1 Mathematical Reasoning, Proofs, and a First Approach to Logic

1.1 Propositions and Logical Formulas

D. (Proposition) is a (mathematical) statement that is either *true* or *false*.

D. (Logical values) (constants) "true" and "false" are usually denoted as 1 and 0.

D. (Formula) correctly formed expression involving propositional symbols.
D. (Conjunction) (logical AND) $A \wedge B$
D. (Disjunction) (logical OR) $A \vee B$
D. (Implication) $A \to B :\iff \neg A \vee B$
D. (Two-sided imp.) $A \leftrightarrow B$: $\Leftrightarrow (A \to B) \wedge (B \to A)$
D. (Operator Precedence) $\neg, \wedge, \vee, \to, \leftrightarrow$

D. (Constant function symbols)
$\top$: constant 1 (true) $\quad$ $\bot$: constant 0 (false)

D. (Equivalence) Two formulas F and G are *equivalent*, denoted $F \iff G$ (or also $F \equiv G$) if they correspond to the same function (table).

D. (Tautology) A formula F that is true for all truth assignments of the involved symbols.

D. (Satisfyability) A formula F is
Satisfiable: F is true for at least one truth assignment
Unsatisfiable: otherwise

L. F is a tautology iff $\neg F$ is unsatisfiable

T. (Transitivity of Implication)

$$(F \to G) \wedge (G \to H) \implies (F \to H)$$

T. (De Morgan)

$$\neg(A \wedge B) \iff (\neg A \vee \neg B) \quad \neg(A \vee B) \iff (\neg A \wedge \neg B)$$

1.2 Quantifiers and Predicate Logic

Let us consider a set U as the *universe* in which we want to reason.
D. (k-ary Predicate) P on U is a function $U^k \to \{0,1\}$.
D. For a universe U and a predicate $P(x)$ we define the following logical statements:

- $\forall x\ P(x)$ is the statement that $P(x)$ is true *for all* $x \in U$.
- $\exists x\ P(x)$ is the statement that $P(x)$ is true for some $x \in U$ (at least one).

1.3 Some Proof Patterns and Techniques

T. (Contraposition) $F \to G \iff \neg G \to \neg F$

T. (Modus Ponens) $F \wedge (F \to G) \implies G$
Prove F, and prove $F \to G$ to derive G.

D. (Direct Proof) of an implication $F \to G$
assume F, then derive G (from F, step-by-step).
Com. Recall that a proof of $F \to G$ neither proves that F is true (or a tautology) nor that G is true (or a tautology), only that the implication holds.

D. (Indirect Proof) of an implication $F \to G$
Prove the contraposition: $\neg G \to \neg F$ (via direct proof)

T. (Composition of Implications) The Implication is transitive: If $F \to G$ and $G \to H$ are tautologies [true], then so is $F \to G$.
Com. The theorem implies that more generally, if one proves a statement F_1 as well as implications $F_1 \implies F_2$, $F_2 \implies F_3, \ldots, F_{n-1} \implies F_n$, then one has also proved F_n.

D. (Case Distinction) A proof of a statement by *case distinction* proceeds by defining a complete list of cases (such that one of the cases is guaranteed to occur), and by proving the statement for each case separately.
T. If $F_1 \vee \cdots \vee F_k$ and $F_i \to G$ fo r $1 \le i \le k$ are tautologies [true], then G is also a tautology [true].

D. (Proof by Contradiction) of a statement F
Just assume $\neg F$ to be true and derive from this assumption a false statement. By using the contraposition, you can then show that F must be true. (As long as the steps that were used in between are proven tautologies).
T. If $\neg F \to \bot$ is a tautology [true], then F is also a tautology [true].

D. (Existence Proof) of $\exists x\ P(x)$

- Constructive: demonstrate such an a.
- Non-Constructive: just show the existence of an a without exhibiting such an a.

D. (Inexistence Proof) of $\neg\exists x\ P(x)$

D. (Proof by Counterexample) of $\neg\forall x\ P(x)$
This means that $\exists x\ \neg P(x)$, which corresponds to an existence proof. The a for which $\neg P(a)$ is true is called the *counterexample*.

