into itself, where $n > 1$. Let $\circ$ be the composition of mappings. Then $(B, \circ, id)$ is a monoid where id is the identity mapping. Any mapping that is not one-to-one is not invertible.

5.1.12 Groups can be characterized as those semigroups $(S, \circ)$ where every equation $a \circ x = b$ and $y \circ a = b$ has a solution. In that case, the solution is unique. From this it immediately follows that

1. If $(S, \circ)$ is not a group, then there is an equation which does not have a solution.
2. Given a semigroup $(S, \circ)$. If there exists an equation with two distinct solutions, then $(S, \circ)$ is not a group, and moreover there is an equation that does not have a solution.

The following two paragraphs prove it.

5.1.13 Proposition. Given a group $(S, \circ)$ with its neutral element e. Then for every two elements $a, b \in S$ there exist unique $x, y \in S$ such that

$$a \circ x = b, \qquad y \circ a = b.$$

□

Justification. Since $(S, \circ, e)$ is a group and $a \in S$, there exists its inverse a^{-1}. If we multiply the equation $a \circ x = b$ by a^{-1} from the left we obtain

$$x = (a^{-1} \circ a) \circ x = a^{-1} \circ (a \circ x) = a^{-1} \circ b.$$

Similarly we obtain $y = b \circ a^{-1}$ from the second equation; indeed, we multiply the second equation by a^{-1} from the right and get the desired solution.

Let us show the uniqueness. Assume that $a \circ x_1 = b$ and $a \circ x_2 = b$. Then $a \circ x_1 = a \circ x_2$. Now, the proposition 5.1.10 completes the argument because it states that $x_1 = x_2$. Similarly from $y_1 \circ a = b$ and $y_2 \circ a = b$ we get $y_1 = y_2$.

5.1.14 Theorem. A semigroup $(S, \circ)$ is a group if and only if every equation of the form $a \circ x = b$ and every equation of the form $y \circ a = b$ has at least one solution.

More precisely: A semigroup $(S, \circ)$ is a group if and only if for every two elements $a, b \in S$ there exist $x, y \in S$ such that $a \circ x = b$ and $y \circ a = b$. □

Justification. First we show that if a semigroup $(S, \circ)$ satisfies the above conditions then it has got a neutral element.

Choose any $a \in S$. There exists $e_a \in S$ such that $e_a \circ a = a$; indeed, it is a solution of $y \circ a = a$. Now, take an arbitrary $b \in S$. We know that $b = a \circ x$ for some $x \in S$, hence

$$e_a \circ b = e_a \circ (a \circ x) = (e_a \circ a) \circ x = a \circ x = b.$$

Similarly, it can be shown that the element f_a for which $a \circ f_a = a$ satisfies $b \circ f_a = b$ for any $b \in S$.

Therefore, from 5.1.4 we get that $e_a = f_a$ is the neutral element of $(S, \cdot)$.

To show that every element $a \in S$ is invertible, it suffices to use the proposition from 5.1.7. Indeed, from the fact that there exist $x, y \in S$ with $a \circ x = e$ and $y \circ a = e$ we know that $x = y$, and $x = a^{-1}$. So, a is invertible. Since a was an arbitrary element of S, $(S, \circ, e)$ is a group. □

5.1.15 Commutative Semigroups, Monoids, Groups. In many examples above (but not in all) it does not matter whether we calculate $a \circ b$ or $b \circ a$, we get the same results.

Definition. A semigroup $(S, \circ)$ (monoid, group) is called *commutative* if it satisfies the *commutative law*, i.e. for every two elements $x, y \in S$

$$x \circ y = y \circ x.$$

□

5.1.16 Subsemigroups. Given a semigroup $(S, \circ)$ and a set $T \subseteq S$. It may happen (but does not need to) that T together with the same operation $\circ$ is again a semigroup. In that case, we will call $(T, \circ)$ a subsemigroup of $(S, \circ)$.

Definition. Given a semigroup $(S, \circ)$. A subset $T \subseteq S$ together with an operation $\circ$ forms a *subsemigroup* of the semigroup $(S, \circ)$, if for every two elements $x, y \in T$ we have $x \circ y \in T$. (In this case $(T, \circ)$ is also a semigroup.) □

Remark. Next, we will say less exactly "T is a subsemigroup" instead of "T forms a subsemigroup". It will be mainly in the situation where the operation is clear from the context.

Examples of subsemigroups. The following are examples of subsemigroups:

1) $\mathbb{N}$ together with addition forms a subsemigroup of $(\mathbb{Z}, +)$.
2) The set of all regular matrices together with multiplication of matrices forms a subsemigroup of $(M_n, \cdot)$, where M_n is the set of all square matrices of order n.
3) The set of all positive real numbers together with multiplication forms a subsemigroup of $(\mathbb{R}, \cdot)$.

Example of a subset that does not form a subsemigroup. The set of all regular square matrices of order n together with addition of matrices does not form a subsemigroup of $(M_n, +)$. Indeed, it does not hold that sum of two regular matrices is a regular matrix, e.g. coincide the identity matrix E. Then E and $-E$ are regular matrices but $E + (-E)$ is the zero matrix which is not regular.

5.1.17 Submonoids.

Definition. Given a monoid $(S, \circ, e)$. A subset $T \subseteq S$ forms a submonoid if it forms a subsemigroup and moreover $e \in T$. (In this case $(T, \circ, e)$ is also a monoid.) □

Examples of submonoids.

1) The set of all natural numbers $\mathbb{N}$ together with addition is a submonoid of $(\mathbb{Z}, +, 0)$, since $0 \in \mathbb{N}$.
2) The set of all regular square matrices of order n together with multiplication of matrices forms a submonoid of $(M_n, \cdot, E)$, since the identity matrix E is regular.
3) Denote by T_X the set of all mappings from a set X into itself. Consider the operation composition of mappings $\circ$. Then $(T_X, \circ, id)$ where id is the identity mapping (defined by $id(x) = x$ for all $x \in X$) is a monoid. The set of all bijections from T_X forms a submonoid of $(T_X, \circ)$, indeed, a composition of two bijections is a bijection, and the identity mapping is a bijection.

5.1.18 Remark. Notice that a subsemigroup $(T, \circ)$ of $(S, \circ, e)$ may contain a neutral element which is different from the neutral element e (but in this case $e \notin T$). If this is the case $(T, \circ)$ is a subsemigroup of $(S, \circ)$ but not a submonoid of $(S, \circ, e)$. Next, there is an example of such a situation.

Example. Let $X = \{1, 2, 3\}$. Denote by S the set of all mappings from X to X. Then $(S, \circ, id)$ is a monoid ($\circ$ is the composition of mappings, id is the identity mapping).

Consider the mapping $f\colon X \to X$ defined by $f(1) = 2$, $f(2) = 3$, $f(3) = 4$, and $f(4) = 2$. Then $f^4 = f$ and $T = \{f, f^2, f^3\}$ forms a subsemigroup of $(S, \circ, id)$. T does not form a submonoid, since $id \notin T$. On the other hand, f^3 is the neutral element of $(T, \circ)$ and $(T, \circ, f^3)$ is in fact a group. Indeed, $f \circ f^3 = f = f^3 \circ f$, $f^2 \circ f^3 = f^2 = f^3 \circ f^2$, and $f^3 \circ f^3 = f^3$.

5.1.19 The Group of Invertible Elements. Every monoid contains a special submonoid, the one formed by all invertible elements. And this submonoid is in fact a group that is called the *group of invertible elements.* Let us first prove the following proposition which justifies the definition coming next.

Proposition. Given a monoid $(S, \circ, e)$. Denote by $S^\star$ the set of all its invertible elements. Then $(S^\star, \circ, e)$ is a submonoid of $(S, \circ)$ which is a group. □

Justification. The above proposition immediately follows from 5.1.9. Indeed, $e \in S^\star$, and if $a, b \in S^\star$ then $a \circ b \in S^\star$. So $S^\star$ forms a submonoid.

Moreover, $(S^\star, \circ, e)$ is a group because if $a \in S^\star$ then $a^{-1} \in S^\star$. □

Definition. The group $(S^\star, \circ, e)$ is called the *group of invertible elements* of the monoid S. □

5.1.20 The following theorem is an important fact and is used in a lot of applications. In fact it holds for any finite group but we will state and prove it only for commutative ones now.

Theorem. Let $(G, \circ, e)$ be a finite commutative group. Then for every $a \in G$ we have $a^{|G|} = e$. □

Justification. Assume that the group has n elements and denote $G = \{a_1, a_2, \ldots, a_n\}$. Take any $a \in G$ and form the set $H = \{a \circ a_1, a \circ a_2, \ldots, a \circ a_n\}$. The H has also n elements; indeed, if $a \circ a_i = a \circ a_j$ in a group then $a_i = a_j$ (see 5.1.10).

Therefore, $G = H$ and because G is a commutative group we have

$$a_1 \circ a_2 \circ \ldots \circ a_n = (a \circ a_1) \circ (a \circ a_2) \circ \ldots \circ (a \circ a_n),$$

and also

$$a_1 \circ a_2 \circ \ldots \circ a_n = a^n \circ (a_1 \circ a_2 \circ \ldots \circ a_n).$$

If we multiply the last equality by $(a_1 \circ a_2 \circ \ldots \circ a_n)^{-1}$ we get $a^n = e$. □

5.2 Applications to $(\mathbb{Z}_n, \cdot, 1)$

Let us first introduce the *Euler function.*

5.2.1 Euler function. Given a natural number $n > 1$. Then the value of Euler function $\phi(n)$ equals to the number of all natural numbers i, $0 \leq i < n$, that are relatively prime to n. □

For example $\phi(6) = 2$, since there are only two natural numbers between 0 and 5 that are relatively prime to 6, namely 1 and 5.

5.2.2 Properties of Euler Function.

1. Let p be a prime number, then $\phi(p) = p - 1$.
2. If p is a prime number and $k \geq 1$ then $\phi(n) = p^k - p^{k-1}$.
3. If n and m are relatively prime natural numbers then $\phi(n \cdot m) = \phi(n) \cdot \phi(m)$.

□

It is not difficult to show the first two properties above. The easiest way how to prove the last one is to use the Chinese Remainder Theorem which is beyond the scope of this course.

5.2.3 The Group of Invertible Elements of $(\mathbb{Z}_n, \cdot, 1)$. We will use the facts from 5.1.19 for the commutative monoid $(\mathbb{Z}_n, \cdot, 1)$. We know $(\mathbb{Z}_n, \cdot)$ is a monoid with its neutral element 1. The set of all invertible elements of it is

$$\mathbb{Z}_n^\star = \{i \,|\, 0 \leq i < n,\ \ i \text{ and } n \text{ are relatively prime}\}.$$

Therefore, $(\mathbb{Z}_n^\star, \cdot, 1)$ is a group with $\phi(n)$ elements where $\phi(n)$ is the Euler function of n.

5.2.4 Euler-Fermat Theorem. Applying 5.1.20 we get a theorem which generalizes of the small Fermat theorem:

Theorem (Euler-Fermat). Given a natural number $n > 1$. Then for every integer a relatively prime to n we have

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

□

Justification. Indeed, take any integer a relatively prime to n. Put b to be the remainder when we divide a by n. Then $b \in \mathbb{Z}_n^\star$. Since $(\mathbb{Z}_n^\star, \cdot, 1)$ is a finite group with $\phi(n)$ elements, the Euler-Fermat Theorem is a consequence of 5.1.20. □

Remark. The small Fermat theorem is an immediate consequence of the Euler-Fermat theorem. Indeed, if n is a prime number then $\phi(n) = n - 1$.

5.3 Subgroups

Analogously as we defined subsemigroups and submonoids we can define subgroups. Subgroups are formed by subsets that not only form itself a group but group with the original operations. More precisely:

Definition. Given a group $(G, \circ, e)$. We say that $H \subseteq G$ forms a *subgroup* of $(G, \circ, e)$ if

1. for every $x, y \in H$ it holds that $x \circ y \in H$, (i.e. forms a subsemigroup);
2. $e \in H$, (i.e. forms a submonoid);
3. for every $x \in H$ it holds that $x^{-1} \in H$.

□

Note, that in this case, $(H, \circ, e)$ is also a group.

Remark. Every group $(G, \circ, e)$ with more than one element has at least two subgroups; indeed, one formed by $\{e\}$ and second formed by G. These two subgroups are called *trivial subgroups*.

5.3.1 How Many Elements a Subgroup Can Have? We will show some useful properties of finite groups and their subgroups. The first theorem shows that a subset of a group can form a subgroup only if its number of elements divides the number of elements of the group. Hence, $(\mathbb{Z}_7, +, 0)$ has only trivial subgroups; indeed, 7 is a prime number with divisors 1 and 7. And any subgroup with 1 element consists of 0, a subgroup with 7 elements is $(\mathbb{Z}_7, +, 0)$.

Theorem. Let $(G, \circ, e)$ be a finite group and $H \subseteq G$ its subgroup. Then the number of elements of H divides the number of elements of G. □

Justification. Let us denote $n = |G|$ and $k = |H|$. For every $g \in G$ we form a subset of G: $g \circ H = \{g \circ x \,|\, x \in H\}$.

