Diskrete Mathematik

Ch. 1 Introduction

Discrete mathematics is concerend with finite and countable infinite mathematical structures.

The main theme of this course: abstraction Without it, writing good software becomes impossible.

A computer program is a discrete mathematical object, meaning that there is a mathematical proof that it is correct.

Mathematical statements are either true or false.

Important Signs:
- $\forall$ for all
- $\exists$ there exists
- $\equiv_3$ congruent to modulo 3

Ch. 2 Mathematical Reasoning & Proof

V2

A mathematical statement that is either true or false is called a proposition. When true, a proposition is often called a theorem or lemma, further if it's unclear it's called a assumption.

To connect mathematical statements we can use:

S and T	- both need to be true
S or T	- one needs to be true
$S \Rightarrow T$	- S implies T
$S \Leftrightarrow T$	- $S \Rightarrow T$ and $T \Rightarrow S$

A proof is used to show that a mathematical statement is true or false.

Informal proof - based on already proofen axioms and known facts, formulated in common language

Formal proof - rigorous and formal, with the advantages of:

- preventing errors
- proof complexity and automatic verification
- precision and deeper understanding

Propositional Logic

The logical values (constants) „true" and „false" are usualy denoted as 1 and 0

Logical operators:

In fact, ¬ and ∧ are sufficient to express every logical function

- $\neg A$ NOT
- $A \wedge B$ AND
- $A \vee B$ OR
- $A \rightarrow B$ Implication
- $A \leftrightarrow B$ " on both sides

These operators are functions (e.x. ∨ is a function of $\{0,1\} \times \{0,1\} \rightarrow \{0,1\}$, this can be shown as a tabel.

A	B	A ∨ B
0	0	0
0	1	1
1	0	1
1	1	1

→ A expression like this is called a formula

There are priority rules for logical operators:

1. ¬
2. ∧ / ∨
3. → / ↔

If two logical functions are equivalent, we denote it with $F \equiv G$.

↖ by using $\equiv$ or $\vDash$ it becomes a statement.

Basic equivalences:

Lemma 2.1.

1) $A \wedge A \equiv A$ *and* $A \vee A \equiv A$ *(idempotence);*
2) $A \wedge B \equiv B \wedge A$ *and* $A \vee B \equiv B \vee A$ *(commutativity);*
3) $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$ *and* $(A \vee B) \vee C \equiv A \vee (B \vee C)$ *(associativity);*
4) $A \wedge (A \vee B) \equiv A$ *and* $A \vee (A \wedge B) \equiv A$ *(absorption);*
5) $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$ *(distributive law);*
6) $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$ *(distributive law);*
7) $\neg\neg A \equiv A$ *(double negation);*
8) $\neg(A \wedge B) \equiv \neg A \vee \neg B$ *and* $\neg(A \vee B) \equiv \neg A \wedge \neg B$ *(de Morgan's rules).*

If a formula is a logical consequence, it is denoted by $F \vDash G$. If we could interpret this as $F \leq G$, but we don't do that!

! When using $\vDash$ or $\equiv$, the result is a statement. !

The distinction between formula and statement is really important!

If a formula is always true, we call it a tautology (T) or valied. If it is true for atleast one truth assignment it is satisfiable, if not, it is unsatisfiable ($\bot$).

Predicate Logic

V3

Propositional logic is often not capable to express a statement (e.g. there are infinitly many prime numbers) in such cases we use predicate logic.

Predicate logic is based on k-ary predicates P in a universe U. P on U is a function $U^k \to \{0,1\}$. The predicate P(x) assigns every x to a value 0=false or 1=true.

In predicate logic, we can also use functions on U, for example addition and multiplication.

Further we use the Quantifiers $\forall$ and $\exists$.

$\forall x\, P(x)$	P(x) is true for all x in U.
$\exists x\, P(x)$	P(x) is true for some x in U.

V4

Formulas are often leave room open for interpretation. They do this by having „free parts" that are not fixed. Such formulas can't be true or false until a interpretation for the „free parts" is applied.

If a formula has no interpretation that allow it to be true, it is called unsatisfiable if there is a interpretation that allows it to be true it is called satisfiable and if it is true for an interpretation, it is a tautology.

Logical Formulas and Mathematical Statements

Formulas and statements are different, because the symbols of a formula can be interpreted differently.

When a interpretation is fixed for a formula, it becomes a mathematical statement. If this is the case we can also tell if a formula is true or false.

While formulas are no statement, we can make statements about them:

- F is unsatisfiable
- F is valid ($\models F$)
- $F \models G$
- $\{F, G\} \models H$
- F is valid $\Rightarrow$ G is valid

Proof Patterns

Composition of Implications

If $S \Rightarrow T$ and $T \Rightarrow U$ are true, then $S \Rightarrow U$ is also true.

Proof. One writes down the truth tables for $(A \to B) \wedge (B \to C)$ and $A \to C$. If the first evaluates to true, then the second one will as well.

Direct Proof of an Implication

The proof of an implication $S \Rightarrow T$, by assuming S and then proving T under this assumption.

Indirect Proof of an Implication

The proof of $S \Rightarrow T$ works by assuming T is false and proving S is false under this assumption.

Ex. Proof that if x is irrational, then $\sqrt{x}$ has to be irrational too.

Modus Ponens

1. Find a statement R
2. Prove R
3. Prove $R \Rightarrow S$

Ex. $2^{3000} \equiv_{3001} 1$ (in Script.)

Case Distinction

1. Find a finite list of m. statements $R_1, \ldots, R_k$ (cases)
2. Prove that one R_i is true.
3. Prove $R_i \Rightarrow S$ for $i = 1, \ldots, k$.

Ex. Checkerboard with one piece missing, can be covered by L-Shapes.

Proof by Contradiction

1. Find a statement T.
2. Prove that T is false.
3. Assume S is false and prove T has to be true for this. (a contradiction)

Existence Proofs

Consider a set $\mathcal{X}$ of parameters and for each $x \in \mathcal{X}$ a statement S_x. A existence proof is a proof that a statement is true for atleast one $x \in \mathcal{X}$. It is constructive if it exhibits an a for which S_a is true, otherwise it is non-constructive.

Existence Proof via Pigeonhole Principle

Schubfach Prinzip

If a set of n objects is partitioned into $k < n$ sets, then atleast one set must contain $\lceil \frac{n}{k} \rceil$ objects.

Ex. Among 100 people, atleast 9 are born in the same month.

Proofs by Counterexample

Considering a set $\mathcal{X}$ of parameters and for each $x \in \mathcal{X}$ a statement S_x. A proof by counterexample is proof that S_x is not true for all $x \in \mathcal{X}$.

Proof by Induction

See notes on A&D.

Ch.3 Sets, Relations & Functions

A set is a collection of objects. For every object it is defined whether it is a element of a set or not (denoted by $x \in A$). Order and mulliple occurrences of elements is irrelevant.

The number of elements in a finite set is the cardinality, denoted by $|A|$.

A set can also be defined by a property of the elements.

$\{x \in A \mid P(x)\}$ elements of A having property P

Sets are equal if they contain the same elements.