D. (Proof by Induction) Is used to prove statements of the form $\forall n\ P(n)$ (or $\forall n \ge k\ P(n)$), where $n \in \mathbb{N}$ (or $n \ge k$).

1. *Basis step:* Prove $P(0)$ (or $P(k)$).
2. *Induction hypothesis:* Assume, that there exists an $n \in \mathbb{N}$ for which $P(n)$ holds.
3. *Induction step:* Prove
$$\forall n\ (P(n) \to P(n+1)) \quad (\forall n \ge k)$$
by reducing the problem into several parts, one resembling to the form of the induction hypothesis, then use it.

T. For every predicate P on $\mathbb{N}$ we have

$$P(0) \wedge \forall n\ (P(n) \to P(n+1)) \iff \forall n\ P(n), \text{ or}$$

$$P(k) \wedge \forall n \ge k\ (P(n) \to P(n+1)) \iff \forall n \ge k\ P(n).$$

Com. The proof makes use of the so-called well-ordering principle, which we assume to be a fact.
Fact. The natural numbers are well-ordered, i.e., every non-empty set of natural numbers has a least element.

2 Sets, Relations, and Functions

2.1 Sets and Operations on Sets

We assume that for every object x and a set A it is defined whether x is an *element* of A, denoted $x \in A$, or whether it is not an element of A, denoted $x \notin A$ (instead of $\neg(x \in A)$).

A set can be described by a *defining property* $\{x \in A | P(x)\}$ where P is a predicate on A; or by *listing its elements* $\{a_0, a_1, \ldots\}$.
Sets can themselves be elements of a set, e.g. $A = \{3, \{4\}\}$.

D. (Set Equaility)
$A = B :\iff \forall x\ (x \in A \leftrightarrow x \in B)$
$A = B :\iff (A \subseteq B) \wedge (B \subseteq A)$
Com. The second statement is usaually a good approach to prove that two sets are equal.
Com. Two sets are equal if they contain the same elements, independently of how they are described.

D. (Cardinality) The number of elements in a finite set A, denoted $|A|$.
L. Sets with different (finite) cardinality are different.
L. For finite sets the cardinality of the Cartesian product of some sets is the product of their cardinalities.

D. (Ordered Pair) of two objects a and b, denoted (a, b).
L. We have $(a, b) = (c, d) \implies a = c \wedge b = d$.

D. (Subset) $A \subseteq B :\iff \forall x (x \in A \to x \in B)$
D. (Proper Subset) $A \subset B :\iff A \subseteq B \wedge A \neq B$
L. The subset relation is *transitive*:

$$A \subseteq B \wedge B \subseteq C \Rightarrow A \subseteq C$$

D. (Empty Set) denoted $\emptyset$ or $\{\}$, is the set with no elements, i.e., $\forall x (x \notin \emptyset)$.
L. The empty set is a subset of every set: $\forall A\ (\emptyset \subseteq A)$
L. The empty set is *unique* (same for all).

$$\emptyset \subseteq \emptyset' \wedge \emptyset' \subseteq \emptyset \Rightarrow \emptyset = \emptyset'.$$

The empty set can be used to construct new sets, without using any other predefined objects as elements. (Therefore for any universe the empty set is the same).
It is important not to confuse $\emptyset$ with $\{\emptyset\}$:

$$|\{\emptyset\}| = 1 \qquad |\emptyset| = 0.$$

D. (Power Set) of A, denoted $\mathcal{P}(A)$ or somtimes 2^A, is the set of all subsets of A:

$$\mathcal{P}(A) := \{S | S \subseteq A\}.$$

T. For a finite set with cardinality k, the power set has cardinality 2^k.

D. (Union) $A \cup B := \{x | x \in A \vee x \in B\}$
D. (Intersection) $A \cap B := \{x | x \in A \wedge x \in B\}$
D. We may also unite/intersect a set of sets $\mathcal{A}$ to a single set containing elements of the sets of $\mathcal{A}$:

$$\bigcup \mathcal{A} := \{x | \exists A \in \mathcal{A} : x \in A\}.$$

$$\bigcap \mathcal{A} := \{x | \forall A \in \mathcal{A} : x \in A\}.$$

If the sets in $\mathcal{A}$ are numbered one also may write $\cap_{i \in I} A_i$ and $\cup_{i \in I} A_i$, respectively.