We show that for every $g_1, g_2 \in G$ the sets $g_1 \circ H$ and $g_2 \circ H$ are either the same or they are disjoint (they do not have a common element).

Assume that $(g_1 \circ H) \cap (g_2 \circ H) \neq \emptyset$. Then there exist $h_1, h_2 \in H$ such that $g_1 \circ h_1 = g_2 \circ h_2$. Since we are in a group, we have

$$g_1 = (g_2 \circ h_2) \circ h_1^{-1} = g_2 \circ (h_2 \circ h_1^{-1}) \text{ and } g_2 = (g_1 \circ h_1) \circ h_2^{-1} = g_1 \circ (h_1 \circ h_2^{-1}). \qquad (5.5)$$

This means that $g_1 \in g_2 \circ H$ and $q_2 \in g_1 \circ H$, (indeed, H is a subgroup so $h_2 \circ h_1^{-1}, h_1 \circ h_2^{-1} \in H$).

Now, take an arbitrary element $x \in g_1 \circ H$. Then $x = g_1 \circ h$ for some $h \in H$. Substituting form 5.5 we get

$$x = (g_2 \circ (h_2 \circ h_1^{-1})) \circ h = g_2 \circ (h_2 \circ h_1^{-1} \circ h) \text{ and so } x \in g_2 \circ H.$$

Indeed, H is a subgroup so $h_2 \circ h_1 \circ h$ belongs to H.

Similarly, one gets that any $z \in g_2 \circ H$ belongs to $g_1 \circ H$. So, we have shown that $g_1 \circ H = g_2 \circ H$.

H is a subgroup, so $e \in H$, and therefore $g \in g \circ H$ for every $g \in G$. This means that every element from G belongs to some $g' \circ H$. Hence, the system $\{g \circ H \,|\, g \in G\}$ forms a partition of G.

To finish the argument, we show that all sets $g \circ H$ have the same number of elements which is $k = |H|$. Denote $H = \{h_1, \ldots, h_k\}$. Then

$$g \circ H = \{g \circ h_1, \ldots, g \circ h_k\}.$$

If $g \circ h_i = g \circ h_j$ then $(g^{-1} \circ g) \circ h_i = (g^{-1} \circ g) \circ h_2$, which means that $h_i = h_j$ (see also 5.1.10).

We have shown that the set of n elements is divided into disjoint parts each of them having k elements. Hence n is divisible by k. (Note that there are n/k distinct sets $g \circ H$.) □

5.3.2 Order of a Finite Group. The number of elements of a finite group $(G, \circ, e)$ is often called its *order*. The above theorem can be formulated as follows: The order of any subgroup $(H, \circ, e)$ of a finite group $(G, \circ, e)$ divides the order of $(G, \circ, e)$.

5.3.3 Subgroup Generated by an Element, Order of an Element. Let $(G, \circ, e)$ be a finite group, choose an element $a \in G$. Consider the set of all powers of a:

$$\{a, a^2, a^3, \ldots, a^k, \ldots\}.$$

Since G is a finite set, there must exist i and j, $i \neq j$, such that $a^i = a^j$. Let us assume that i is the exponent which is smaller than j. We are in a group, so there exists a^{-1}. Therefore

$$a^i = a^j \text{ implies } a^{i-1} = a^{j-1}, \text{ etc. } e = a^0 = a^{j-i}.$$

Hence, we have proved the first part of the following proposition:

Proposition. Let $(G, \circ, e)$ be a finite group, $a \in G$. Then there exists the smallest positive integer r for which $a^r = e$. Moreover, $\{a, a^2, \ldots, a^r\}$ forms a subgroup of $(G, \circ, e)$. □

Justification. The second part follows from the fact that

1. $a^i \circ a^j = a^{i+j} = a^k$ where $k \equiv i + j \bmod r$.
2. $a^r = e \in \{a, a^2, \ldots, a^r\}$.
3. $(a^i)^{-1} = a^{r-i}$.

Definition. The subgroup formed by $\{a, a^2, \ldots, a^r\}$ is called the *subgroup generated by a* and will be denoted by $\langle a \rangle$.

The number of elements of $\langle a \rangle$ (i.e. the smallest positive r for which $a^r = e$) is called the *order of a* and it is denoted by $r(a)$. □

Note that the order of a is in fact the order of the subgroup $\langle a \rangle$.

5.3.4 The fact that $\langle a \rangle$ forms a subgroup of $(G, \circ, e)$ gives us

Corollary. Given a finite group $(G, \circ, n)$ with n elements. Then the order of any element $a \in G$ divides n.

This proposition is a direct consequence of 5.3.1. Indeed, $\langle a \rangle$ is a subgroup of the group $(G, \cdot, e)$ having $r(a)$ elements.

5.3.5 Theorem. Given a finite group $(G, \circ, e)$ with n elements. Then for every $a \in G$ we have

$$a^n = e.$$

Justification. Indeed, since $r(a)$ divides n, we get

$$a^n = a^{k\, r(a)} = (a^{r(a)})^k = e^k = e.$$

□

5.3.6 A Characterization of the Order r(a). The following proposition will help us for example to find the order of of powers of a given element (see **??**) of a finite group.

Proposition. A number r equals to the order $r(a)$ of a in a finite group $(G, \cdot, e)$ if and only if the following two conditions are satisfied:

1) $a^r = e$.
2) If $a^s = e$ for some natural number s then r divides s.

□

Justification. a) Let us assume that r satisfies the two conditions above. Then clearly, r is the smallest positive integer for which $a^r = e$; hence $r = r(a)$.

b) Denote the order $r(a)$ by r. We show that r satisfies the two conditions above. The first condition is obvious. Consider any s for which $a^s = e$. Divide s by r, we get $s = qr + z$ where the remainder z satisfies $0 \le z < r$. Then

$$e = a^s = a^{qr+z} = (a^r)^q \cdot a^z = e^q \cdot a^z = a^z.$$

Since z is strictly smaller than r, and r is the smallest positive number for which $a^i = e$, we get $z = 0$. And hence r divides s. □

5.3.7 Cyclic Group, a Generating Element of a Group. There is a special type of groups, in fact the “most simple” ones, where the calculation corresponds to the addition in $\mathbb{Z}_r$. More precisely:

Definition. Given a group $\mathcal{G} = (G, \circ, e)$. If there exists an element $a \in G$ for which $\langle a \rangle = G$ we say that the group is *cyclic* and that a is a generating element of $(G, \circ, e)$. □

Remark. Note that a cyclic group does not need to be finite. Even in an infinite group $(G, \circ, e)$ we can form a subgroup generated by $a \in G$, indeed,

$$\langle a \rangle = \{\ldots, a^{-2}, a^{-1}, a^0, a^1, a^2, \ldots\} = \{a^i \mid i \in \mathbb{Z}\}.$$

If $\langle a \rangle = G$ then the group is cyclic.

5.3.8 Examples.

1. $(\mathbb{Z}_n, +, 0)$ (for any natural number $n > 1$) is a cyclic group with its generating element 1.
2. For every prime number p the group $(\mathbb{Z}_p^\star, \cdot, 1)$ is a cyclic group. It is not straightforward to show it. Moreover, to find a generating element is a difficult task for some primes p.
3. The group $(\mathbb{Z}_8^\star, \cdot, 1)$ **is not** cyclic. We have $\mathbb{Z}_8^\star = \{1, 3, 5, 7\}$ and $3^2 = 1$, $5^2 = 1$ and $7^{-1} = 1$. So, there is no element with order 4.
4. $(\mathbb{Z}, +, 0)$ of all integers together with addition is a cyclic group; its generating element is 1.

5.3.9 Observation. One can reformulate the definition of a finite cyclic group: A finite group $\mathcal{G} = (G, \circ, e)$ of order n is cyclic if and only if there exists $a \in G$ with its order $r(a) = n$.

5.3.10 Order of a Power of a. If we know the order of an element of a in a finite group $(G, \circ, e)$ then we can determine the order of a^i for any $i \in \mathbb{N}$, see the following proposition.

Proposition. Let $\mathcal{G} = (G, \circ, e)$ be a finite group. Let $a \in G$ have order $r(a)$. Then

$$r(a^i) = \frac{r(a)}{\gcd(r(a), i)}.$$

□

Justification. We will show that the number $\frac{r(a)}{\gcd(r(a),i)}$ satisfies the conditions of proposition 5.3.9 and hence it is $r(a^i)$.

Denote $r = r(a)$, and $d = \gcd(i, r)$. Then we can write $i = d\,i'$ and $r = d\,r'$ where i' and r' are relatively prime. With this notation $\frac{r(a)}{\gcd(r(a),i)}$ equals to r'.

We show the first condition from 5.3.6: we have

$$(a^i)^{r'} = a^{i\,r'} = a^{i'\,d\,r'} = (a^{d\,r'})^{i'} = (a^r)^{i'} = e.$$

The second condition from 5.3.6: Assume that $(a^i)^s = a$. Then $a^{i\,s} = e$. Since r is the order of a, necessarily r divides $i\,s$. Further

$$i\,s = k\,r, \text{ i.e. } i'\,d\,s = k\,r'\,d \text{ and } i'\,s = k\,r'.$$

Numbers i' and r' are relatively prime, and r' divides $i'\,s$, hence r' divides s. So r' is the order of a^i as required. □

5.3.11 Observation. The proposition above helps to find orders of all elements b belonging to $\langle a \rangle$. Indeed, we know that the subgroup $\langle a \rangle$ is a cyclic group having a as its generating element. So we can use the proposition from 5.3.9 for every element $b \in \langle a \rangle$. Especially, if we know a generating element of a cyclic group we can find orders of all elements of the group.

5.3.12 The proposition in 5.3.9 can be used to calculate the number of generating elements in any finite cyclic group. Indeed, if a is a generating element of a finite cyclic group $\mathcal{G} = (G, \circ, e)$ with n elements, then $b = a^i$ is also a generating element of $\mathcal{G}$ if and only if $\gcd(i, n) = 1$; and there are $\phi(n)$ such i's. Hence we get the following corollary

Corollary. Given a finite cyclic group $\mathcal{G} = (G, \circ, e)$ with n elements. Then $\mathcal{G}$ has $\phi(n)$ different generating elements. □

5.3.13 Subgroups of a Finite Cyclic Group. Subgroups of a finite cyclic group are easy to describe. The next proposition states that a finite cyclic group with n elements has a subgroup of order d for any divisor d of n. Notice, that it is not true for a finite group which is not cyclic.

Proposition. Given a finite cyclic group $\mathcal{G} = (G, \circ, e)$ with n elements. Then for every natural number d which divides n there exists a subgroup of $\mathcal{G}$ with d elements. □

Justification. Denote by a one of generati+ng elements of the group $\mathcal{G}$. Then the subgroup $\langle a^k \rangle$ where $k = \frac{n}{d}$ had d elements. Indeed, we have

$$\langle a^k \rangle = \{a^k, a^{2k}, \ldots, a^{dk} = e\}.$$

5.3.14 Remark. A finite cyclic group has only subgroups that itself are cyclic.

Justification. Let $\mathcal{G} = (G, \circ, e)$ be a finite cyclic group with a generating element a. Consider two elements $b, c \in G$; then $b = a^i$ and $c = a^j$ for some $i, j \in \{1, 2, \ldots, |G|\}$. Any subgroup which contains these two elements must contain also all elements of the form a^{ix+jy} where x and y are any integers. From the Bezout's Theorem we know that the equation $ix + jy = k$ has integer solutions if and only if the greatest common divisor of i and j divides k. Therefore the smallest subgroup containing $b = a^i$ and $c = a^j$ is $\langle a^d \rangle$ where $d = \gcd(i, j)$.

Chapter 6

Structures with Two Binary Operations

In the last two lectures we investigated groupoids, semigroups, and groups as examples of a set with one binary operation. Now, we will be interested in structures that consist of a nonempty set together with two binary operations, as fields, lattices and Boolean algebras.

6.1 Rings and Fields

Consider the set of all real numbers $\mathbb{R}$. On $\mathbb{R}$, two binary operations are defined: addition $+$ and multiplication $\cdot$. We know that $(\mathbb{R}, +, 0)$ is a commutative group, $(\mathbb{R}, \cdot, 1)$ a commutative monoid. Moreover, for the operations the distributivity laws hold: For all $a, b, c \in \mathbb{R}$ it holds that

$$a\,(b+c) = a\,b + a\,c \quad \text{and} \quad (b+c)\,a = b\,a + c\,a.$$

Another example: Consider the set of all square matrices M_n of order n. We can add two matrices, we can multiply two matrices. In fact, $(M_n, +, O)$ (O is the zero matrix) is a commutative group, $(M_n . \cdot, E)$ (E is the identity matrix) is a monoid. And moreover the operations $+$ and $\cdot$ satisfy the distributivity laws.

Two examples above do not have the same properties; indeed, $(M_n, \cdot, E)$ is not commutative, in $(\mathbb{R}, \cdot, 1)$ every number $x \neq 0$ has its inverse, whereas only regular matrices are invertible in $(M_n, +, \cdot)$. The following notions capture such differences.