A set can be a element of another set

A set A is a subset of another set B if all elements of A are also part of B.

A ordered pair (a,b) is defined as $\{\{a\},\{a,b\}\}$.

A set is empty if it does not contain any elements. This is often denoted as $\emptyset$.

$\emptyset$ is a subset of every set

With the empty set, we could define the natural numbers. V7

$$0 \stackrel{\text{def}}{=} \emptyset \quad , \quad 1 \stackrel{\text{def}}{=} \{\emptyset\} \quad , \quad 2 \stackrel{\text{def}}{=} \{\emptyset, \{\emptyset\}\}$$

Defining the function $s(n) = n \cup \{n\}$

One can then start to define operations like +

$$m + 0 = m \qquad m + s(n) = s(m+n)$$

Power set

The power set of a set A, denoted $\mathcal{P}(A)$, is the set of all subsets of A.

$$\mathcal{P}(A) = \{S \mid S \subseteq A\}$$

e.x. $\mathcal{P}(\emptyset) = \{\emptyset\}$
$\mathcal{P}(a) = \{\emptyset, \{a\}\}$

$\emptyset$ is part of every power set.

$\mathcal{P}(A)$ has cardinality $2^{|A|}$.

Union & Intersection

$$A \cup B = \{x \mid x \in A \vee x \in B\}$$ union logical or

$$A \cap B = \{x \mid x \in A \wedge x \in B\}$$ intersection logical and

This can be extended to a set of sets.

$$\bigcup \mathcal{A} = \{x \mid x \in A \text{ for some } A \in \mathcal{A}\}$$

$$\bigcap \mathcal{A} = \{x \mid x \in A \text{ for all } A \in \mathcal{A}\}$$

Typically sets in $\mathcal{A}$ are indexed by a Index set I. In this case one writes $\{A_i\}_{i \in I}$. The union becomes $\bigcup_{i \in I} A_i$ and the intersection $\bigcap_{i \in I} A_i$.

Complement

$$\bar{A} = \{x \in U \mid x \notin A\}$$ logical negation

Difference

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

The Cartesian Product

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

The set of all ordered pairs (a,b) (defined as $(a,b) = \{\{a\}, \{a,b\}\}$) where a is from set A and b is from set B.

Cardinality: $|A \times B| = |A| \cdot |B|$

Generally: $\bigtimes_{i=1}^{k} A_i = \{(a_1, \dots, a_k) \mid a_i \in A_i \text{ for } 1 \leq i \leq k\}$

Not associative!

Relations

V8

We can define relations between a set A an a set B. This (binary) relation ρ is a subset of $A \times B$. If $A = B$ then ρ is a relation on A.

$$(a,b) \in \rho \equiv a \rho b$$

Relations can be interpreted as graphs or as matrixes: $A = \{a,b,c\}$ $B = \{d,e,f\}$

$|A| \times |B|$ matrix

$$M^{\rho} = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} d & e & f \end{array} \\ \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{array}$$

$$\rho = \{(a,d), (b,d), (c,e), (a,f)\}$$

The concept of relations can be generalized from binary to k-ary for given sets $A_1, \ldots, A_k$. This can for example be useful for database modeling. But in this course we only look at binary relations.

We can use set operations on relations. (e.g. $\leqslant \cup \geqslant$ is the identity relation)

Further we can take the inverse of a relation ρ.

$$\hat{\rho} \stackrel{\text{def.}}{=} \{(a,b) \mid (b,a) \in \rho\}$$

We can also take the composition of relations.

$$\rho \circ \sigma \stackrel{\text{def}}{=} \{(a,c) \mid \exists b \in B \, ((a,b) \in \rho \wedge (b,c) \in \sigma)\}$$

The composition $\rho \circ \rho = \rho^2$. Generalized ρ^n.

Special Properties

Name	Definition (id = identity)	Example
reflexive	$a \rho a$ is true for all $a \in A$, i.e. if $id \subseteq \rho$	$\leq$
irreflexive	$a \not\rho a$ is true for all $a \in A$, $\rho \cap id = \emptyset$	$<$
symmetric	$a\rho b \Leftrightarrow b\rho a$ is true, $\rho = \hat{\rho}$	$=$
antisymmetric	$(a\rho b \wedge b\rho a) \Rightarrow a=b$, $\rho \cap \hat{\rho} \subseteq id$	$\leq$
transitive	$(a\rho b \wedge b\rho c) \Rightarrow a\rho c$, $\rho^2 \subseteq \rho$	$\leq$

Transitive Closure $\rho^* = \bigcup_{n \in \mathbb{N}\setminus\{0\}} \rho^n$

Equivalence Relations

VG

A equivalence relation is a relation on a set A that is reflexive, symetric and transitive (e.g. $\equiv_m$).

For $a \in A$, the set of elements of A that are equivalent to a is called the equivalence class $[a]_\theta$.

$$[a]_\theta \stackrel{def}{=} \{b \in A \mid b \,\theta\, a\}$$

The set A/θ of equivalence classes of an equivalence relation θ on A is a partition of A.

Partial Order Relations

A partial order on a set A is a relation that is reflexive, antisymetric and transitive. A set A with a partial order $\preccurlyeq$ on A is called a partially ordered set or poset and is denoted as $(A; \preccurlyeq)$.

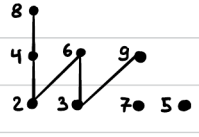
For a partial order we can define $\prec$ similar to how $<$ is obtained from $\leq$.

$$a \prec b \overset{\text{def}}{\Leftrightarrow} a \preccurlyeq b \wedge a \neq b$$

If any two elements of a poset are comparable, it is called totally ordered by $\preccurlyeq$.

In a poset a element a covers b, if and only if $a \prec b$ with no $a \prec c$ and $c \prec b$.

A Hasse diagram of a poset is the graph whose vertices are elements of A and where there is a edge from a to b if and only if b covers a.



Hasse diagram of the poset $(\{2,3,4,5,6,7,8,9\}; |)$

For two given Posets a relation $\leq$ can be defined on $A \times B$.

$$(a_1, b_1) \leq (a_2, b_2) \overset{\text{def}}{\Leftrightarrow} a_1 \preccurlyeq a_2 \wedge b_1 \sqsubseteq b_2$$

this is again a partial order relation. Further the relation $\leq_{lex}$ is defined as

$$(a_1, b_1) \leq_{lex} (a_2, b_2) \overset{\text{def}}{\Leftrightarrow} a_1 \prec a_2 \vee (a_1 = a_2 \wedge b_1 \sqsubseteq b_2)$$

Again this is a partial order relation. $\leq_{lex}$ is usefull if A and B are totally ordered.

Special Elements of Posets

1. minimal / maximal element $(a, b \in A$ exists no $b \prec a \setminus a \prec b)$
2. least / greatest element $(a, b \in A$ for all $a \preccurlyeq b \setminus b \preccurlyeq a)$
3. lower / upper bound of $S \subseteq A$ $(a \in A, b \in S$ for all $a \preccurlyeq b \setminus b \preccurlyeq a)$
4. greatest lower / least upper bound of $S \subseteq A$ (greatest / least element of all lower / upper bounds)

For a set $\{a, b\} \subseteq A$ a meet is $a \wedge b$ and a join is $a \vee b$. If every pair has a join and meet it is called a lattice.