D. (Complement) For a given universe of discourse, U, the *complement* of a set A, denoted $\overline{A}$ (or sometimes A^c) is: $\overline{A} := \{x \in U | x \notin A\}$ or simply $\overline{A} = \{x | x \notin A\}$.

D. (Difference) The *difference* of sets B and A, denoted $B - A$ (or sometimes $B \backslash A$) is the complement of A, relative to B: $B - A := \{x \in B | x \notin A\}$.

T. (Algebra of Sets) For any sets A, B, and C, and a universe U, the following laws hold:

Idempotence:	$A \cap A = A, \quad A \cup A = A$
Commutat.:	$A \cap B = B \cap A, \quad A \cup B = B \cup A$
Associat.:	$A \cap (B \cap C) = (A \cap B) \cap C$
	$A \cup (B \cup C) = (A \cup B) \cup C$
Absorption:	$A \cap (A \cup B) = A, \quad A \cup (A \cap B) = A$
Distribut.:	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Complem.ity:	$A \cap \overline{A} = \emptyset, \quad A \cup \overline{A} = U$
Consistency:	$A \subseteq B \Leftrightarrow A \cap B = A \Leftrightarrow A \cup B = B.$

D. (Cartesian Prod.) $A \times B = \{(a, b) | a \in A \wedge b \in B\}$
More generally, the Cartesian product of k sets $A_1, \ldots, A_k$ is the set of all lists of length k (also called k-tuples) with the i-th component from A_i:

$$\times_{i=1}^{k} A_i = \{(a_1, \ldots, a_k) | a_i \in A_i \text{ for } 1 \leq i \leq k\}$$

2.2 Relations

D. (Relation) A (binary) relation ρ from a set A to a set B (also called an (A, B)-relation) is a subset of $A \times B$. If $A = B$, then ρ is called a relation on A.
Instead of $(a, b) \in \rho$ one usually writes $a\ \rho\ b$, and sometimes we write $a\ \not\rho\ b$ if $(a, b) \notin \rho$.
Com. Two special relations from A to B are the empty relation (i.e., the empty set $\emptyset$) and the complete relation consisting of all pairs (a, b).

D. (Identity Relation) on any set A is defined as $\mathrm{id} = \{(a, a) | a \in A\}$.

D. (Inverse of a Relation) ρ from A to B is denoted $\widehat{\rho}$, such that $a\ \rho\ b \iff b\ \widehat{\rho}\ a$
Matrix: take transpose, Graph: reverse edges

D. (Composition of Relations) Let ρ be a relation from A to B and σ be a relation from B to C. Then the *composition*, denoted $\rho\sigma$ (or also $\rho \circ \sigma$), is the relation from A to C where

$$a\ \rho\sigma\ c :\iff \exists b \in B\colon\ (a\ \rho\ b) \wedge (b\ \sigma\ c)$$

Matrix: "special" multiplication
Graph: natural composition of the graphs, where $a\ \rho\sigma\ c$ if and only if there is a walk from a (over some b) to c.

D. The n-fold composition of a relation ρ on a set A is denoted ρ^n.
Matrix: n-fold "special" multiplication
Graph: creating edges for all possible start and endpoints of walks of length n.

L. (Associativity of Composition) $\rho(\sigma\phi) = (\rho\sigma)\phi$

L. (Inverses of Compositions) $\widehat{\rho\sigma} = \widehat{\sigma}\widehat{\rho}$

D. (Properties of Rel.) A relation ρ on a set A is called

- **reflexive** if: $\forall a \in A\ (a\ \rho\ a)$
  - i.e. if $\mathrm{id} \subseteq \rho$
  - Matrix: all diagonal elements are 1
  - Graph: all vertices have a loop
- **irreflexive** if: $\forall a \in A\ (a\ \not\rho\ a)$

- Matrix: all diagonal elements are 0
- Graph: no loops
- Note that irreflexive is not the logical negation of reflexive.