6.1.1 A (Commutative) Ring with Identity.

Definition. A nonempty set M together with two binary operations $+$ and $\cdot$ is called a *ring with identity* if $(M, +, 0)$ is a commutative group, $(M, \cdot, 1)$ is a monoid and two distributive laws hold

$$a \cdot (b+c) = a \cdot b + a \cdot c \quad \text{and} \quad (b+c) \cdot a = b \cdot a + c \cdot a$$

for every $a, b, c \in M$.

If moreover the multiplication $\cdot$ is commutative then the ring is called a *commutative ring with identity.* □

Convention. We will denote a ring with identity by $(M, +, \cdot)$, where $M \neq \emptyset$. Further, we denote the neutral element of the commutative group $(M, +)$ as 0, and the neutral element of $(M, \cdot)$ by 1. Moreover, $-a$ is the opposite (inverse) element to a in $(M, +, 0)$, and a^{-1} the inverse of a in $(M, \cdot, 1)$ if it exists.

Remark. Notice, that in any ring with identity we have $0 \cdot a = 0 = a \cdot 0$ and $(-a) \cdot b = a \cdot (-b) = -ab$.

6.1.2 Examples of Rings.

1. $(\mathbb{R}, +, \cdot)$ where $\mathbb{R}$ is the set of all real numbers with addition $+$ and multiplication $\cdot$ forms a commutative ring with identity.
2. $(M_n, +, \cdot)$ where M_n is the set of all real square matrices of order n, $+$ is the matrix addition and $\cdot$ is the matrix multiplication forms a ring with identity which is **not commutative** (note that multiplication of matrices is not commutative).
3. $(\mathbb{Z}, +, \cdot)$ where $\mathbb{Z}$ is the set of all integers together with addition and multiplication forms a commutative ring with identity.
4. $(\mathbb{Z}_n, +, \cdot)$ where $+$ and $\cdot$ are operations in $\mathbb{Z}_n$ forms a commutative ring with identity where the class containing 1 is the identity element. This ring has got n elements.

6.1.3 Zero Divisors. There are rings with identity where we can obtain 0 even if we multiply two nonzero elements. We give two such examples.

1. Consider the product of the two following nonzero matrices:

$$\begin{pmatrix} 2 & 2 \\ 3 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

2. In $(\mathbb{Z}_6, \cdot, 1)$ we have $2 \cdot 3 = 0$ and $2 \neq 0 \neq 3$.

On the other hand, no product of two nonzero real numbers (or integers) equals 0. This motivates the following notion.

Definition. An element of a ring $(M, +, \cdot)$ is called a *zero divisor* if $a \neq 0$ and there exists $b \neq 0$ such that $a \cdot b = 0$. □

If a is a zero divisor in a ring $(M, +, \cdot)$ with identity then a is not invertible in $(M, \cdot, 1)$. Indeed, if a^{-1} exists then from $a \cdot b = 0$ we immediately get

$$b = (a^{-1} \cdot a) \cdot b = a^{-1} \cdot (a \cdot b) = a^{-1} \cdot 0 = 0.$$

So if a is invertible then from $a \cdot b = 0$ it follows that $b = 0$.

Let us mention that a ring with identity without zero divisors is called an *integral domain.*

6.1.4 A Field. A commutative ring with identity where every nonzero element is invertible (hence, with the similar properties as real numbers) is called a *field.* More precisely:

Definition. A commutative ring with identity is called a *field* if every nonzero element of M is invertible in $(M, \cdot, 1)$, and if $0 \neq 1$. □

Let us note that the condition $0 \neq 1$ only means that every field must have at least two elements, namely 0 and 1. So we exlude the “trivial” commutative ring having just one element 0. (In this ring $(\{0\}, +)$ is the same as $(\{0\}, \cdot)$.)

It is not difficult to see that $(\mathbb{Z}_2, +, \cdot)$ is a field with exactly two elements 0 and 1.

Examples.

a) $(\mathbb{R}, +, \cdot)$ is a field.
b) $(\mathbb{Z}_p, +, \cdot)$ where p is a prime number is a field.
c) $(\mathbb{Q}, +, \cdot)$ where $\mathbb{Q}$ is the set of all rational numbers is a field.
d) $(\mathbb{C}, +, \cdot)$ where $\mathbb{C}$ is the set of all complex numbers is a field.
e) $(M_n, +, \cdot)$, $n > 1$, **is not** a field because only regular matrices are invertible.
f) $(\mathbb{Z}_n, +, \cdot)$ where n is composite number **is not** a field, indeed, every divisor $i \neq 1$ of n is not invertible.

6.1.5 Remark. In the course of linear algebra one works with vector spaces, matrices, etc. Scalars there are taken from the field of real numbers or the field of complex numbers. This is not necessary; linear algebra can be built over *any* field (commutative ring with identity does not suffice). There are only small differences; for example, if we study vector spaces over a finite field with k elements, then any vector space of dimension n has only k^n elements. Similarly, there are only finitely many solutions of a system of linear equtions over a field with k elements.

We know that $(\mathbb{Z}_p, +, \cdot)$, p a prime, is an example of a finite field. These are not the only one examples. It can be shown that for any prime p and integer $k \geq 1$ there is a field with p^k elements (which is in some sense unique). Moreover, there are no finite fields with other number of elements. The construction of fields with p^k elements for $k > 1$ is beyond the scope of this course.

6.1.6 The Group of Invertible Elements of a Finite Field. Let $(F, +, \cdot)$ be a finite field. Then we know that $F \setminus \{0\}$ forms a group, the group of invertible elements of $(F, \cdot, 1)$. It can be proved that the group $(F^\star, \cdot, 1)$ is always cyclic. In other words, it has a generating element a (here called a primitive element) such that every non-zero element of F is a power of a. So once we know the correspondence between non-zero elements and a^i, multiplication and cancellation becomes rather easy. The proof of the following proposition is beyond the scope of the course.

Theorem. Let $(F, +, \cdot)$ be a finite field. Then the group of invertible elements of the monoid $(F, \cdot, 1)$ is a cyclic group. □

6.2 Boolean Algebras

There is an other type of structures with two binary operations than rings and fields — lattices and their special case Boolean algebras. First, let us recall some facts known from the set theory.

Properties of Subsets. Consider a nonempty set U and the set of all its subsets $\mathcal{P}(U)$. Let $\cap$ denote the intersection of sets, and $\cup$ the union of sets. Then for every sets $A, B, C \subseteq U$ we have

1. $A \cap (B \cap C) = (A \cap B) \cap C,\ \ A \cup (B \cup C) = (A \cup B) \cup C$ (associative law).
2. $A \cap B = B \cap A,\ \ A \cup B = B \cup A$ (commutative law).
3. $A \cap A = A,\ \ A \cup A = A$.
4. $A \cap (B \cup A) = A,\ \ A \cup (B \cap A) = A$.

6.2.1 A Lattice. The example above motivates the notion of a lattice. It will be a nonempty set together with two operations that will satisfy the above four properties. More precisely:

Definition. A *lattice* consists of a nonempty set M together with two binary operations on M; one is *meet* $\wedge$ and the other is *join* $\vee$ which satisfy the following four conditions

1. $A \wedge (B \wedge C) = (A \wedge B) \wedge C,\ \ A \vee (B \vee C) = (A \vee B) \vee C$ (associative law).
2. $A \wedge B = B \wedge A,\ \ A \vee B = B \vee A$ (commutative law).
3. $A \wedge A = A,\ \ A \vee A = A$.
4. $A \wedge (B \vee A) = A,\ \ A \vee (B \wedge A) = A$.

□

Hence, $(\mathcal{P}(U), \cap, \cup)$ is an example of a lattice.

Lemma. In every lattice we have

$$a \wedge b = a \quad \text{if and only if} \quad a \vee b = b.$$

□

Justification. Assume that $a \wedge b = a$. Then $b = b \vee (a \wedge b)$ (property 4); hence, $b = b \vee a$ because $a \wedge b = a$.

Similarly, assume that $a \vee b = b$. Then $a = a \wedge (b \vee a) = a \wedge (a \vee b) = a \wedge b$ because $a \vee b = b$. □

6.2.2 Partial Order on a Lattice. On $\mathcal{P}(U)$ we have not only two operations (union and intersection) but we have a partial order $\subseteq$ at the same time (for the definition of a partial order see 3.3.14). The following proposition states that in **any** lattice we can define a partial order, so any lattice is a poset. (Note that the opposite implication does not hold, there are posets which are not lattices.)

Proposition. Given a lattice $(M, \wedge, \vee)$. Define a relation $\sqsubseteq$ on M by:

$$a \sqsubseteq b \ \text{ if and only if } a \wedge b = a \ (\text{iff } a \vee b = b).$$

Then the relation $\sqsubseteq$ on a lattice $(M, \wedge, \vee)$ is reflexive, antisymmetric, and transitive; i.e. it is a partial order on M (and $(M, \sqsubseteq)$ is a poset). □

Justification. Let $(M, \wedge, \vee)$ be a lattice. Because for every element $a \in M$ we have $a \wedge a = a$, it holds that $a \sqsubseteq a$. In other words, the relation $\sqsubseteq$ is reflexive.

Assume that for some $a, b \in M$ it holds that $a \sqsubseteq b$ and $b \sqsubseteq a$. Then the first fact means that $a \wedge b = a$ and the second one $b \wedge a = b$. Since $a \wedge b = b \wedge a$, we get $a = b$, and the relation is antisymmetric.

Assume that for some $a, b, c \in M$ it holds that $a \sqsubseteq b$ and $b \sqsubseteq c$. Then $a \wedge b = a$ and $b \wedge c = b$. Hence

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

Therefore $a \sqsubseteq c$ and the relation is transitive. □

For the lattice $(\mathcal{P}(U), \cap, \cup)$ the partial order $\sqsubseteq$ is the relation "to be a subset", i.e. $A \sqsubseteq B$ if and only if $A \subseteq B$.

6.2.3 Remark. The operations $\wedge$ and $\vee$ are sometimes called an *infimum* and a *supremum*. This is because in the poset $(M, \sqsubseteq)$ the element $a \wedge b$ is the greatest lower bound and $a \vee b$ is the smallest upper bound of the set $\{a, b\}$.

6.2.4 The Smallest Element 0, and the Greatest Element 1.

Definition. Given a lattice $(M, \wedge, \vee)$. An element $\mathbf{0}$ for which

$$\mathbf{0} \wedge a = \mathbf{0}, \ \ \mathbf{0} \vee a = a \quad \text{for every } a \in M.$$

is called the *smallest element* of $(M, \wedge, \vee)$.

An element $\mathbf{1}$ for which

$$\mathbf{1} \wedge a = a, \ \ \mathbf{1} \vee a = \mathbf{1} \quad \text{for every } a \in M.$$

is called the *greatest element* of $(M, \wedge, \vee)$. □

Note that for $\mathbf{0}$ and any $a \in M$ we have $\mathbf{0} \sqsubseteq a$ for every $a \in M$; analogously, $a \sqsubseteq \mathbf{1}$ for every $a \in M$. Therefore, $\mathbf{0}$ is really the smallest element and $\mathbf{1}$ the greatest element of the poset $(M, \sqsubseteq)$.

6.2.5 Distributive Lattices. In $(\mathcal{P}(U), \cap, \cup)$ there are other laws that hold, two distributive laws are among them. On the other hand there are lattices where distributivity laws do not hold.

Definition. A lattice $(M, \wedge, \vee)$ is called a *distributive lattice* if it satisfies the distributive laws: For every elements $a, b, c \in M$ it holds that

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

□

It can be shown that if one of the distributivity laws holds so does the other one. Indeed, let us show e.g. that $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ for **every** $a, b, c \in M$ implies $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$.

Let us calculate:

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

we used the first distributivity law for $(a \vee b)$, a, and c. Further,

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

We used the absorption law 4, and commutativity. Now, the first distributivity law yields

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

as required.

Moreover, it is not difficult to see that in any lattice we have

$$(a \wedge b) \vee (a \wedge c) \sqsubseteq a \wedge (b \vee c), \ \ a \vee (b \wedge c) \sqsubseteq (a \vee b) \wedge (a \vee c).$$

6.2.6 Complements in Distributive Lattices. Another notion known from sets is a complement of a set A, i.e. the set of all those elements which do not belong to A.

Definition. Given a distributive lattice $(M, \wedge, \vee)$ with the smallest element $\mathbf{0}$ and the greatest element $\mathbf{1}$. We say that an element b is a *complement of* a, if

$$a \wedge b = \mathbf{0}, \ \text{ and } \ a \vee b = \mathbf{1}. \tag{6.1}$$

The complement of a is denoted by $\overline{a}$. □

The notation from the above definition is justified by the following proposition.

Proposition. Let $(M, \wedge, \vee)$ be a distributive lattice. Then if for $a, b, c \in M$ we have

$$a \vee b = a \vee c \quad \text{and} \quad a \wedge b = a \wedge c$$

then $b = c$.

Specially, if a has a complement, then the complement is unique. □

Justification. Assume that $a \vee b = a \vee c$ and $a \wedge b = a \wedge c$. Let us compute

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

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

as stated. □

Remark. Let us mention that a complement can be defined also in lattices that are not distributive; only one element can have more that one complement in lattices that are not distributive.