Functions

Functions are a special form of relations. A function $f: A \mapsto B$ from a domain A to a codomain B is a relation from A to B $(a\,f\,b)$.

$$\forall a \in A \quad \exists b \in B \quad a\,f\,b \qquad \text{(f is totally defined)}$$
$$\forall a \in A \quad \forall b, b' \in B \quad (a\,f\,b \wedge a\,f\,b' \rightarrow b = b') \qquad \text{(f is well defined)}$$

The set of all functions $A \mapsto B$ is B^A.

A partial function of $A \mapsto B$ is a relation so that 2. holds true.

For a subset S of A, the image is:

$$f(S) \stackrel{\text{def}}{=} \{f(a) \mid a \in S\}$$

The subset $f(A)$ of B is the image / range of f, denoted as $\text{Im}(f)$.

For a subset T of B, the preimage of T $(f^{-1}(T))$ is

$$f^{-1}(T) \stackrel{\text{def}}{=} \{a \in A \mid f(a) \in T\}$$

injective	for $a \neq b$ it is $f(a) \neq f(b)$
surjective	if $f(A) = B$ / for every $b \in B$, $b = f(a)$
bijective	both injective and bijective

Compositions of functions $g \circ f$ is defined as $(g \circ f)(a) = g(f(a))$. Different to normal relations!
It is associative.

Countable and Uncountable Sets

i. two sets A and B are equinumerous, $A \sim B$, if there exist a bijection $A \mapsto B$.

ii. The set B dominates the set A, $A \preccurlyeq B$, if $A \sim C$ for $C \subseteq B$ or if there exist a injective function $A \mapsto B$.

iii. A set A is countable if $A \preccurlyeq B$ and uncountable otherwise.

$$A \preccurlyeq B \wedge B \preccurlyeq A \Rightarrow A \sim B$$

For finite sets we have $A \sim B$ if $|A| = |B|$.

All finite sets are countable.

Important Countable Sets

- $\{0,1\}^*$ ← finite sequence, the set of finite binary sequences.
- $\mathbb{N}\times\mathbb{N}$, the set of ordered pairs of natural numbers
  - $A\times B$ is countable if $A \preccurlyeq \mathbb{N} \wedge B \preccurlyeq \mathbb{N}$
  - $\mathbb{Q}$ is countable

It follows that:

i. $n \in \mathbb{N}$, the set A^n of n-tuples over A is countable

ii. The union $\bigcup_{i\in\mathbb{N}} A_i$ of a countable list $A_1, \ldots, A_i$ of countable sets is countable

iii. The set A^* of finite sequences of A (∞) is countable.

Important Uncountable Sets

- $\{0,1\}^\infty$, the set of all infinit binary sequences
  - can be proven with Cantor's diagonalization argument

Existence of Uncountable Functions

p. 67 script

A function $f: \mathbb{N} \mapsto \{0,1\}$ (← false, true) is called computable if there is a program that, for every $n \in \mathbb{N}$ has a output $f(n)$.

Ex. prime: $\mathbb{N} \to \{0,1\}$

There are uncomputable functions $\mathbb{N} \to \{0,1\}$

Ex. Halting problem important for CS

Ch.4 Number Theory

V12

Number Theory is the mathematical theory of the natural numbers $\mathbb{N}$ or more generally of the Integers $\mathbb{Z}$. While we focus on the numbers $\mathbb{Z}$, everything should also be applicable to any ring. (def. by ring axiomes). We assume that we know what integers, operations and basic facts about integers are.

Divisors

For $a, b \in \mathbb{Z}$ we say a divides b, denoted $a|b$, if b is a multiple of a.

$$a|b \overset{\text{def.}}{\iff} \exists c \; b = a \cdot c$$

(c: quotient, a: divisor)

If a is not a divisor of b, we get a rest. We can say that unique integers q, r exist for any b, $a \neq 0$.

$$b = q \cdot a + r \quad \text{where} \quad 0 \leq r < |a|$$

(r: remainder)

The remainder r is denoted as $R_a(b)$ or $b \bmod a$.

For two integers a, b (not both 0), an integer d is called the greatest common divisor, if

$$d|a \wedge d|b \wedge \forall c \left((c|a \wedge c|b) \rightarrow c|d\right)$$

If $\gcd(a,b) = 1$, then a and b are relatively prime.

Further holds, $\gcd(m,n) = \gcd(m, n-q\cdot m) = \gcd(m, R_m(n))$.

For $a,b \in \mathbb{Z}$, the ideal generated by a and b, denoted (a,b) is the set $\{u\cdot a + v\cdot b \mid u,v \in \mathbb{Z}\}$. Similarly, the ideal generated by a single integer a is $(a) = \{u\cdot a \mid u \in \mathbb{Z}\}$.

For any (a,b) exists a (c) such that $(a,b) = (c)$. If $a \neq 0$, $b \neq 0$ then d is the gcd of a and b. It follows that $\gcd(a,b) = ua + vb$.

The least common multiple ℓ is defined as,

$$a \mid \ell \wedge b \mid \ell \wedge \forall c\,((a \mid c \wedge b \mid c) \rightarrow \ell \mid c)$$

Primes

A positive integer $p > 1$ is called prime if the only positive divisor of p are 1 and p. Every other integer is a composite.

The fundamental theorem of arithmetic states that every positive integer can be writen uniquely as the product of primes.

Congruences and Modular Arithmetic

We say a is congruent to b modulo m, if m divides $a-b$.

$$a \equiv_m b \overset{\text{def.}}{\Longleftrightarrow} m \mid (a-b)$$

For any $m \geq 1$, $\equiv_m$ is a equivalence relation on $\mathbb{Z}$.

If $a \equiv_m b$ and $c \equiv_m d$ then $a+c \equiv_m b+d$ and $ac \equiv_m bd$.

If $f(x_1, \dots, x_k)$ is a multi-variable polynomial with k variables. Then $a_i \equiv_m b_i$ for $1 \leq i \leq k$,

$$\Rightarrow f(a_1, \dots, a_k) \equiv_m f(b_1, \dots, b_k)$$

Modular arithmetic

Since $\equiv_m$ is a equivalence relation on $\mathbb{Z}$ we get the equivalence classes $[0], [1], \dots, [m-1]$. Each equivalence class $[a]$ has a natural reprensentative $R_m(a) \in [a]$ in the set

$$\mathbb{Z}_m := \{0, \dots, m-1\}$$ of remainders modulo m.

For any $a, b, m \in \mathbb{Z}$ with $m \geq 1$:

$$a \equiv_m R_m(a)$$
$$a \equiv_m b \iff R_m(a) = R_m(b)$$

It follows that

$$R_m(f(a_1 \dots, a_k)) \equiv_m R_m(f(R_m(a_1), \dots, R_m(a_k))).$$

Multiplicative Inverse

Considering $ax \equiv_m b$, obviously if x is a solution, so is $x + k \cdot m$ for any $k \in \mathbb{Z}$. Of special interest is the case where $\gcd(a,m) = 1$ and $b = 1$.

$$ax \equiv_m 1 \iff \gcd(a,m) = 1 \qquad x \in \mathbb{Z}_m \text{ is unique}$$

x is called the multiplicative inverse of a mod m. Often denoted $x \equiv_m a^{-1}$ or $x \equiv_m \frac{1}{a}$.