- **symmetric** if: $\forall a, b \in A\ (a\ \rho\ b \leftrightarrow b\ \rho\ a)$
  - Matrix: symmetric (w. resp. to diag.) $M = M^\top$
  - Graph: If there is a connection between two nodes, then the inverse connection must also exist (or in case of a self-reference, a loop). Hence, the graph can also just be drawn as an undirected graph.
- (**asymm.** if: $\forall a, b \in A\ (a\ \rho\ b \to b\ \not\rho\ a)$ was not treated)
- **antisymm.** if: $\forall a, b \in A (a\ \rho\ b \wedge b\ \rho\ a \to a = b)$
  - equivalently $\forall a, b \in A (a\ \rho\ b \wedge a \neq b \to b\ \not\rho\ a)$
  - **L.** iff $\rho \cap \widetilde{\rho} \subseteq \mathrm{id}$
  - Graph: there is no cycle of length 2, (i.e., there is just one arrow between two nodes (no arrow pointing back) but there may be loops.
  - Matrix: Diagonal 1 or 0, Left and right side of diagonal are the opposites.
  - Note that antisymmetric is not the logical negation of symmetric (we'd call the negation asymmetric).
- **transitive** if: $\forall a, b, c \in A (a\ \rho\ b \wedge b\ \rho\ c \to a\ \rho\ c)$
  - **L.** iff $\rho^2 \subseteq \rho$
  - Graph: Must look like a graph resulting from the transitive hull determination algorithm.

Number of Relations over a set of cardinality n:

- 2^{n^2} relations.
- 2^{n^2-n} reflexive (or irreflexive) relations.
- $2^{\sum_{k=0}^{n} k} = 2^{\frac{n(n+1)}{2}} = 2^{\binom{n+1}{2}}$ symmetric relations.
- $2^{\binom{n}{2}}$ symmetric and reflexive (or irreflexive) relations.
- $2^n 3^{\binom{n}{2}}$ antisymmetric relations.
- $3^{\binom{n}{2}}$ antisymmetric and reflexive (or irreflexive) relations.
- 2^n antisymmetric and symmetric relations (subset of id).

There is no simple formula for the number of transitive relations. First use the other constraints and then try all possibilities and verify if they are transitive.
In general it helps to think about the conditions that the matrix representation must satisfy.

D. (Transitive Closure) of a relation ρ, denoted ρ^*

$$\rho^* = \cup_{n=1}^{\infty} \rho^n$$

Matrix: OR-ing the results of all infinite "special" multiplication matrices
Graph: building transitive hull (i.e. adding an edge between the start and endpoint of all possible walks)

2.3 Equivalence Relations

D. (Equivalence Relation) is a relation that is *reflexive*, *symmetric*, and *transitive*.
D. (Equivalence Class) for an equivalence relation θ on a set A and for $a \in A$, is the set of elements of A that are equivalent to a. Denoted

$$[a]_\theta := \{b \in A | b\ \theta\ a\}$$

Com. Two trivial equivalence relations on a set A are the complete relation $A \times A$ for which there is only one equivalence class A, and the equality relation (=) for which the equivalence classes are all singletons $\{a\}$ for $a \in A$.

L. The intersection of two equivalence relations is an equivalence relation

---

D. (Partition) of a set A is a set $\{S_i \subseteq A\}_{i \in I}$ of mutually disjoint subsets of A that cover A, i.e.,

$$S_i \cap S_j = \emptyset \text{ for } i \neq j \quad \text{and} \quad \cup_{i \in I} S_i = A.$$

D. (Set of Equivalence Classes) of an equivalence relation θ, denoted by $A/\theta := \{[a]_\theta | a \in A\}$, is called the *quotient set of A by θ*, or simply A *modulo* θ, or $A \mod \theta$.
T. The set A/θ of equivalence classes of an equivalence relation θ on A is a partition of A

2.4 Partial Order Relations

D. (Partial Order) on a set A is a relation that is reflexive, antisymmetric, and transitive.
D. (Partially Ordered Set) or poset is a set A, together with a partial order $\preceq$ on A, denoted as $(A; \preceq)$.
Com. Graph: Non cycles, but this is not a complete characterisation