6.2.7 Boolean Algebras. Definition. A *Boolean algebra* is a distributive lattice with the smallest element $\mathbf{0}$ and the greatest element $\mathbf{1}$ in which every element has its complement. □

Proposition. Let $(B, \wedge, \vee)$ be a Boolean algebra with the smallest element $\mathbf{0}$, the greatest element $\mathbf{1}$, and the complement . Then for every elements $a, b, c \in B$ it holds that

1. $\overline{\mathbf{0}} = \mathbf{1}$, $\overline{\mathbf{1}} = \mathbf{0}$.
2. $\overline{a \wedge b} = \overline{a} \vee \overline{b}$, $\overline{a \vee b} = \overline{a} \wedge \overline{b}$.
3. $\overline{\overline{a}} = a$.

□

Justification. 1. follows from the fact that $\mathbf{0} \wedge \mathbf{1} = \mathbf{0}$ and $\mathbf{0} \vee \mathbf{1} = \mathbf{1}$.

2. It is an easy calculation. We show the first identity. We have

$$(a \wedge b) \wedge (\overline{a} \vee \overline{b}) = ((a \wedge b) \wedge \overline{a}) \vee ((a \wedge b) \wedge \overline{b}) = (\mathbf{0} \wedge b) \vee (a \wedge \mathbf{0}) = \mathbf{0}.$$

We used distributivity, associativity, commutativity and the fact that $a \wedge \overline{a} = \mathbf{0}$.

Similarly,

$$(a \wedge b) \vee (\overline{a} \vee \overline{b}) = (a \vee (\overline{a} \vee \overline{b}) \wedge (b \vee (\overline{a} \vee \overline{b})) = (\mathbf{1} \vee \overline{b}) \wedge (\mathbf{1} \vee \overline{a}) = \mathbf{1}.$$

We used distributivity, associativity, commutativity and the fact that $a \vee \overline{a} = \mathbf{1}$.

3. Indeed, a is the element for which $\overline{a} \vee a = \mathbf{0}$ and $\overline{a} \wedge a = \mathbf{1}$. □

Remark. We know that $\mathcal{P}(U)$ with the operations $\cap$ and $\cup$, where $\mathbf{0} = \emptyset$, $\mathbf{1} = U$ and $\overline{A} = \{x \in U \,|\, x \notin A\}$ forms a Boolean algebra. There are other Boolean algebras. We will end this chapter with description of all finite Boolean algebras. The one used mostly is the smallest one (and sometimes called "the Boolean algebra").

6.2.8 Boolean algebra B_2. The smallest Boolean algebra has 2 elements, 0 and 1 and operations are defined by:

Let $B_2 = \{0, 1\}$. Define

1. $\wedge$ is the logical product, i.e. $i \wedge j = \min\{i, j\}$,
2. $\vee$ is the logical addition, i.e. $i \vee j = \max\{i, j\}$,
3. $\mathbf{0} = 0$, $\mathbf{1} = 1$, and
4. $\overline{0} = 1$, $\overline{1} = 0$.

It is straightforward to verify all the properties that a Boolean algebra has.

6.2.9 Finite Boolean Algebras. For any $n = 2^k$ there is a Boolean algebra with n elements. It is the following one:

Denote by B_n the set of all k-tuples of 0 and 1, i.e.

$$B_n = \{(a_1, a_2, \ldots, a_k) \mid a_i \in \{0, 1\}\}.$$

Define

1. $(a_1, a_2, \ldots, a_k) \wedge (b_1, b_2, \ldots, b_k) = (a_1 \wedge b_1, a_2 \wedge b_2, \ldots, a_k \wedge b_k)$,
2. $(a_1, a_2, \ldots, a_k) \vee (b_1, b_2, \ldots, b_k) = (a_1 \vee b_1, a_2 \vee b_2, \ldots, a_k \vee b_k)$,
3. $\mathbf{0} = (0, 0, \ldots, 0)$, $\mathbf{1} = (1, 1, \ldots, 1)$,
4. $\overline{(a_1, a_2, \ldots, a_k)} = (\overline{a_1}, \overline{a_2}, \ldots, \overline{a_k})$.

Then B together with the above operations forms a Boolean algebra with $n = 2^k$ elements.

Notice that B_n can be viewed as the set of characteristic functions of subsets of $U = \{1, 2, \ldots, k\}$. On the other hand, you can look at B_n as a cartesian product of k copies of B_2 where operations are coordinatewise.

It can be proved that the Boolean algebras above are unique up to renaming elements (i.e. up to an isomorphism) and that there are no other finite Boolean algebras.

Chapter 7

Graphs

7.1 Directed and Undirected Graphs

Theory of graphs is a modern branch of mathematics. The first problem which is believed to lie in the foundation of it is the problem of seven bridges in Königsberg in Prussia (now Kaliningrad in Russia) over the river Pregel. There are two islands in the city. One island is connected by two bridges with each riverside, and by one bridge to the other island, moreover there is one bridge from the second island to each riverside. The problem was to find a walk through the city that would cross each bridge exactly once, and "swimming is forbidden", which means that once an island / a riverside is reached the walk must continue from the island / riverside.

In 1736 Leonard Euler proved that such a walk does not exist. He described the above situation by four points (vertices) representing two islands and two riversides, connected by seven lines (edges) representing bridges. We will learn more about his argument in the section Euler graphs 7.7.

Let us start with the notion of a directed graph even though the above problem led to an undirected one.

7.1.1 Directed Graphs. Roughly speaking, a directed graph consists of points called vertices, lines that go from one vertex to other vertex called edges. Because there may be more than one edge from A to B, in a general graph we distinguish between a name of an edge and the order pair A and B. More precisely:

Definition. A *directed graph* is a triple $G = (V, E, \varepsilon)$ where V is a nonempty finite set of *vertices* (also called *nodes*), E is a finite set of names of *directed edges* (also called *arrows*), and ε is an *incidence relation* which assigns to any edge $e \in E$ an ordered pair (u, v) of vertices $u, v \in V$. □

Further notions for directed graphs. Let $\varepsilon(e) = (u, v)$. Then u is called the *initial vertex* of e, denoted by $IV(e)$, and v the *terminal vertex* of e, denoted by $TV(e)$. We also say that vertices u, v are *end vertices* of e, or that e is *incident* to u, v.

If $u = v$ we call the edge e a *directed loop*.

If for two edges e_1 and e_2 we have $\varepsilon(e_1) = \varepsilon(e_2)$, the edges e_1 and e_2 are called *parallel.* □

7.1.2 Undirected Graphs. Roughly speaking, undirected graphs are graphs in which any edge "can be used in both directions" or "where the direction is not important". More precisely:

Definition. An *undirected graph* is a triple $G = (V, E, \varepsilon)$ where V is a nonempty finite set of *vertices* (also called *nodes*), E is a finite set of names of *edges*, and ε an *incidence relation* which assigns to any edge $e \in E$ a set $\{u, v\}$ where $u, v \in V$. □

Further notions for undirected graphs. Let $\varepsilon(e) = \{u, v\}$. Then u and v are called *end vertices* of e.

If $u = v$ then we call the edge e an *undirected loop.*

If for two edges e_1 and e_2 we have $\varepsilon(e_1) = \varepsilon(e_2)$, the edges e_1 and e_2 are called *parallel edges.*

If $\varepsilon(e) = \{u, v\}$ we say that vertices u, v are *incident to the edge* e and that the edge e is *incident to vertices* u, v.

7.1.3 Simple Graphs.

Definition. A graph (directed or undirected) is called *simple* if it does not contain parallel edges. □

Remark. In a simple graph the incidence relation is not necessary. Indeed, assume that G is a simple directed graph. Then any edge can be named by the ordered pair $(IV(e), TV(e))$ of its initial vertex $IV(e)$ and terminal vertex $TV(e)$. Therefore, in a simple directed graph the set of edges E will be a subset of $V \times V$. Similarly, if G is undirected graph, then for each $\{u, v\}$, $u, v \in V$ there is at most one edge with $\varepsilon(e) = \{u, v\}$. Hence $\{u, v\}$ can serve as the name of e. Here, $E \subseteq \{\{u, v\} \mid u, v \in V\}$.

In the following text, a simple graph (directed or undirected) will be denoted by $G = (V, E)$.

We will denote the set of vertices of a graph G by $V(G)$ and the set of edges by $E(G)$. If there is no fear of misunderstanding, we will write only V for the set of vertices, and E for the set of edges.

7.1.4 Vertex Degrees. Roughly speaking, degree of a vertex is the number of edges that end or start in a given vertex. In directed graphs it will be useful to distinguish between the number of edges that start in v (out degree) and the number of edges that terminate in v (in degree).

Definition. Given a directed graph $G = (V, E, \varepsilon)$. The *in-degree* of a vertex v, denoted by $d^-(v)$, equals to the number of edges for which v is the terminal vertex; i.e.

$$d^-(v) = |\{e \in E; TV(e) = v\}|.$$

The *out-degree* of a vertex v, denoted by $d^+(v)$, equals to the number of edges for which v is its initial vertex; i.e.

$$d^+(v) = |\{e \in E; IV(e) = v\}|.$$

The *degree* of a vertex v, denoted by $d(v)$, is

$$d(v) = d^-(v) + d^+(v),$$

(i.e. it is the number of edges that are incident to v). □

Notice that any loop is calculated twice; indeed, once for the in-degree, once for the out-degree of v.

Definition. Given an undirected graph $G = (V, E, \varepsilon)$. The *degree* of a vertex v, denoted by $d(v)$, equals to the number of edges for which v is their end vertex where a loop is calculated twice. □

Notice that if we ”forget” direction in a directed graph G and obtain an undirected graph G', then for every vertex v we have $d_G(v) = d_{G'}(v)$, i.e. the degrees are the same. Indeed, this is due to the fact that each loop is calculated twice even in an undirected graph.

7.1.5 One of the easy but extremely useful facts about degrees in a graph is the following proposition. (In the English literature, it is also called Handshaking lemma.)

Proposition. For every graph G (directed or undirected) we have

$$\sum_{v \in V} d(v) = 2\,|E|,$$

where $|E|$ is the number of edges in G. □

Justification. Let e be an edge of a graph G. Then e adds 2 to the sum of all degrees; indeed, if e is a loop then it is calculated twice by definition, if e is not a loop then it has two end vertices. Therefore, the sum of all degrees is twice the number of edges. □

Since by the proposition above, the sum of all degrees is an even number, we get the next corollary.

Corollary. Every graph has an even number of vertices with odd degree. □

7.1.6 Walks in a Graph. In many problems using graphs, "walking" through vertices and edges is the main goal. We start with the most general notion, the one that only requires that a next edge starts in the vertex where the previous edge terminates. In directed graphs we distinguish between directed and undirected walks. The difference is, roughly speaking, that in directed walks we must go from the initial vertex of an edge to its terminal vertex (so in the direction of the edge). In an undirected walk, we can go even against the direction of the edge, i.e. from the terminal vertex to the initial vertex of an edge.

Definition. Given a directed graph G. A *directed walk* in G is a sequence of vertices and edges

$$v_1, e_1, v_2, e_2, \ldots, v_{k-1}, e_{k-1}, v_k$$

such that for every $i = 1, 2, \ldots, k-1$ it holds that $v_i = IV(e_i)$ and $v_{i+1} = TV(e_i)$.

An *undirected walk* in a directed or undirected graph G is a sequence of vertices and edges

$$v_1, e_1, v_2, e_2, \ldots, v_{k-1}, e_{k-1}, v_k$$

such that for every $i = 1, 2, \ldots, k-1$ vertices v_i and v_{i+1} are end vertices of e_i.

Given a directed (or undirected) walk. We say that v_1 is the *initial vertex* of the walk, and v_k is the *terminal vertex* of the walk. Also we say that the *walk goes from v_1 to v_k*. □

Remark: We define an undirected walk also in an undirected graph because the definition is the same. Note that the notion of a directed walk in undirected graphs is meaningless.

7.1.7 A Trivial Walk. A walk is defined as a sequence of vertices and edges with some additional properties. The sequence may contain only one vertex and no edge. In that case it is called trivial.

Definition. A *trivial walk* is a walk that contains only one vertex and no edge. □

We consider it as a directed and also an undirected walk.

7.1.8 Closed Walks. Roughly speaking, a walk (directed or undirected) is closed if the initial vertex of it equals the terminal one and the walk contains at least one edge.

Definition. A directed (an undirected) walk is called *closed* if $k > 1$ and $v_1 = v_k$. Otherwise it is called an *open* walk. □

Hence, a trivial walk is not closed; indeed, $k = 1$.

7.1.9 Trails and Paths, Cycles and Circuits. In 7.1.6 we defined the notion of a walk. There are special types of walks that play an important role in applications. These are trials and their special cases paths.

Definition. A directed (an undirected) walk $v_1, e_1, \dots, e_{k-1}, v_k$ is called a *directed trail* (an *undirected trail*) if it contains every edges at most once. In other words, for $i \neq j$ it holds that $e_i \neq e_j$.

A directed (or an undirected) trail $v_1, e_1, \dots, e_{k-1}, v_k$ is called a *directed path* (an *undirected path*) if it contains every vertex at most once with the exception that v_1 may be the same as v_k.

A *cycle* is a closed directed path, i.e. a directed path with $v_1 = v_k$, $k > 1$.