To calculate the multiplicative inverse, the extended Euclidean algorithm is used.

Ex. $\gcd(99, 78) = \ldots$

$$\begin{aligned}
99 &= 78 + 21 &&\iff 21 = 99 - 78 \\
78 &= 3 \cdot 21 + 15 &&\iff 15 = 78 - 3 \cdot 21 = 4 \cdot 78 - 3 \cdot 99 \\
21 &= 15 + 6 &&\iff 6 = 21 - 15 = 4 \cdot 99 - 5 \cdot 78 \\
15 &= 2 \cdot 6 + 3 &&\iff 3 = 15 - 2 \cdot 6 = -11 \cdot 99 + 14 \cdot 78 \\
6 &= 2 \cdot 3 + 0 \text{ ends when } r = 0 \text{ or } r = 1 && \qquad \uparrow \text{ inverse if } r = 1
\end{aligned}$$

$\Rightarrow$ if $r = 0$ there is no inverse

Chinese Remainder Theorem

Let $m_1 \dots m_r$ be pairwise relatively prime and let $M = \prod_{i=1}^{r} m_i$. For every list $a_1, \dots, a_r$ with $0 \le a_i < m_i$ for $1 \le i \le r$, the system

$$\begin{aligned} x &\equiv_{m_1} a_1 \\ x &\equiv_{m_2} a_2 \\ &\vdots \\ x &\equiv_{m_r} a_r \end{aligned}$$

has one unique solution $0 \le x < M$.

Ex. solving CRT

⓪

$$\begin{aligned} x &\equiv_3 2 \\ x &\equiv_4 1 \\ x &\equiv_5 4 \end{aligned}$$

①

$$M_i = \frac{M}{m_i}$$

- $M_1 = \frac{3 \cdot 4 \cdot 5}{3} = 20$
- $M_2 = \frac{3 \cdot 4 \cdot 5}{4} = 15$
- $M_3 = \frac{3 \cdot 4 \cdot 5}{5} = 12$

②

$$N_i \cdot M_i \equiv_{m_i} 1$$

- $N_1 \cdot 20 \equiv_3 1 \iff N_1 \cdot 2 \equiv_3 1 \Rightarrow N_1 = 2$
- $N_2 \cdot 15 \equiv_4 1 \iff N_2 \cdot 3 \equiv_4 1 \Rightarrow N_2 = 3$
- $N_3 \cdot 12 \equiv_5 1 \iff N_3 \cdot 2 \equiv_5 1 \Rightarrow N_3 = 3$

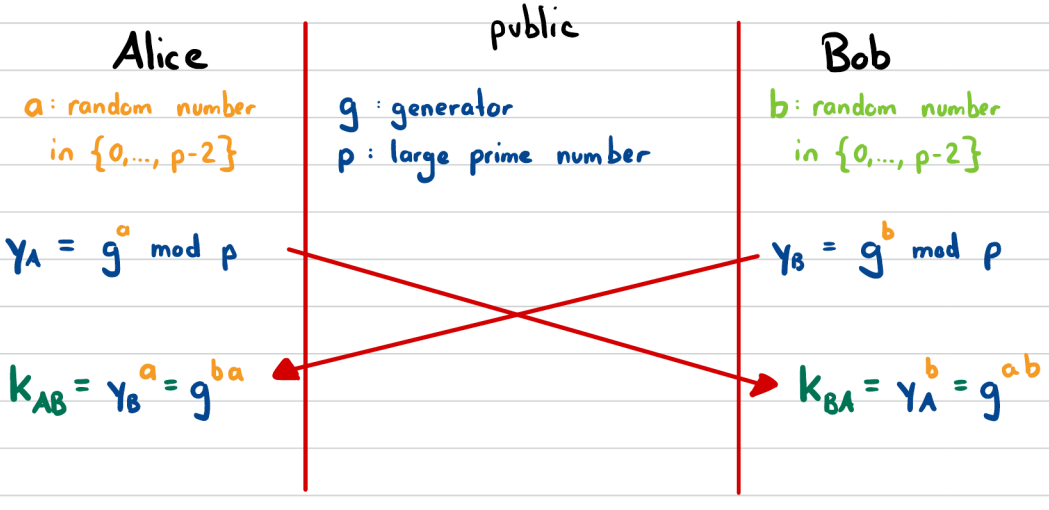
③

$$\sum_{i=1}^{r} a_i \cdot M_i \cdot N_i = 2 \cdot 20 \cdot 2 + 1 \cdot 15 \cdot 3 + 4 \cdot 12 \cdot 3 = 80 + 45 + 144 = 269$$

$$\equiv_{\underbrace{3 \cdot 4 \cdot 5}_{60}} 29 = x$$

Diffie Hellman

Diffie Hellman is one of the most used protocols for key-exchange. It uses the principle of a one-way function, a function that is easy to calculate but extremly hard to reverse.



It's easy to see that $k_{AB} = k_{BA}$. This is now the secret key, only known by Bob and Alice.

Ch.5 Algebra

An operation on a set S is a function $S^n \to S$, where $n \geq 0$ is called the „arity" of an operation.

arity 1 = unary arity 2 = binary

An algebra is a pair $\langle S; \Omega \rangle$ where S is a set (carrier) and $\Omega = (\omega_1, \omega_2, \dots \omega_n)$ is a list of operations on S.

$\langle \mathbb{Z}; +, -, 0, \cdot, 1 \rangle$ is an example of a algebra.

binary operators: $+, \cdot$
unary operators: $-$
neutral elements: $0, 1$

Monoids & Groups

A (left/right) neutral element of an algebra $\langle S; * \rangle$ is an element $e \in S$ such that $e * a = a$ or $a * e = a$ or both for all $a \in S$. There is at most one neutral element per operation.

A binary operation $*$ on a set S is associative if $a * (b * c) = (a * b) * c$.

← one operation

A Monoid is an algebra $\langle M; *, e \rangle$ where $*$ is associative and e is the neutral element.

A (left/right) inverse b of an element a in an algebra $\langle S; *, e\rangle$ exists if $b * a = e$ or $a * b = e$ or both.

A group is an algebra $\langle G; *, \hat{}, e\rangle$ satisfying the axioms:

- G1: $*$ is associative
- G2: e is a neutral element
- G3: every element a has an inverse $\hat{a}$.

These axioms are not minimal and can be simplified.

A group or monoid is called commutative or abelian if $a * b = b * a$ for all $a, b \in G$.