A *circuit* is a closed undirected path, i.e. i.e. an undirected path with $v_1 = v_k$, $k > 1$. □

7.1.10 Remarks. A cycle (a circuit) can also be defined as an open directed (undirected) path together with one edge from the terminal vertex to the initial vertex (between initial and terminal vertices) of the path.

Every path is a trail, and every trail is a walk; the opposite does not hold. Also every cycle is a circuit, but not every circuit in a directed graph is a cycle.

Notice, that a trivial walk is a trail and a path, but it is neither a circuit nor a cycle.

7.1.11 Reachability.

Definition. Given a directed or undirected graph $G = (V, E, \varepsilon)$. We say that a vertex v is *reachable* from a vertex w if there exists an undirected path from w to v. □

Remarks. 1. In the definition above we could require the existence of a walk from w to v and we would get the same notion. Indeed, any path is a walk; on the other hand, any walk from w to v contains a path form w to v:

Assume that $P = v_1, e_1, v_2, \dots, v_{k-1}, e_{k-1}, v_k$ is a walk which is not a path. Let $v_i = v_j$ for $i < j$. Then

$$P_1 = v_1, e_1, \dots, v_i, e_j, \dots v_k$$

is a walk from v_1 to v_k which is shorter. (Roughly speaking, we cut off one closed walk of P.) If P_1 is a path, we are done. If not, we again find $r \neq s$ such that $v_r = v_s$ and we omit the walk from v_r to v_s. Since any walk cannot have less than 0 edges, the procedure must end, and we are left with a path from v_1 to v_k.

2. The relation of reachability is reflexive; indeed, v is reachable from itself by the trivial path v.

3. The relation of reachability is symmetric; it means that if v is reachable from w then so is w from v. The reason is that any path from w to v can be viewed as a path from v to w. Indeed, if $w, e_1, \dots, e_{k-1}, v$ is an undirected path, then so is $v, e_{k-1}, \dots, e_1, w$.

4. Note that the relation of reachability is transitive; it means if v is reachable form w and u is reachable from v, then u is reachable from w. If we join a path from w to v and a path from v to u, we get a walk from w to u. And the walk contains a path with the same initial and terminal vertices.

7.1.12 Connected Graphs.

Definition. A graph G (directed or undirected) is called *connected* if for every two vertices u, v of G there exists an undirected path form u to v (i.e. every vertex is reachable from any vertex).

If a graph is not connected then it is called *disconnected.* □

7.2 Trees

Trees form a class of undirected (or directed) graphs that one will come across in many applications, not only in computer science but also in many engineering applications. Even in this course we have spoken about syntactic trees for propositional or predicate formulas.

7.2.1 Definition. A graph G (directed or undirected) is called a *tree* if it is connected and it does not contain a circuit. □

7.2.2 Trees satisfy an interesting property – every tree with n vertices has $n-1$ edges. To show this we need the following lemma.

Lemma. Let G be a tree with at least two vertices. Then G contains at least one vertex with degree 1. □

Justification: We proceed by contradiction: Assume that G does not have a vertex v with $d(v)=1$, so $d(u)\geq 2$ for any vertex u. Let us from a walk as follows:

Start in any vertex and denote it by v_1. Since $d(v_1)\geq 2$, there is an edge, say e_1, incident with v_1. Denote by v_2 its second end vertex. Necessarily $v_2\neq v_1$. Indeed, otherwise e_1 is a loop and every loop is a circuit. Since $d(v_2)\geq 2$ there is an edge e_2, $e_2\neq e_1$, incident with v_2. Denote by v_3 the other end vertex of e_2. Since G does not contain a circuit, v_3 is a new vertex of degree at least 2, so we continue. In that way we obtain a path $v_1, e_1, v_2, \ldots, e_{n-1}, v_n$ where n is the number of vertices. But $d(v_n)\geq 2$, hence there exists an edge different form e_{n-1}, say e_n, incident to v_n. The other end vertex of e_n must be one of $v_1, v_2, \ldots, v_n$ (indeed, we do not have any other vertex), therefore e_n closes a circuit – a contradiction with the fact that G is without circuits. □

7.2.3 Theorem. Every tree with n vertices has precisely $n-1$ edges. □

Justification: We proceed by mathematical induction:

Basic step. If $n=1$ there is only one tree – a vertex with no edge. If $n=2$ there is again only one tree – two vertices joint by one edge. Hence for $n=1$ or $n=2$ the assertion is true.

Inductive step. Assume that every tree with n vertices has $n-1$ edges. Consider any tree G with $n+1$ vertices. From the lemma above, we know that G contains a vertex v with $d_G(v)=1$. Let us remove v and the one edge incident to v from the graph G. We obtain a graph G' which is connected and has no circuit, so G' is a tree with n vertices. By the induction assumption, G' has $n-1$ edges, hence G has $n-1+1=n$ edges. The proof is complete. □

7.2.4 Proposition. Every tree with at least two vertices contains at least two vertices of degree 1. □

Justification. There are two different (and easy) proofs.

1. We know that $\sum_{v\in V} d(v) = 2\,|E|$ (see 7.1.5), and hence $\sum_{v\in V} d(v) = 2\,(n-1)$, where $n\geq 2$ is the number of vertices. If there was only one vertex v with $d(v)=1$, then $\sum_{v\in V} d(v)\geq 1+2(n-1)$; a contradiction.

2. There is also a direct proof which does not use 7.2.3. Consider a maximal path in the tree G, i.e. a path that is not contained in any longer path. Denote the path by $P=u_1, e_1, \ldots, e_{k-1}, u_k$. Then $d(u_1)=1=d(u_k)$. Indeed, if $d(u_1)\geq 2$ then either the path P is not maximal, or there is a circuit containing u_1. Similarly for u_k. □

7.2.5 Several Characterizations of Trees. The following theorem gives two other characterizations of trees.

Theorem. Given a graph G. Then the following are equivalent:

1. G is a tree.

2. G contains no circuit and if we add any new edge (the set of vertices remains the same), then we close just one circuit.
3. G is connected and removing any edge disconnects G.

□

Justification. 1. implies 2. Assume that G is a tree. Then from the definition, G does not contain a circuit. Assume that we add a new edge e between u and v. Since G is connected, there exists a path P from u to v. Now, the path P together with e (e does not belong to P) forms a circuit. Hence e has closed a circuit.

Assume that e has closed two circuits, say C_1 and C_2. Then there are two distinct paths P_1 and P_2 from u to v. If we join P_1 and P_2 we get a new circuit in this case belonging to G – a contradiction with the fact that G does not have a circuit.

2. implies 3. Assume that G contains no circuit and adding any new edge closes just one circuit. Then G is connected; indeed, assume that there was no path between u and v; in this case, adding a new edge with end vertices u and v does not close a circuit.

Let us remove an edge e with end vertices x and y. If G stays connected then there is a path P in G from x to y which does not contain e. Hence, P together with e forms a circuit – a contradiction with the fact that G does not contain a circuit. So removing any edge disconnects the graph G.

3. implies 1. Assume that G is connected and removing any of its edges disconnects it. We have to show that G is a tree, i.e. it is connected and does not contain a circuit. Assume for contrary that G contained a circuit. Take any edge e from the circuit (it must exist since any circuit contains at least one edge). Then removing e from G does not disconnect it – a contradiction. □

7.2.6 Remarks. Given any connected graph G. If we add one edge (without adding a new vertex) to the set of edges of G, then the new graph remains connected.

If a graph G contains no circuit and we remove one edge, then the new graph will also contain no circuit.

A tree is a graph tha t has the smallest number of edges to be connected and the biggest number of edges to be without circuits.

7.2.7 Subgraphs. Roughly speaking, a subgraph of a given graph is obtained by: forgetting some (maybe none) vertices, forgetting some (maybe none, maybe all) edges, but if an edge is in the subgraph then also both its end vertices are in the subgraph. More precisely:

Definition. Given a graph $G = (V, E, \varepsilon)$. A *subgraph* of G is a triple $G' = (V', E', \varepsilon')$ where

- $V' \subseteq V$,
- $E' \subseteq E$, and
- ε' is the restriction of ε on the set E'.

□

We distinguish two special types of subgraphs:

Definition. Given a graph $G = (V, E, \varepsilon)$.

- A *factor* is a subgraph where $V' = V$, i.e. it contains all vertices of G.
- Let $A \subseteq V$. The *subgraph induced by* A is the subgraph with $V' = A$ and $e \in E$" if and only if $e \in E(G)$ and the end vertices of e belong to A.

□

Notice, that the subgraph induced by A is the "maximum" subgraph with the set of vertices A.

7.2.8 Components of Connectivity. Since not every graph is connected, it is useful to "divide" the set of vertices into parts that are connected. Roughly speaking, a component of connectivity is a maximal part of the graph "which is connected". More precisely:

Definition. Given a graph G (directed or undirected). A *component of connectivity* (sometimes also called a *component of weak connectivity*) is a maximal subset A of $V(G)$ such that the subgraph induced by A is connected. □

By a maximal subset we mean that the subgraph induced by A is connected but for every $v \in V(G)$, $v \notin A$, the subgraph induced by $A \cup \{v\}$ is not connected.

Remarks.

1. We can define the components of connectivity also in a different way. Given a graph G with the set of vertices V. Define a relation R on V to be the reachability relation, i.e. by

$$u\,R\,v \ \text{ if and only if } \ v \text{ is reachable from } u.$$

We know from 7.1.11 that R is an equivalence relation on V. Components of connectivity are now the classes of the equivalence R.

2. A graph is connected if and only if it has only one component of connectivity.

7.3 Spanning trees

We know that trees are connected graphs with the minimal number of edges. Hence trees become very useful in applications where our goal is to connect some places using the least number of connections. Let us form an undirected graph as follows: Vertices are places, edges are connections between them. Now, we are looking for a subgraph which is a factor (we need to connect **all** places), and which is a tree – but at the same time a factor of the given graph – such subgraphs will be called a spanning tree.

7.3.1 Definition. Given a connected graph G. A factor of G which is a tree is called a *spanning tree* of G. □

The following proposition characterizes graphs that have a spanning tree.

Proposition. A graph G has a spanning tree if and only it it is connected.

Justification. If a graph has a spanning tree it must be connected because a spanning tree is a connected graph.

Assume that G is connected. If G does not have a circuit then it is itself a tree, so it is its (only) spanning tree. Assume that G has a circuit, say C. Let us remove one edge of C from G. We obtain a subgraph of G, say G_1 which is connected (we removed an edge from a circuit). Therefore if G_1 does not have a circuit, then it is a tree and also a spanning tree of G. If G_1 contains a circuit we proceed in a similar way: we remove one edge of an existing circuit. After removing a finite number of edges (in fact $|E| - (|V| - 1)$) we get a connected graph with $|V| - 1$ edges, and it has to be a tree. So it is a spanning tree of G.

7.3.2 A Minimal Spanning Tree. In some applications, we know the price of an edge (e.g. the price which has to be paid for construction of a given connection). In this case we are not interested in an arbitrary spanning tree but in a spanning tree that has the least sum of prices of chosen edges. And this is, roughly speaking, a minimal spanning tree.

Definition. Given a connected graph G together with a mapping c which assigns to every $e \in E(G)$ a number $c(e)$.

A *minimal spanning tree* of $G = (V, E)$ is a spanning tree $K = (V, L)$ such that $\sum_{e \in L} c(e)$ is the smallest one (among all spanning trees of G). □

Let us mention that the number $c(e)$ is often called the *weight of* e or the *price of* e. Also, a graph G together with a mapping $c\colon E(G) \to \mathbb{R}$ is called a *weighted graph.* We will denote a weighted graph by (G, c).

Proposition. Any connected weighted graph (G, c) has a minimal spanning tree (which is not necessary unique). □

Justification. We already know that any connected graph has a spanning tree. On the other hand, there are only finitely many spanning trees (as $n-1$ element subsets of a finite set of edges). So there must be a spanning tree with the least possible weight. □

A minimal spanning tree does not need to be unique; indeed, take any connected graph G with n vertices and n edges ($n > 2$), and define $c(e) = 1$ for every edge e. Then any spanning tree of G is a minimal one, and there are at least two spanning trees. (Note that a graph with n vertices and n edges cannot be a tree itself.)

7.3.3 An Algorithm for Finding a Minimal Spanning Tree. There are several algorithms that find a minimal spanning tree for a given connected weighted graph (G, c). We will show only one of them, the Kruskal's algorithm. It is an examples of so called “greedy algorithms”, in other words, algorithms that in each step choose the most “promising” edge.

Kruskal's Algorithm for Finding a Minimal Spanning Tree.

Input: A connected graph G with the weight function c.
Output: A minimal spanning tree of G represented by its set of edges L.

1. Sort edges by their weights into non decreasing sequence, i.e.

   $$c(e_1) \le c(e_2) \le \ldots \le c(e_m).$$

   Put $L := \emptyset$.

2. Go through edges in the given order. Insert e_i into L if and only if it does not close a circuit in L. If e_i closes a circuit, skip e_i and continue with e_{i+1}.

3. If L has $n-1$ edges (n is the number of vertices of G) end the algorithm, (V, L) is a minimal spanning tree.

□

Remarks.

1. If the input is a disconnected graph then the above algorithm will end with $|L| < |V|-1$ edges. So we do not need to check beforehand whether a given graph is connected.

2. If several edges have the same weight then the ordering in the first step of the algorithm “determines” the minimal spanning tree which the algorithm finds.

7.3.4 Example. Given an undirected graph $G = (V, E)$ with $V = \{1, \ldots, 7\}$ by the following matrix of weights (it means, at the position (i, j) we have either $c(\{i, j\})$ if $\{i, j\} \in E$, or "$-$" if $\{i, j\} \notin E$). Find a minimal spanning tree in (G, c) using the Kruskal's algorithm

$$\begin{pmatrix}
- & 6 & 9 & - & - & - & 9 \\
6 & - & 2 & 1 & 3 & - & - \\
9 & 2 & - & 1 & - & - & 15 \\
- & 1 & 1 & - & 10 & 13 & 3 \\
- & 3 & - & 10 & - & 10 & 1 \\
- & - & - & 13 & 10 & - & 15 \\
9 & - & 15 & 3 & 1 & 15 & -
\end{pmatrix}$$

Solution. First we sort the edges (we state the weight of an edge in the brackets)

$$e_1 = \{2,4\}(1), e_2 = \{3,4\}(1), e_3 = \{5,7\}(1), e_4 = \{2,3\}(2), e_5 = \{2,5\}(3), e_6 = \{4,7\}(3),$$

$$e_7 = \{1,2\}(6), e_8 = \{1,3\}(9), e_9 = \{1,7\}(9), e_{10} = \{4,5\}(10), e_{11} = \{5,6\}(10), e_{12} = \{4,6\}(13),$$

$$e_{13} = \{3,7\}(15), e_{14} = \{6,7\}(15).$$

Put $L = \emptyset$.

Now we will go through edges in the given order and include e_i into L if and only if it does not close a circuit.

1. e_1 does not close a circuit, hence $L := \{\{2,4\}\}$.
2. e_2 does not close a circuit, hence $L := \{\{2,4\},\{3,4\}\}$.
3. e_3 does not close a circuit, hence $L := \{\{2,4\},\{3,4\},\{5,7\}\}$.
4. e_4 closes a circuit formed by e_1, e_2 and e_4, hence L is the same as in 3.
5. e_5 does not close a circuit, hence $L := \{\{2,4\},\{3,4\},\{5,7\},\{2,5\}\}$.
6. e_6 closes a circuit formed by e_1, e_5, e_3 and e_6, hence L is the same as in 5.
7. e_7 does not close a circuit, hence $L := \{\{2,4\},\{3,4\},\{5,7\},\{2,5\},\{1,2\}\}$.
8. e_8 closes a circuit formed by e_1, e_2, e_7 and e_8, hence L is the same as in 7.
9. e_9 closes a circuit formed by e_3, e_5, e_7 and e_9, hence L is the same as in 7.
10. e_{10} closes a circuit formed by e_1, e_5 and e_{10}, hence L is the same as in 7.
11. e_{11} does not close a circuit, hence $L := \{\{2,4\},\{3,4\},\{5,7\},\{2,5\},\{1,2\},\{5,6\}\}$.

Since L contains $7 - 1 = 6$ edges, therefore

$$L = \{\{2,4\},\{3,4\},\{5,7\},\{2,5\},\{1,2\},\{5,6\}\}$$

is the set of edges of a minimal spanning tree of G. The weight (price) of L is

$$c(L) = 1 + 1 + 1 + 3 + 6 + 10 = 22.$$

7.4 Directed Trees

In this section we will introduce the notion of a rooted tree – a directed tree that contains a vertex r from which any other vertex is reachable by a directed path.

7.4.1 A Root.

Definition. Given a directed graph $G = (V, E, \varepsilon)$. We say that a vertex $r \in V$ is a *root* of G if there exists a directed path from r to any vertex of G. □

Note that in the definition of a root we can require existence of a directed walk instead of a directed path. Let us also note that a directed graph can have more than one root; indeed, in a cycle every vertex is a root. There are also directed graphs that do not have a root – find an example.

7.4.2 A Rooted Tree. A directed graph with a root which, at the same time, is a tree plays an important role in applications. For example, data structures based on rooted trees are widely used in computer science.

Definition. A directed graph which is a tree and contains a root is called a *rooted tree.* □

Since every graph with a root is connected (the opposite does not hold), we could define a rooted tree as a directed graph with a root that does not contain a circuit.

7.4.3 Proposition. Every rooted tree contains precisely one root. □

Justification. Assume, in contrary, that a rooted tree has roots r_1 and r_2, $r_1 \neq r_2$. Then there exists a directed path P_1 from r_1 to r_2 (indeed, r_1 is a root), as well as a directed path P_2 from r_2 to r_1 (indeed, r_2 is a root). Moreover, P_1 together with P_2 form a closed walk, and every closed walk contains at least one circuit. And this contradicts the fact that a tree does not contain a circuit. □

7.4.4 Remark. Given a rooted tree $G = (V, E)$ with its root r. Then for every vertex $v \in V$ there is a unique directed path from r to v. Indeed, there is a directed path form r to v by definition, and if there are two different directed paths then there is a closed undirected walk that always contains a circuit, which contradicts the fact that G is without circuits.

7.4.5 Successor, Predecessor, Leaf. We can distinguish different "types" of vertices in a rooted tree.

Definition. Let $G = (V, E)$ be a rooted tree. If (u, v) is an edge of G then u is called *predecessor* of v, and v is called a *successor* of u. A vertex which does not have a successor is called a *leaf.* □

Note that any leaf must have in-degree 1 and out-degree 0, so $d(v) = 1$ for any leaf. The other implication does not hold. A root of a tree may also have degree 1; but in this case it is the out-degree.

7.4.6 Levels, the Hight of a Rooted Tree. Vertices of a rooted tree can be divided into so called levels according to the number of edges that the only directed path from the root to the vertex has.

Definition. Given a rooted tree $G = (V, E)$ with the root r. A vertex $v \in V$ *belongs to k-th level* if the unique directed path from r to v contains precisely k edges.

The *hight of a directed tree* is the biggest k such that there is a vertex in k-th level. □

Remarks. 1. The hight of a rooted tree is, in fact, the number of edges in the longest directed path from r (to a leaf).

2. Notice that the root itself forms the 0-th level. Indeed, the only directed path from r to r is the trivial path containing 0 edges.

7.4.7 A Subtree Determined by a Vertex.

Definition. Given a rooted tree G. A *subtree generated by* a vertex v is the subgraph of G induced by the set of all vertices directly reachable from v. □

Remark. It is easy to see that a subtree generated by v is again a rooted tree, its root is v.

7.4.8 Binary Rooted Trees. Rooted trees where each vertex has at most two successors play an important role in data structures.

Definition. A rooted tree is called a *binary rooted tree* if every vertex has at most two successors.

In binary rooted trees we speak about the *right successor* and the *left successor* of a vertex v. A *right subtree*, a *left subtree*, of v is the subtree generated by the right successor, or the left successor of v, respectively. □

An example of a binary rooted tree is a data structure called a heap. It is a basis of the heap sort, one of the fast sorting algorithms.

7.5 Acyclic Graphs

Trees are connected graphs without circuits. Acyclic graphs are directed graphs that do not contain a cycle. Unlike trees, acyclic graphs do not need to be connected.

7.5.1 Definition. A directed graph is called *acyclic* if it does not contain a cycle. □

We will give another characterization of acyclic graphs; acyclic graphs are those directed graphs that admit a topological sort of vertices or/and a topological sort of edges.

7.5.2 Topological Sort of Vertices.

Definition. Given a directed graph $G = (V, E, \varepsilon)$ with n vertices. A sequence of vertices

$$v_1, v_2, \dots, v_n$$

is called a *topological sort*, also *topological ordering*, if the following holds: for every edge e with initial vertex v_i and terminal vertex v_j it necessarily holds that $i < j$. □

Informally, edges go from a vertex with a smaller index to a vertex with a bigger index.

7.5.3 Topological Sort of Edges.

Definition. Given a directed graph $G = (V, E, \varepsilon)$ with m edges. A sequence of edges

$$e_1, e_2, \dots, e_m$$

is called a *topological sort*, also *topological ordering*, if for every two edges e_i, e_j for which the terminal vertex of e_i is the initial vertex of e_j it necessarily holds that $i < j$. □

7.5.4 Proposition. In every acyclic graph there exists at least one vertex with in-degree 0. □

Justification. We proceed by contradiction. Assume that every vertex of a graph G has its in-degree at least 1. Choose any vertex v, and denote it by v_1. Since $d^-(v_1) \geq 1$, there is an edge e_1 with $TV(e_1) = v_1$. Denote by v_2 the vertex $IV(e_1)$. We have $v_2 \neq v_1$; indeed, otherwise e_1 is a cycle. Since $d^-(v_2) \geq 1$, there is an edge e_2 with $TV(e_2) = v_2$. Denote v_3 the initial vertex of e_2. Then v_3 is a new vertex; indeed, if $v_3 = v_1$ or $v_3 = v_2$ then G contains a cycle. Again, $d^-(v_3) \geq 1$, hence there is e_3 such that $TV(e_3) = v_3$, and $v_4 = IV(e_3)$ must be a new vertex, otherwise G contains a cycle.

In such a way, we get a directed path $v_n, e_{n-1}, v_{n-1}, e_{n-2}, \dots, v_2, e_1, v_1$ containing all vertices of G. But $d^-(v_n) \geq 1$, so there must be an edge e_n with $TV(e_n) = v_n$. On the other hand, $IV(e_n)$ must be among vertices $v_1, \dots, v_n$ (indeed, we do not have more vertices); therefore G contains a cycle – a contradiction.

7.5.5 Theorem. Given a directed graph G. Then the following conditions are equivalent:

1. G is an acyclic graph;
2. G has a topological sort of vertices;
3. G has a topological sort of edges.

Justification. First, we show that 2 implies 3. Assume that $v_1, v_2, \dots, v_n$ is a topological sort of vertices of G. We list at first all edges with the initial vertex v_1, then all edges with the initial vertex v_2, etc. and finally all edges with the initial vertex v_n. This sequence contains all edges, indeed, every edge has its initial vertex, hence was listed. The fact that it is a topological sort of edges of G is evident.

3 implies 2: Let $e_1, e_2, \dots, e_m$ be a topological sort of edges of G. We go through edges in this order and we list the initial vertex of the edge if it has not been listed before. In this way we get a sequence $v_1, v_2, \dots, v_k$ of (not all) vertices of G. We add, in an arbitrary order, the remaining vertices. The sequence $v_1, v_2, \dots, v_k, v_{k+1}, \dots, v_n$ is a topological sort of vertices of G.

It is easy to notice that if a graph G is not acyclic then it does not have a topological sort of vertices; indeed, a cycle cannot be topologically ordered. The fact that any acyclic graph has a topological sort of vertices will be shown by an algorithm bellow that constructs a topological sort of vertices.

7.5.6 An Algorithm for Topological Sort. The algorithm is based on the proposition 7.5.4. First, we find all vertices with in-degree 0, since any such vertex v can be the first vertex of a topological sort of vertices. Then we remove all edges e with the initial vertex v. This will be done by decreasing the in-degree of $TV(e)$ by 1. If the new in-degree of $TV(e)$ becomes 0, then $TV(e)$ may be inserted in the topological sort of G. The set M will always contain all vertices with the actual in-degree 0; i.e. vertices that can be listed in a topological sort of G.

More precisely:

An Algorithm for Finding a Topological Sort of Vertices.

Input: An acyclic graph G.
Output: A topological sort of vertices of G.

```
1) For each vertex $v$ we calculate its in-degree $d^-(v)$.
2) The set $M$ contains all vertices with in-degree 0.
   We put $i := 1$.
3) While $M \neq \emptyset$ we do
   3a) We choose a vertex $v$ from $M$ and delete it from $M$.
       We put $v_i := v$, $i := i + 1$.
   3b) For every edge $e$ with $IV(e) = v$ we do
       $d^-(TV(e)) := d^-(TV(e)) - 1$. If $d^-(TV(e)) = 0$ we insert $TV(e)$ into the set $M$.
```

7.6 Strong Connectivity

7.6.1 Strongly Connected Graphs. We have defined connected directed graphs as directed graphs where any two vertices are joined by an undirected path. Now, we introduce the notion of strong connectivity where we require that any two vertices are joined by a **directed** path.

Definition. Given a directed graph G. We say that G is *strongly connected* if for any two vertices u, v there exists a directed path form u to v and a directed path from v to u. □

Remark. In the definition of a strongly connected graph we could, for any two vertices u and v, require only the existence of a directed path from u to v. Indeed, we require a directed path for **every** pair of vertices u, v, hence also for the pair v, u.

Notice that there always is a directed path from a vertex v to itself; indeed, it is the trivial path.

7.6.2 The following proposition gives a condition which guarantees that a connected directed graph is strongly connected.

Proposition. A connected directed graph is strongly connected if and only if every its edge is contained in a cycle. □

Justification: Assume that G is strongly connected and take an edge e, say from u to v. Since G is strongly connected, there exists a directed path P from v to u. If we add the edge e to P we obtain a cycle. So we have shown that e is contained in a cycle.

Assume that a directed graph G is connected and every its edge belongs to some cycle. We will show that for every $u, v \in V$ there is a directed path from u to v.