Direct Products

V15

The direct product of n groups $\langle G_1; *_1\rangle \dots \langle G_n; *_n\rangle$ is the algebra

$$\langle G_1 \times \dots \times G_n ; \star \rangle$$

where the operation $\star$ is component-whise (so are inverse and the NE)

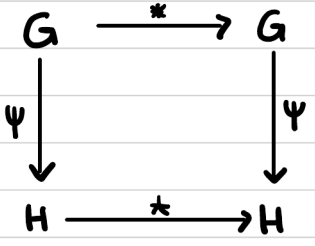
$$(a_1, \dots, a_n) \star (b_1, \dots, b_n) = (a_1 * b_1, \dots, a_n * b_n)$$

Group homomorphism

A function ψ from a group $\langle G; *, \hat{}, e \rangle$ to a group $\langle H; \star, \sim, e' \rangle$ is a group homomorphism if

$$\psi(a * b) = \psi(a) \star \psi(b)$$

If ψ is bijective, then it is called a isomorphism and we write $G \simeq H$.



Subgroups

A subset $H \subseteq G$ of a group $\langle G; *, \hat{}, e \rangle$ is called a subgroup of G if $\langle H; *, \hat{}, e \rangle$ is a group, i.e. H is closed with respect to all operations:

1) $a * b \in H$ for all $a, b \in H$
2) $e \in H$
3) $\hat{a} \in H$ for all $a \in H$

For any group $\langle G; *, \hat{}, e \rangle$, there exist two trivial subgroups: the subsets $\{e\}$ and G.

For now we will look at groups with the "$\cdot$" multiplicative operation, multiplicative notation, without loss of generality.

For $n \in \mathbb{Z}$, a^n is defined:

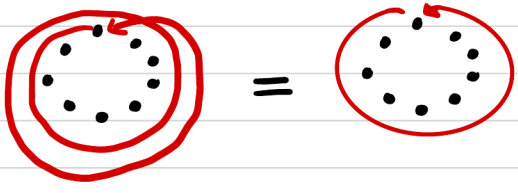
- $a^0 = e$
- $a^n = a \cdot a^{n-1}$ for $n \geq 1$
- $a^n = a^{-1^{|n|}}$ for $n \leq -1$

Let G be a group and $a \in G$. The order of a, denoted $\text{ord}(a)$, is the least $m \geq 1$ such that $a^m = e$, if such exists. Otherwise $\text{ord}(a)$ is infinite.

In a finite group G, every element has a finite order.
$|G|$ is called the order of G.
cardinality

Cyclic groups

If G is a group and $a \in G$ has finite order, then for any $m \in \mathbb{Z}$ we have $a^m = a^{R_{\text{ord}(a)}(m)}$



For group G and $a \in G$, the group generated by a, denoted $\langle a \rangle$, is defined

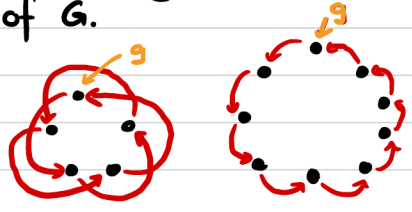
$$\langle a \rangle \stackrel{\text{def}}{=} \{a^n \mid n \in \mathbb{Z}\}$$

It is easy to see that $\langle a \rangle$ is the smallest subgroup of G containing the element a. For finite groups we have

$$\langle a \rangle \overset{\text{def}}{=} \{e, a, a^2, \ldots, a^{\text{ord}(a)-1}\}$$

A group $G = \langle g \rangle$ generated by an element g is called cyclic and g is the generator of G.

If g is a generator, so is g^{-1}.



A cyclic group of order n is isomorphic to $\langle \mathbb{Z}_n ; \oplus \rangle$ and hence abelian.

The Order of Subgroups

! **Lagrange:** Let G be a finite group and $H \subseteq G$. Then the order of H divides the order of G, $|H| \mid |G|$

⇒ For a finite group G, the order of every $a \in G$, divides $|G|$.

⇒ Let G be a finite group. Then $a^{|G|} = e$ for every $a \in G$.

⇒ Every group of prime order is cyclic and every element except the neutral element is a generator.

The group $\mathbb{Z}_m^*$ and Euler's function

While $\mathbb{Z}_m = \{0, \dots, m-1\}$ is a group in respect to $\oplus$ it is not for $\odot$. We need to exclude all elements without inverse.

$$\mathbb{Z}_m^* \overset{\text{def}}{=} \{a \in \mathbb{Z}_m \mid \gcd(a,m) = 1\}$$ (excludes all with no inverse)

The Euler function $\varphi : \mathbb{Z}^+ \to \mathbb{Z}^+$ is defined as the cardinality of $\mathbb{Z}_m^*$:

$$\varphi(m) = |\mathbb{Z}_m^*|$$

If the prime factorization of m is:

$$m = \prod_{i=1}^{r} p_i^{e_i} \Rightarrow \varphi(m) = \prod_{i=1}^{r} (p_i - 1) p_i^{e_i - 1}$$

$\langle \mathbb{Z}_m^*; \odot, ^{-1}, 1 \rangle$ is a group.

If $m \geq 2$ and $\gcd(a,m) = 1$, then $a^{\varphi(m)} \equiv_m 1$. In particular for every prime p and a not divisable by p, $a^{p-1} \equiv_p 1$

$\mathbb{Z}_m^*$ is cyclic if and only if $m = 2$, $m = 4$, $m = p^e$ or $m = 2p^e$, where p is a odd prime and $e \geq 1$.

RSA - Encryption

RSA public-key cryptosystem is another widely used protocol for secure communication.

To understand how it works, we need the following theorem, that follows from Lagrange's theorem.

Let G be some finite group (multiplicatively written) and let $e \in \mathbb{Z}$ be relatively prime to $|G|$. The function $f: x \to x^e$ is a bijection and the **e-th root** of $y \in G$, namely $x \in G$ satisfying $x^e = y$ is

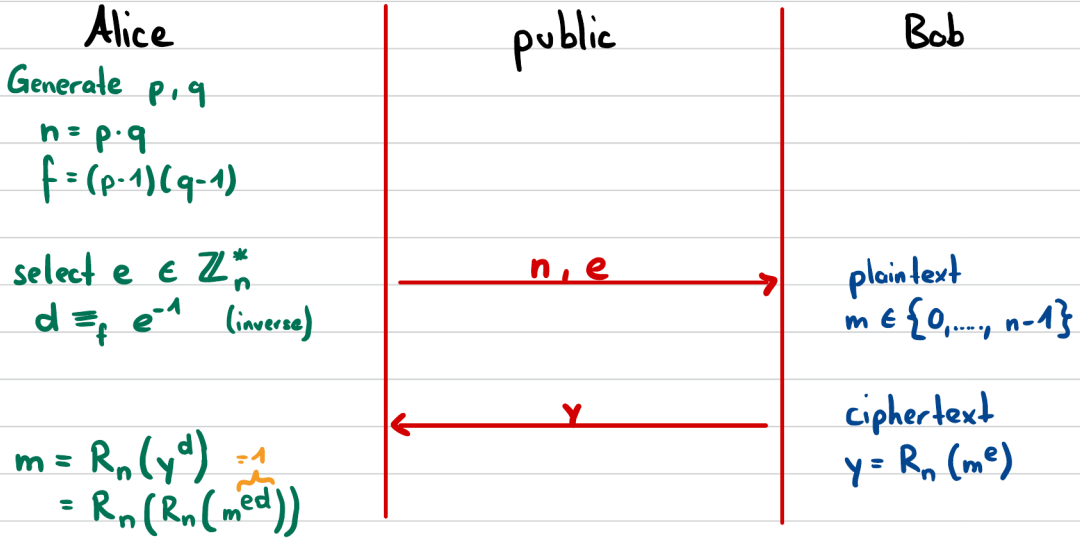
$$x = y^d$$

where d is the multiplicative inverse of e modulo $|G|$,

$$ed \equiv_{|G|} 1$$

If $|G|$ is known, then d can be computed from $ed \equiv_{|G|} 1$ with the extended Euclidean algorithm. But there is no general method to compute it without knowing $|G|$.