Take two vertices u and v. We know that G is a connected graph, hence there is an undirected path, say P, from u to v. If it is a directed path, we are done. Assume that in P there is an edge e for which P goes from the terminal vertex y of e to the initial vertex x of e. Denote by C a cycle in G containing e. Denote by P_e the cycle C without e. Then P_e

is a directed path from x to y. Replace the edge e by P_e in P . We do the procedure above for every edge e of P such that the terminal vertex of e precedes the initial vertex of e. In such a way, we get a directed walk from u to v. Now, any directed walk from u to v contains a directed path from u to v. We have shown that for every pair u and v there is a directed path from u to v. Hence G is strongly connected. □

7.6.3 Components of Strong Connectivity. If a directed graph is not strongly connected we can ask about maximal subsets for which the induced subgraph is strongly connected. Such maximal sets of vertices are called strongly connected components.

Definition. Given a directed graph G. A set K of vertices is called a *strongly connected component*, also a *component of strong connectivity*, if the subgraph induced by K is strongly connected and it is maximal with this property (i.e. if we add to K any other vertex then the induced subgraph is not strongly connected). □

Remark. Every vertex of a directed graph G is contained in exactly one strongly connected component of G. The same does not hold for edges; if a graph is not strongly connected then it can contain edges that belong to no strongly connected component. Indeed, edges between two distinct components of strong connectivity.

7.6.4 Condensation of a Graph. The structure of strongly connected components of a directed graph is captured by so called condensation of a directed graph. For strongly connected graphs the condensation is a trivial graph which has one vertex and no edge.

Definition. Given a directed graph $G = (V, E)$. The *condensation of* G is a directed graph $\bar{G} = (\bar{V}, \bar{E})$ where $\bar{V}$ is the set of all strongly connected components of G, and there is an edge from a component K_1 to a component K_2 if and only if $K_1 \neq K_2$ and there exist vertices $u \in K_1$, $v \in K_2$ such that (u, v) is an edge of G. □

Remark. Note the the condensation of a directed graph is always an acyclic graph.

7.7 Euler graphs

In the section 7.1, we promised to give the theoretical background for the Euler's solution of the problem of seven bridges. Before that, let us recall that a trail in a graph is a walk where every edge is used at most once.

7.7.1 Euler Trails, Euler Graphs.

Definition. Given a directed graph G. A directed (an undirected) trail in G is called an *Euler trail* if it contains all edges.

Given an undirected graph G. An undirected trail in G is called an *Euler trail* if it contains all edges. □

Note that Euler trail can be open or closed. It is easy to notice that if a graph has an open Euler trail (a closed Euler trail) then it cannot have a closed (an open) one.

Definition. A directed (undirected) graph G is called an *Euler graph* if it contains a closed directed (undirected, respectively) Euler trail. □

7.7.2 Applications. Let us give two applications of Euler trails.

- **Drawing graphs with the smallest possible number of trails.** Given a connected undirected graph G. The task is to find the least possible number of edge disjunctive trails such that every edge of G is contained in some trail. It is evident that if G contains an Euler trail then every Euler trail is a solution. If in G there is no Euler trail then at least two trails are necessary to cover every edge of G.

Solutions of this problem could be used for example if we draw a graph using a computer and we want to minimize the number of ”shifts of the drawing pen”.

- **Chinese Postman Problem.** A postman has to go along every street. How to do it if he wants to minimize the number of kilometers he has to walk?

  Let us construct a graph $G = (V, E)$ where V is the set of all crossings, and E is the set of streets which a postman must walk along. If in G, there exists an Euler closed trail then it is an optimal solution; indeed, in this case the postman will go along each street exactly once.

  If there is no closed Euler trail, the postman will have to go twice along some streets. To find a solution which will minimize the number of kilometers the postman must walk, we need the information how long each street is. So for every edge e a positive number $c(e)$ is given – its Length. We add parallel edges to the graph G in such a way that the resulting graph will contain a closed Euler trail and the sum of lengths of added edges will be the smallest possible.

7.7.3 Proposition. Given a connected directed graph G with at least two vertices. Then G contains a closed directed Euler trail if and only if for every vertex v of G the following holds

$$d^-(v) = d^+(v).$$

(In other words, the number of edges that terminate in v is the same as the number of edges that start in v.)

In a connected graph G there is a closed undirected Euler trail if and only if each vertex has an even degree. □

Justification. We give the proof only for directed graphs; for undirected graphs the proof is similar and easier.

It is evident that if a graph G has a closed directed Euler trail then it should satisfy the condition that $d^+(v) = d^-(v)$ for any vertex v. Indeed, if we ”go” along such a closed trail we can match the edge along which we enter v with the edge along which we leave v. Hence there must be the same number of edges ”going into v” as is the number of edged that ”go out of v”.

The other implication follows from an algorithm which constructs a closed directed Euler trail for a connected directed graph that satisfy the condition above. □

7.7.4 A Procedure for Finding a Closed Directed Euler Trail. Let G be a graph satisfying the condition from 7.7.3 with at least 2 vertices. We choose an arbitrary vertex v of G. Since G is connected and satisfies the condition above, for every vertex v there is at least one edge that starts in v and at least one edge that terminates in v.

We randomly form a directed trail starting in v as follows: We put $v_1 := v$, we go along an edge e_1 with the initial vertex v_1. Denote by v_2 the terminal vertex of e_1. Since $d^-(v_2) = d^+(v_2)$, there is an edge e_2 with the initial vertex v_2 that has not been used yet, we go along e_2 into its terminal vertex v_3. We continue in this manner till it is possible. The only situation when the trail cannot be extended is when we return to $v = v_1$ and every edge with the initial vertex v has already been used. Hence, we have randomly constructed a closed directed trail T.

If the trail T contains all edges of G we are ready; it is a closed directed Euler trail.

If the closed directed trail T does not contain all edges there exists an edge e which is not contained in T. Denote by x the initial vertex of e. Since G is connected, there exists an undirected path from v to x. This path must start with some edge incident to v and so contained in T. Hence there must be a vertex w where the undirected path from v to x leaves the trail T. Then it means that there is an edge incident to w that is not included in T.

Since $d^-(w) = d^+(w)$, there is also an edge e' with the initial vertex w and not contained in T. Let us randomly form a maximal closed directed trail T_1 starting in w formed by edges that are not contained in T. Due to the condition in the proposition, T_1 must end in w. We disconnect T in w and insert T_1. So we get a new closed directed trail, say T'. If T' contains all edges, T' is a closed directed Euler trail. If not, there must be a vertex w' on T' and some edge incident to w' which is not contained in T' and we repeat the procedure.

Since G has only a finite number of edges, we must end up with a closed directed Euler trail. □

7.7.5 Open Euler Trials. Now, we state a characterization of connected graphs (directed and undirected) that contain an open Euler trial.

Proposition. Given a connected directed graph G. Then G contains an open directed Euler trail if and only if there exist two vertices u_1, u_2 such that

$$d^-(u_1) = d^+(u_1) - 1, \;\; d^-(u_2) = d^+(u_2) + 1,$$

and for every other vertex v of G it holds that $d^-(v) = d^+(v)$.

In a connected graph G there exists an open undirected Euler trail if and only if there are two vertices with odd degree and all other vertices have even degree. □

Justification. Again, we prove the proposition for directed Euler trials only. The proof for undirected Euler trials is analogous.

It is easy to see that if there is an open Euler trail in G then G satisfies the condition from the proposition; indeed, u_1 is the vertex where the Euler trail starts, and u_2 where it terminates.

To construct an open Euler trail we proceed similarly as in 7.7.4. We start in u_1 and randomly construct a maximal trail T. Because of the conditions, T must terminate in u_2. If T contains all edges, it is an open Euler trail. If not, there must be a vertex w, $w \neq u_1$, $w \neq u_2$, where no all edges incident with w were used. We insert into T a maximal trail from w to w containing edges not from T. After a finite number of inserting closed trails, we get an open Euler trail. □

7.8 Hamiltonian graphs

Recall that a path is a trail where vertices are not repeated, only in a closed path (a circuit if undirected, or a cycle if directed) initial and terminal vertices are the same.

7.8.1 Hamiltonian paths, circuits, and cycles. Another type of important walks in a graph are paths (closed or open) that contain all vertices. Such paths/circuits/cycles will be called Hamiltonian paths/circuits/cycles. More precisely:

Definition. Given a graph G. An open path is called a *Hamiltonian path* if it contains all vertices of G (and hence every vertex precisely once).

Similarly, a *Hamiltonian circuit* is a circuit that contains every vertex of the graph; a *Hamiltonian cycle* is a cycle that contains every vertex of the graph. □

7.8.2 Problems concerning Hamiltonian paths can be divided into two groups — existential and optimization problems. In existential problems we want to find out whether a given graph contains a Hamiltonian path (circuit, cycle, respectively). In optimization problems every edges of a graph has its *weight* (or *length*) which is an integer. In this case, the problem is to find a Hamiltonian path (circuit, cycle, respectively) with optimal sum of weights (lengths) of its edges.

Unlike problems where we have to find Euler trails, the problems concerning Hamiltonian paths, circuits, and/or cycles are rather difficult. A fast algorithm is not known that would solve either existential or optimization problems concerning Hamiltonian paths. Despite of this fact (and maybe due to this fact), the problems of Hamiltonian paths have lot of applications. In the next paragraph, we state some of them.

7.8.3 Applications. Let us give two applications of the problem of Hamiltonian paths.

- **Traveling Salesman Problem.** The original problem is the following: Given n towns, we denote them $1, 2, \ldots, n$. For every pair of distinct towns i, j their distance $d(i, j)$ is given. The task is to find a succession of towns in which a salesman can visit the towns so that the total distance traveled is the smallest possible one.

  The problem can be reformulated as a graph problem: Given a complete graph G with the set of vertices $V = \{1, 2, \ldots, n\}$, i.e. for every two distinct vertices $i \neq j$ there is an edge $\{i, j\}$ in G. For every $\{i, j\}$, we define its length to be $d(i, j)$. The goal is to find a Hamiltonian circuit C for which
  $$\sum_{e \in C} d(e)$$
  is the smallest possible.

- **Planning of processes.** Assume the following situation: we have a device on which processes $p_1, p_2, \ldots, p_n$ are carried out. Moreover, there are pairs of processes p_i, p_j such that if p_i should follow p_j it is necessary to clean, convert, etc. the device. So in this case, it is necessary to pay some price in order to realize p_j just after finishing p_i.

  The task is to find a sequence of processes such that the sum of prices for its realization is zero. If such sequence does not exist, the problem is to find a sequence where the sum of prices is as small as possible.

  This second case leads to the Traveling Salesman Problem.

7.8.4 There are easy necessary conditions for a graph to have a Hamiltonian path (circuit, cycle, respectively). Let us state some of them.

- If in a graph G there is a Hamiltonian path then G must be connected. (Indeed, a disconnected graph cannot have a Hamiltonian path.)
- If in a graph G there is a Hamiltonian circuit then every vertex of G must have degree at least 2.
- If in a graph G there is a Hamiltonian cycle then G must be strongly connected.

A nontrivial necessary and sufficient condition for finding whether a given graph has a Hamiltonian path (circuit, circle, respectively) is not known.

Chapter 8

Combinatorics

In this lecture we will focus on so called enumerative combinatorics, it means a way how to count the number of certain objects. At first, we introduce two main principles that will help us to solve more complicated tasks.

8.1 Multiplication and Addition Principles

8.1.1 Multiplication Principle. Assume that a certain activity can be divided into k independent consecutive steps. If step 1 can be done in n_1 ways, step 2 can be done in n_2 ways, etc., and step k can be done in n_k ways, then the number of distinct ways the activity can be done is

$$n_1 \cdot n_2 \cdot \ldots \cdot n_k.$$

□

8.1.2 Example. How many distinct binary words of length n there are?

Solution. A binary word of length n is any sequence $a_1\,a_2\,\ldots\,a_n$ where for all i we have $a_i \in \{0,1\}$. Such n-tuples can be formed as follow: we choose a_1, then a_2, etc, a_n. For each a_i there are 2 possibilities, indeed, either 0 or 1. So there are $2 \cdot 2 \cdot \ldots \cdot 2 = 2^n$ different binary words. □

8.1.3 Addition Principle. Assume that we have n sets $A_1, A_2, \ldots, A_n$ pairwise disjoint (which means that for $i \neq j$ it is $A_i \cap A_j = \emptyset$). Further, assume that each set A_i has k_i elements. The number of elements that can be chosen from A_1 or A_2 or ... or A_n is

$$k_1 + k_2 + \ldots + k_n.$$

Notice that it is the same as the number of elements the set $A_1 \cup A_2 \cup \ldots \cup A_n$ has. □

8.1.4 Example. How many ways can we select two different kinds chocolate bars if we have 4 different dark bars, 5 different milk bars, and 3 different white bars?

Solution. Using the multiplication principle, we know that there are $4 \cdot 5$ different choices of one dark and one milk bar, $4 \cdot 3$ different choices of one dark and one white bar, and $5 \cdot 3$ different choices of one milk and one white bar. These choices are pairwise disjoint, so the number of different choices is

$$20 + 12 + 15 = 47.$$

□

8.2 Permutations, Combinations, Variations

8.2.1 Permutations. To permute objects means to change order of the given objects.

Definition. Given n distinct elements $a_1, a_2, \ldots, a_n$. A *permutation* of $a_1, a_2, \ldots, a_n$ is any ordering of elements $a_1, a_2, \ldots, a_n$. □

Recall that a permutation of $a_1, a_2, \ldots, a_n$ can be viewed as a bijective (i.e. one-to-one and onto) mapping from $\{1, 2, \ldots, n\}$ to $\{a_1, a_2, \ldots, a_n\}$.

Proposition. The number of different permutations of elements $a_1, a_2, \ldots, a_n$ equals to $n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$. □

Justification. We use the multiplication principle. For the first element we have n possibilities, indeed, any element $a_1, a_2, \ldots, a_n$. For the second element we can now choose one of $n-1$ different elements (not to the one which was chosen as the first one). For the third element we have $n-2$ possibilities, etc. Hence, all together there are

$$n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 2 \cdot 1$$

distinct permutations. □

8.2.2 Factorial. For $n \geq 1$ the number

$$n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$$

is called n *factorial* and denoted by $n!$.

For $n = 0$ we define $0! = 1$.

8.2.3 Example. In a shop there are 6 types of chocolate. How many different ways these 6 types could be exhibited in a row?

Solution. Any permutation of $t_1, \ldots, t_n$ (where t_i represents the i-th type) describes one such exhibition. Hence, there are

$$6! = 6 \cdot 5 \cdot \ldots \cdot 2 \cdot 1 = 720$$

different ways.

8.2.4 Example. How many permutations of letters A, B, C, D, E, F contains CDE as a substring?

Solution. Since the letters CDE must be consecutive and in this order, we can assume that CDE is a new symbol, say Y. Then the question is: how many permutations of A, B, F, Y. There are

$$4 \cdot 3 \cdot 2 \cdot 1 = 24$$

such permutations. □

8.2.5 Variations.

Definition. A *k-variation* of n distinct elements $a_1, a_2, \ldots, a_n$ is a sequence of k (distinct) elements of the set $\{a_1, a_2, \ldots, a_n\}$. The number of distinct k-variations is denoted by $P(n, k)$. □

Remark. k-variations are also called *k-permutations.*

Proposition. The number of k-variations of a set of n distinct elements ($k \leq n$), is

$$P(n, k) = n \cdot (n-1) \cdot \ldots \cdot (n-k+1) = \frac{n!}{(n-k)!}.$$

□

Justification. The proof is similar to the proof of the number of permutations. Indeed, for the first element we have n distinct possibilities, for the second element we have $n-1$ distinct possibilities, etc., ..., for the k-th element we have $n-k+1$ distinct possibilities. Now, the multiplication principle finishes the argument. □

Example. A password for a credit card contains four distinct digits.

- How many passwords can be formed?
- How many passwords that do not start with 0 can be formed?

Solution.

1) These are 4-variations of ten digits $0, 1, \ldots, 9$. Hence, the number of distinct passwords is

$$10 \cdot 9 \cdot 8 \cdot 7 = 5040.$$

2) By the multiplication principle, the number of passwords is

$$9 \cdot (9 \cdot 8 \cdot 7) = 4536.$$

Indeed, for the first digit we have nine possibilities (digits $1, \ldots, 9$), and this digit is followed by 3-variation of the remaining digits and 0.

8.2.6 Combinations.

Definition. Given a finite set $A = \{a_1, a_2, \ldots, a_n\}$ of n distinct elements. An *k-combination* of A is an unordered selection of k elements of A (in other words, an k element subset of A). The number of distinct k-combinations of n element set is denoted by $C(n,k)$. □

8.2.7 Proposition. The number of distinct k-combinations of n element set equals

$$C(n,k) = \binom{n}{k} = \frac{n!}{(n-k)!\,k!}.$$

□

Justification. There are $P(n,k)$ distinct k-variations of n distinct elements. k-variations that differ only by ordering correspond to the same k-combination. Since there are $k!$ permutations of a k element set, we get

$$C(n,k) = \frac{P(n,k)}{k!} = \frac{n \cdot \ldots \cdot (n-k+1)}{k!} = \frac{n!}{(n-k)!\,k!}.$$

□

8.2.8 Binomial Coefficients. Let $k \le n$ be two natural numbers. Then the number

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

is called a *binomial coefficient* (or a *combinatorial number*).

8.2.9 Proposition.

1) For all $n \in \mathbb{N}$ we have $\binom{n}{0} = 1$
2) For all $n \in \mathbb{N}$ we have $\binom{n}{1} = n$.
3) For all $k \le n$, $k, n \in \mathbb{N}$, we have

$$\binom{n}{k} = \binom{n}{n-k}.$$

4) For all $k \leq n$, $k, n \in \mathbb{N}$, it holds that

$$\binom{n}{k-1} + \binom{n}{k} = \binom{n+1}{k}.$$

□

Justification. Properties 1), 2), and 3) are easy consequences of the definition of binomial coefficients.

We will show the property 4): We have

$$\binom{n}{k-1} + \binom{n}{k} = \frac{n!}{(n-k+1)!\,(k-1)!} + \frac{n!}{(n-k)!\,k!} =$$

$$\frac{n!}{(n-k)!\,(k-1)!}\left(\frac{1}{n-k+1} + \frac{1}{k}\right) = \frac{n!}{(n-k)!\,(k-1)!}\left(\frac{n+1}{(n-k+1)\,k}\right) =$$

$$\frac{(n+1)!}{(n+1-k)!\,k!} = \binom{n+1}{k}.$$

□

Remark. The last property from 8.2.9 is a basis of so called Pascal triangle.

8.2.10 Variations and Combinations with Repetition. If we allow repetitions then the number of k-variations of n elements is

$$n^k.$$

□

Justification. Indeed, every chosen element can be one of the n elements. Since there are k elements to be chosen, the multiplication principle gives the total number n^k. □

If we choose k elements of A where repetition is allowed then the number of combinations is

$$\binom{n+k-1}{k} = \frac{(n+k-1)!}{k!\,(n-1)!}.$$

□

Idea of a justification. Let us show the idea on an example. Assume that we should choose strings of length 4 consisting of letters from the set $\{A, B, C, D, X, Y\}$, repetitions are allowed, but the order letters is not important. Hence, $AAAX$, $CXXY$ are examples of such strings, and strings $AAAX$, $AXAA$, $AAXA$ are considered to be the same. How many distinct strings can be formed?

Any such string can be represented as an 9-tuple consisting of five symbols $|$ and four symbols $\cdot$. The symbol $|$ shows the change of a letter (so we have $6-1=5$ of them, indeed, change from A to B, from B to C, from C to D, from D to X, and from X to Y). The symbol $\cdot$ stands for a letter at the respective positions in the list $\{A, B, C, D, X, Y\}$. For instance

$$AAAX \quad \text{is represented by} \quad \cdot \; \cdot \; \cdot \; | \; | \; | \; | \; \cdot \; |$$

Indeed, there are three A's, no B, no C, no D, one X, and no Y. Similarly,

$$CXXY \quad \text{is represented by} \quad | \; | \; \cdot \; | \; | \; \cdot \; \cdot \; | \; \cdot$$

Indeed, there is no A, no B, one C, two X's, and one Y.

Hence, such a string is represented by choosing four $\cdot$ out of nine positions where a $\cdot$ can be placed (or equivalently, by choosing five $|$ out of nine positions where $|$ can be placed). Therefore, the number of distinct string is

$$\binom{4+6-1}{4} = \binom{4+6-1}{6-1}.$$

Generally, we choose subsets of k elements out of a set of $k+n-1$ distinct places, which equals to

$$\binom{n+k-1}{k}.$$

8.2.11 Example. In a shop 6 types of chocolate bars are sold. Three friends come to a shop and each of them buys one chocolate bar. How many ways could that be done if

1) each friend chooses different type of chocolate bars;
2) they may choose the same type of chocolate bars?

Solution.

1) Call the friends A, B, and C. The number of different choices is $\frac{6!}{3!} = 120$, since we choose triples (where the order is important) out of 6 different types and there is no repetition.

2) If repetitions are allowed then there are $6 \cdot 6 \cdot 6 = 216$ possibilities.

8.2.12 Binomial Theorem. Let us recall the binomial theorem.

Theorem. Let n be a natural number. Then for every real numbers x, y it holds that

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k}\, y^k.$$

□

8.2.13 Principle of Inclusion and Exclusion. The addition principle deals with the number of elements which a union of pairwise disjoint sets has. But often we need to know the number of elements a union of two sets A and B has even when A and B are not disjoint.

Theorem. For any sets A, B, C we have

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|.$$

□

Justification. The formula for the number of elements that $A \cup B$ has is evident. Indeed, we sum the number of elements of the both sets and subtract the number of elements of their intersection, since they were calculated twice.

The justification for a union of three sets is similar, only tedious and we omit it. □

8.2.14 Proposition. Let A and B be two sets, $|A| = n$, $|B| = k$. Then there are k^n distinct mappings from A to B. □

Justification. The proposition above is an easy consequence of the multiplication principle. Denote $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_k\}$. □

8.2.15 Dirichlet's, Pigeonhole Principle. This principle is an easy observation but applicable in many counting problems.

Theorem (Pigeonhole principle). Let A and B be two sets, $|A| = n$, $|B| = k$. If $n > k$ then there does not exist a one-to-one mapping from A to B. □

Justification. We will use the multiplication principle. Denote $A = \{a_1, a_2, \ldots, a_n\}$. Let us construct an arbitrary mapping $f\colon A \to B$ which could be one-to-one. For $f(a_1)$ we have k different choices, for $f(a_2)$ only $k-1$ (indeed, we cannot use $f(a_1)$), for $f(a_3)$ only $k-2$ choices, etc, for $f(a_k)$ only a single element of B. Since $n > k$, there is $a_{k+1} \in A$ and $f(a_{k+1})$ must be the same as some of $f(a_1), f(a_2), \ldots, f(a_k)$. Hence, f is not one-to-one. □

8.3 Asymptotic Growth of Functions

8.3.1 Symbol $\mathcal{O}$. Let $g(n)$ be a nonnegative function. We say that a nonnegative function $f(n)$ *is* $\mathcal{O}(g(n))$ if there exists a positive constant c and a natural number n_0 such that

$$f(n) \leq c\, g(n) \quad \text{for every } n \geq n_0.$$

$\mathcal{O}(g(n))$ can be considered as a class of nonnegative function $f(n)$:

$$\mathcal{O}(g(n)) = \{f(n) \mid \exists c > 0, n_0 \text{ such that } f(n) \leq c\, g(n) \;\; \forall n \geq n_0\}.$$

8.3.2 Symbol Ω. Let $g(n)$ be a nonnegative function. We say that a nonnegative function $f(n)$ *is* $\Omega(g(n))$ if there exists a positive constant c and a natural number n_0 such that

$$f(n) \geq c\, g(n) \quad \text{for every } n \geq n_0.$$

$\Omega(g(n))$ can be considered as a class of nonnegative functions $f(n)$:

$$\Omega(g(n)) = \{f(n) \mid \exists c > 0, n_0 \text{ such that } f(n) \geq c\, g(n) \;\; \forall\, n \geq n_0\}.$$

8.3.3 Remark. We have $f(n)$ is $\Omega(g(n))$ if and only if $g(n)$ is $\mathcal{O}(f(n))$.

8.3.4 Symbol Θ. Let $g(n)$ be a nonnegative function. We say that a nonnegative function $f(n)$ *is* $\Theta(g(n))$ if there exist positive constants c_1, c_2 and a natural number n_0 such that

$$c_1\, g(n) \leq f(n) \leq c_2\, g(n) \quad \text{for every } n \geq n_0.$$

$\Theta(g(n))$ can be considered as a class of nonnegative functions $f(n)$:

$$\Theta(g(n)) = \{f(n) \mid \exists c_1, c_2 > 0, n_0 \text{ such that } c_1\, g(n) \leq f(n) \leq c_2\, g(n) \;\; \forall\, n \geq n_0\}.$$

8.3.5 Remark. $f(n)$ is $\Theta(g(n))$ if and only if $f(n)$ is $\mathcal{O}(g(n))$ and $\Omega(g(n))$.

8.3.6 Notation. Since the symbols $\mathcal{O}, \Omega, \Theta$ represent sets of functions, we write $f(n) \in \mathcal{O}(g(n))$. Some authors prefer the notation $f(n) = \mathcal{O}(g(n))$. If the later notation is used it is necessary to take in mind that the equality sigh used there does not have all the properties as a classical equality has. Similarly for other symbols.

8.3.7 Proposition. $f(n) \in \Theta(g(n))$ if and only if $g(n) \in \Theta(f(n))$.

8.3.8 Examples.

1. For every $a > 1$ and $b > 1$ we have

$$\log_a(n) \in \Theta(\log_b(n)).$$

2. The logarithm with base 2 is usually denoted by lg, i.e. $\lg(n) = \log_2(n)$. It holds that

$$\lg n! \in \Theta(n \lg n).$$

The second part of the above proposition follows from the following theorem.

8.3.9 Theorem (Gauss). For every $n \geq 1$ it holds that

$$n^{\frac{n}{2}} \leq n! \leq \left(\frac{n+1}{2}\right)^n.$$