With that we can look at how RSA works. In a group $\mathbb{Z}_n^*$, where $n = p \cdot q$ is the product of two large primes, the order $|\mathbb{Z}_n^*| = \varphi(n) = (p-1)(q-1)$ can only be computed knowing p and q.



Rings

Definition of a Ring: A ring $\langle R; +, -, 0, \cdot, 1\rangle$ is an algebra for which

i. $\langle R; +, -, 0\rangle$ is a commutative group $(a+b=b+a)$
ii. $\langle R; \cdot, 1\rangle$ is a monoid
iii. $a(b+c) = ab + cd$ and $(b+c)a = ba + ca$ for all $a, b, c \in R$
(distributive law)

Examples are $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$

It holds that for all $a, b \in R$:

i. $0a = 0$
ii. $(-a)b = -(ab)$
iii. $(-a)(-b) = ab$
iv. if R is non-trivial then $1 \neq 0$

The characteristic of a ring is the order of 1 (the number 1) in the additive group if it is finite, otherwise it is 0. The characteristic gets denoted char(R).

Divisors (R denotes a commutative ring)

For $a,b \in R$, $a \neq 0$, if a divides b, $a|b$, there exists a $c \in R$ such that $b = a \cdot c$. b is a multiple and a is a divisor.

Note that every $a \neq 0$ divides 0 and 1/-1 divide every b.

$\Rightarrow$ i) $a|b$ and $b|c$ then $a|c$ (transitivity)
ii) $a|b$ then $a|b \cdot c$ for all c
iii) $a|b$ and $a|c$ then $a|(b+c)$

For $a,b \in R$ not both 0, there exists a gcd d such that

$$d|a \wedge d|b \wedge \forall c((c|a \wedge c|b) \rightarrow c|d)$$

Units, Zerodivisors and Integral Domains

zerodivisor: element $a \neq 0$ such that $ab = 0$ for some $b \in R, b \neq 0$.

unit: element in R that is invertible, i.e. $uv = vu = 1$, we write $v = u^{-1}$. The set of all invertibles is denoted R^* and is a multiplicative group.

integral domain: a ring (non trivial) without zerodivisor. $\forall a \forall b\ (ab=0 \rightarrow a=0 \vee b=0)$

↳ In a integral domain if $a|b$ there exist a unique c such that $a \cdot c = b$, called **quotient**.

Polynomial Rings

A **polynomial** $a(x)$ over a ring R is of form:

$$a(x) = \underset{R \ni}{a_d} \cdot x^d + a_{d-1} \cdot x^{d-1} + \ldots + a_0 \cdot x^0 = \sum_{i=0}^{d} a_i x^i$$

for some non-negative integer d. The **degree** of $a(x)$, $\deg(a(x))$, is the greatest i for which $a_i \neq 0$. For the polynomial 0, the degree is minus infinity. Let $R_{[x]}$ denote the set of polynomials (in x) over R.

Polynomials can be understood as finite lists $(a_0, a_1, \ldots, a_d)$ of elements in R.

⇒ For any ring R, $R_{[x]}$ is a ring. If D is a integral domain, so is $D_{[x]}$.

The units of $D_{[x]}$ are the constant polynomials that are units of D: $D^*_{[x]} = D^*$

degree = 0

Fields

A field is a nontrivial commutative ring F in which every nonzero is a unit, i.e. $F^* = F \setminus \{0\}$

$\mathbb{Z}_p$ is a field if and only if p is prime. We can denote this by $GF(p)$, standing for Galois Field.

A field is an integral domain. (no zero divisors)

Polynomial over a field F are of special interest, since they have similar properties in common with the integers

A polynomial $a(x) \in F[x]$ (polynomial over x) is called monic if the leading coefficient is 1.

Example. $GF(3)[x]$: $1 \cdot x^3 + 2x^2 + x = (x+1)(x^2+x)$ (monic)

A polynomial $a(x) \in F[x]$ with degree at least 1 is called irreducible if it is divisable only by constant polynomials and by constant multiples of $a(x)$. (corresponds to primality in $\mathbb{Z}$)

Note that when multiplying polynomials the degree gets added.

The concept of gcd can also be carried over from $\mathbb{Z}$ to $F[x]$.
The monic polynomial $g(x)$ of largest degree is the gcd of $a(x)$ and $b(x)$ if $g(x) \mid a(x)$ and $g(x) \mid b(x)$.

Also the concept of remainders can be carried over. For any $a(x)$ and $b(x) \neq 0$ in $F[x]$ there exists a unique monic $q(x)$ quotient and a unique $r(x)$ remainder such that

$$a(x) = b(x) \cdot q(x) + r(x) \quad \text{and} \quad \deg(r(x)) < \deg(b(x))$$

Analogies between $\mathbb{Z}$ and $F[x]$*

These are some abstractions underlying both $\mathbb{Z}$ and $F[x]$:

- a and b are associates if $a = u \cdot b$ for some unit u, denoted $a \sim b$.
- In an integral domain, a non-unit $p \in D \setminus \{0\}$ is irreducible if, whenever $p = ab$, then either a or b is a unit.
- $a \sim b \iff a \mid b \wedge b \mid a$
- A Euclidean domain is an integral domain D with a degree function $d: D \setminus \{0\} \to \mathbb{N}$ such that every a and $b \neq 0$ in D there exists q and r such that $a = bq + r$ and $d(r) < d(b)$ or $r = 0$.
- In a euclidean domain every element can be factored uniquely (up to taking associates) into irreductible elements.

Polynomials as functions

For a ring R, a polynomial $a(x) \in R[x]$ can be interpreted as a function $R \to R$ by evaluating $a(x)$ at $\alpha \in R$. This gives us a function $R \to R: \alpha \to a(\alpha)$.

A $\alpha \in R$ so that $a(\alpha) = 0$ is called a root of $a(x)$. For a field F, $\alpha \in F$ is a root of $a(x)$ iff $x - \alpha$ divides $a(x)$.

If a polynomial of degree ≥ 2 has no roots, it is irreducible.

If α is a root, then its multiplicity is the highest power of $x - \alpha$ that divides $a(x)$.

Example: GF(2): $x^5 + x$ has root 1 with multiplicity 4, since $(x+1)^4$ divides $x^5 + x$.

$$x - 1 = x + 1 \quad \to \quad x^5 + x \;:\; (x+1)^4 = 0$$

Polynom interpolation

A polynomial $a(x) \in F[x]$ of degree $\leq d$ can be uniquely determined by any $d+1$ values of $a(x)$, i.e. $a(\alpha_1), \ldots, a(\alpha_{d+1})$ for any distinct $\alpha_1, \ldots, \alpha_{d+1}$.

↳ same as in $\mathbb{R}$!

Example: let $\beta_i = a(\alpha_i)$

$$u_i = \frac{(x-\alpha_1)\cdots(x-\alpha_{i-1})(x-\alpha_{i+1})\cdots(x-\alpha_{d+1})}{(\alpha_i-\alpha_1)\cdots(\alpha_i-\alpha_{i-1})(\alpha_i-\alpha_{i+1})\cdots(\alpha_i-\alpha_{d+1})}$$

$$a(x) = \sum_{i=1}^{d+1} \beta_i u_i(x)$$

	$a(x)$	$u_1(x)$	$u_2(x)$	...	$u_{d+1}(x)$
α_1	β_1	1	0	...	0
α_2	β_2	0	1	...	0
α_3	β_3	0	0	...	0
⋮	⋮	⋮	⋮		⋮
α_{d+1}	β_{d+1}	0	0	...	1

Finite Fields

We have already seen the finite field GF(p), but what other finite fields are there.

$F[x]_{m(x)}$

The same way as in $\mathbb{Z}$ we can compute $F[x]$ modulo a polynomial $m(x)$.

$$a(x), b(x) \in F[x] \qquad a(x) \equiv_m b(x) \overset{\text{def}}{\iff} m(x) \mid (a(x)-b(x))$$

Congruence modulo $m(x)$ is a equivalence relation on $F[x]$, and each equivalence class has a unique representation of degree $< \deg(m(x))$.

Let $m(x)$ be a polynomial of degree d over F. Then

$$F[x]_{m(x)} \overset{\text{def}}{\iff} \{a(x) \in F[x] \mid \deg(a(x)) < d\}$$

We can state a simple fact about the cardinality of $F[x]_{m(x)}$, when F is finite with q element and $\deg(m(x)) = d \Rightarrow |F[x]_{m(x)}| = q^d$

$F[x]_{m(x)}$ is a ring with respect to addition and multiplication modulo $m(x)$.

$a(x)\,b(x) \equiv_{m(x)} 1$ only has a so $b(x)$ if $\gcd(a(x), m(x)) = 1$
The solution is unique or in other words,

$$F[x]^*_{m(x)} = \{a(x) \in F[x]_{m(x)} \mid \gcd(a(x), m(x)) = 1\}$$

Constructing Extension Fields

The ring $F[x]_{m(x)}$ is a field if and only if $m(x)$ is irreducible (prime).

⇒ The ring $\mathbb{R}[x]_{x^2+1}$ is isomorphic to $\mathbb{C}$.

Error - Correcting Codes

Finite fields are important in CS, one of the applications is ECC.

To encode a message we use a injective function that maps a list $(a_0, \dots, a_k) \in A^k$ of $k+1$ symbols to a list $(c_0, \dots, c_n) \in A^n$ of $n > k+1$ encoded symbols, called a codeword.

↑ A = alphabet

$$E: A^k \to A^n : (a_0, \dots a_k) \longmapsto E(a_0, \dots, a_k) = (c_0, \dots, c_n)$$

For such encoding functions E, one often considers the set

$$C = \mathrm{Im}(E) = \{E(a_0, \ldots, a_k) \mid a_0, \ldots, a_k \in A\}$$

of codewords, which is called an error-correction code.

An (n,k)-ecc over an alphabet A with $|A| = q$ is a subset of A^n with cardinality q^k

In CS one often uses $A = \{0,1\}$ or $A = \{0,1\}^8$.

The Hamming distance of two strings of equal length, is the number of positions in which they are different.

The minimum distance of an ECC is the minimum distance between any two codewords.

Decoding

A decoding function D for an (n,k)-ECC is a function $D: A^n \to A^k$.

A good decoding function takes an input $(r_0, \ldots, r_n) \in A^n$ and decodes it to the most plausible information $(a_0, \ldots, a_k)$.

A decoding function can be characterized in terms of errors t that can be corrected.

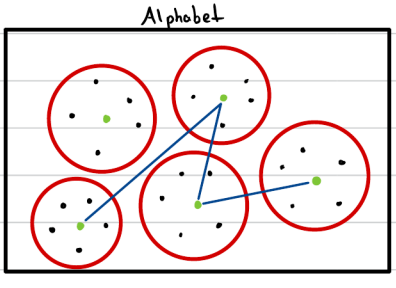
A decoding function D is t-error correcting for encoding function E if for any $(a_0, \ldots, a_k)$

$$D((r_0, \ldots, r_n)) = (a_0, \ldots, a_k)$$

for any $(r_0, \ldots, r_n)$ with Hamming distance at most t from $E((a_0, \ldots, a_k))$. A code C is t-error correcting if there exists E and D such that $C = \text{Im}(E)$ where D is t-error correcting

A code C with minimum distance d is t-error correcting if and only if:

$$\boxed{d \geq 2t+1}$$



codewords t-error-correcting

Hamming distance

Codes based on Polynomial Evaluation

A very powerful class of codes can be obtained by polynomial interpolation if A has a field structure, i.e. $A = GF(q)$.

Consider the encoding function:

$$E((a_0, \ldots, a_k)) = (a(\alpha_0), \ldots, a(\alpha_n))$$

where $a(x)$ is the polynomial $a_k x^k + \ldots + a_1 x + a_0$. This code has a minimum distance of $n-k+1$.

Ch.6 Logic

In this part we discuss the foundations of logic in a mathematically rigorous manner. On a fundamental level, the goal of logic is to express mathematical statements and express and verify proofs of such statements.

There are multiple different logics with different notations, but the concepts we discuss are general.

Proof Systems

For a formal treatment we need a well-defined syntax. Typically we express things as strings over some alphabet Σ.

In this section we study two types of mathematical objects: statements and proofs for these type of statements. By a type of statement we mean for example the class of statements of the form n is prime.

Let $S \subseteq \Sigma^*$ be the set of mathematical statements of a type and $P \subseteq \Sigma^*$ be the set of proof strings. Every statement $s \in S$ is either true or false, this can be expressed by a truth function:

$$T : S \to \{0,1\}$$

This function defines the meaning (semantic) of objects in S.

A $p \in P$ is either a proof for $s \in S$ or not. This is expressed by a verification function:

$$\phi : S \times P \longrightarrow \{0,1\}$$

Without a strong loss of generality we can say that $S = P = \{0,1\}^*$.

A proof system is quadruple $\Pi = (S, P, \tau, \phi)$.

A proof system $\Pi = (S, P, \tau, \phi)$ is sound if no false statements have a proof. For all $s \in S$ for which there exists a $p \in P$ with $\phi(s,p) = 1$, we have $\tau(s) = 1$.
It is complete if every true statement has a proof. For all $s \in S$ with $\tau(s) = 1$, there exists a $p \in P$ with $\phi(s,p) = 1$.

In addition ϕ should be efficient to compute.

Ex. see p.130

Important:
- while ϕ must be efficient, proof generation generaly is not.
- a proof system is restricted to one type of statement.
- ϕ can proceed in many ways
- Statements and their negation are asymetric, meaning they are of different difficulty to prove, if even possible.

Elementary General Concepts of Logic

The goal is to provide a proof system that can be applied to a very larg class of mathematical statements.

However a proof system can never be applied to all possible mathematical statements.

A proof consists of a sequence of syntactic steps, called a derivation or a deduction. Each step consists of applying one of the allowed syntactic rules. The set of allowed syntactic rules is called a calculus.

Syntax, Semantics, Interpretation, Model

The syntax is the alphabet Λ of allowed symbols and specifies which strings in Λ^* are formulas (correct).

Ex. $A \vee B$ ✓ $A \vee$ ✗ — in each logic some symbols are understood as variables that can take a value in a domain.

The semantics define under which condition a formula is true or false.

A variable or predicate is called free if it needs to be assigned a fixed value before the formula has a truth value. This assignment is called a interpretation.

A semantic defines a function free so that $i \in free(F) \Rightarrow f_i$ is free.

A interpretation consists of a set $Z \subseteq \Lambda$ of symbols of Λ, a domain for each symbol in Z and a function that assigns to each symbol in Z a value in its associated domain.

Often such a domain is defined in terms of a so-called universe U.

A interpretation is called suited for a formula F if it assigns a value to all free symbols in F.

A semantic also defines a function $\sigma(F, A)$ assigning to each formula F and each interpretation A suitable for F, a truth value in $\{0,1\}$. One often simply writes $A(F)$.

A suitable interpretation A for which $A(F)=1$ is called a model for F, $A \models F$, if M is a set of formulas we write $A \models M$. If A is not a model we write $A \not\models F$

Satisfiability, Tautology, Consequence, Equivalence

A formula F is called satisfiable if there exists a model for F and unsatisfiable otherwise. Unsatisfiable is denoted $\bot$.

A formula F is a tautology T if it is true for every suitable interpretation.

A formula G is a logical consequence of F, $F \models G$, if every interpretation suitable for F and G, which is a model for F is also a model for G.

If F is a tautology one only writes $\models F$.

Two formulas F and G are equivalent, $F \equiv G$, if $F \models G$ and $G \models F$.

The Logical Operators $\vee$, $\wedge$ and $\neg$

If F and G are formulas, then $\neg F$, $(F \vee G)$ and $(F \wedge G)$ are formulas.

disjunction conjuction

$\leftrightarrow$ and $\rightarrow$ are simply notational conventions. The semantics of these logical operators is defined as follows:

$$\begin{array}{lll} \mathcal{A}(\neg F) = 1 & \text{iff} & \mathcal{A}(F) = 0 \\ \mathcal{A}(F \wedge G) = 1 & \text{iff} & \mathcal{A}(F) = 1 \text{ and } \mathcal{A}(G) = 1 \\ \mathcal{A}(F \vee G) = 1 & \text{iff} & \mathcal{A}(F) = 1 \text{ or } \mathcal{A}(G) = 1 \end{array}$$

Some basic equivalences can now be stated:

Lemma 6.1. *For any formulas F, G, and H we have*

1) $F \wedge F \equiv F$ *and* $F \vee F \equiv F$ *(idempotence);*
2) $F \wedge G \equiv G \wedge F$ *and* $F \vee G \equiv G \vee F$ *(commutativity);*
3) $(F \wedge G) \wedge H \equiv F \wedge (G \wedge H)$ *and* $(F \vee G) \vee H \equiv F \vee (G \vee H)$ *(associativity);*
4) $F \wedge (F \vee G) \equiv F$ *and* $F \vee (F \wedge G) \equiv F$ *(absorption);*
5) $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$ *(distributive law);*
6) $F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H)$ *(distributive law);*
7) $\neg\neg F \equiv F$ *(double negation);*
8) $\neg(F \wedge G) \equiv \neg F \vee \neg G$ *and* $\neg(F \vee G) \equiv \neg F \wedge \neg G$ *(de Morgan's rules);*
9) $F \vee \top \equiv \top$ *and* $F \wedge \top \equiv F$ *(tautology rules);*
10) $F \vee \bot \equiv F$ *and* $F \wedge \bot \equiv \bot$ *(unsatisfiability rules).*
11) $F \vee \neg F \equiv \top$ *and* $F \wedge \neg F \equiv \bot$.

Logical Consequences vs. Unsatisfiability

F is only a tautology if $\neg F$ is unsatisfiable.

The following statements are equivalent:

- $\{F_1, \dots, F_k\} \models G$
- $\{F_1 \wedge \dots \wedge F_k\} \rightarrow G$ is a tautology
- $\{F_1, \dots, F_k, \neg G\}$ is unsatisfiable

There are four types of statements one may want to prove:

- Theorems
- Statements about F or M
- The statement $A \models F$ for given A, F
- Statements about the logic, for example that a calculus is sound.

Logical Calculi

The goal of logic is to provide a framework for expressing and verifying the proof. A proof should be a purely syntactic derivation consisting of simple, easy verifiable step. In each step, a new syntactical object is derived and at the end the desired theorem appears.

A well-defined set of rules to manipulate formulas is called a calculus (pl. calculi).

Hilbert-Style Calculi

In Hilbert-style calculus the syntactical-objects that are manipulated are formulas.

A derivation rule is a rule for deriving a formula from a set of set of formulas (precondition). We write

$$\{F_1, F_2, \ldots, F_n\} \vdash_R G \quad \text{or} \quad \frac{F_1 \; F_2 \; \ldots \; F_n}{G} \; (R)$$

if G can be derived from $\{F_1, F_2, \ldots, F_n\}$ by R.

The application of a derivation rule R to set M of formulas means:

1. Select $N \subseteq M$
2. For the placeholder in R, select formulas $\in N$ such that $N \vdash_R G$
3. Add G to M

A calculus K is a finite set of derivation rules $K = \{R_1, R_2, \ldots, R_n\}$.

A derivation of a formula G from a set M in a calculus K is a finite sequence of applications of rules in K.

- $M_0 := M$
- $M_i := M_{i-1} \cup \{G_i\}$ for $1 \leq i \leq n$ where $N \vdash_{R_j} G_i$ for some $N \subseteq M_{i-1}$ and some $R_j \in K$.
- $G_n = G$

We write $M \vdash_K G$.

A derivation rule R is correct if for every set M and every formula F,

$$M \vdash_R F \Rightarrow M \models F$$

if this is the case for every rule in a calculus, then the calculus is sound.

If for every M and F, $M \models F \Rightarrow M \vdash_K F$, then the calculus K is complete.

Derivations from Assumptions

A natural way to prove $F \to G$ is to assume F and to derive G.

If $F \vdash_K G$ holds for a sound calculus, then $\models (F \to G)$. More generally if $\{F_1, \ldots, F_n\} \vdash G$, then $\models (\{F_1, \ldots, F_n\} \to G)$.

For a given calculus, one can prove new derivation rules. In general if $\{F_1, \ldots, F_n\} \vdash_K G$, then one could extend K by the rule $\{F_1, \ldots, F_n\} \vdash G$.

Propositonal Logic