---

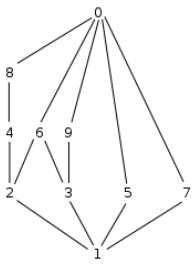
D. For a partial order relation $\preceq$ we can define the relation $a \prec b$ as follows:

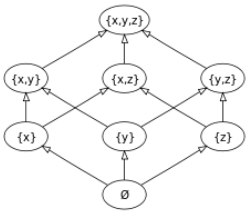
$$a \prec b :\Longleftrightarrow a \preceq b \wedge a \neq b.$$

D. (Comparability) For a poset $(A; \preceq)$, two elements a and b are *comparable* if $a \preceq b$ or $b \preceq a$ (is true); otherwise they are called *incomparable*.
D. (Total Order) If any two elements of a poset $(A; \preceq)$ are comparable, then A is called *totally ordered* (or *linearly ordered*) by $\preceq$.
Example: Totally ordered: $(\mathbb{Z}, \leq)$. Not totally ordered: $(P(A); \subseteq)$ for $|A| \geq 2$; or $(\mathbb{Z}, |)$.
D. (Well Ordered) is a poset $(A; \preceq)$ if it is totally ordered and if every non-empty subset of A has a least element.
Com. Note that every totally ordered finite poset is well-ordered. The property of being well-ordered is of interest only for infinite posets.

D. (Covering) In a poset $(A; \preceq)$ an element b is said to *cover* an element a if $a \prec b$ and there exists no c with $a \prec c$ and $c \prec b$ (i.e., between a and b).
Example: company, direct supervisor

D. Hasse Diagram only contains the covering edges. direction can be omitted, it's assumed that edges point uwpards.
Example: The pictures show the hasse diagrams of the posets $(\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}; |)$ and $(P(\{x, y, z\}); \subseteq)$. Note that the left one is a lattice.





T. (Cartesian product of posets) For given posets $(A;\preceq)$ and $(B;\sqsubseteq)$ the relation $\leq$ is defined on $A \times B$ by

$$(a_1, b_1) \leq (a_2, b_2) :\Longleftrightarrow a_1 \preceq a_2 \wedge b_1 \sqsubseteq b_2$$

is a partial order relation.

T. (Lexicographic Order) For given posets $(A;\preceq)$ and $(B;\sqsubseteq)$, the relation $\leq_{\text{lex}}$ defined on $A \times B$ by

$$(a_1, b_1) \leq_{\text{lex}} (a_2, b_2) :\Longleftrightarrow a_1 \prec a_2 \vee (a_1 = a_2 \wedge b_1 \sqsubseteq b_2)$$

is a partial order relation.

L. If both $(A;\preceq)$ and $(B;\sqsubseteq)$ are totally ordered, then so is the lexicographic order $A \times B$, e.g. $(A \times B; \leq_{\text{lex}})$.

D. (Special Elements) Let $(A;\preceq)$ be a poset, and let $S \subseteq A$ be some subset of A. Then

1. $a \in S$ is **a** *minimal (maximal) element* of S if there exists no $b \in S$ with $b \prec a$ ($b \succ a$).
2. $a \in S$ is **the** *least (greatest) element* of S if $a \preceq b$ ($a \succeq b$) for all $b \in S$.
3. $a \in A$ is **a** *lower (upper) bound* of S if $a \preceq b$ ($a \succeq b$) for all $b \in S$.
4. $a \in A$ is **the** *greatest lower bound* $\text{glb}(S)$ *(least upper bound* $\text{lub}(S)$*)* of S if a is the greatest (least) element of the set of all lower (upper) bounds of S.

D. (Meet and Join) Let $(A;\preceq)$ be a poset. If a and b (i.e., the set $\{a, b\} \subseteq A$) have a greatest lower bound, then it is called the *meet* of a and b, often denoted $a \wedge b$. If a and b have a least upper bound, then it is called the *join* of a and b, often denoted $a \vee b$.

D. (Lattice) A poset $(A;\preceq)$ in which every pair of elements has a meet and a join is called a *lattice*.

2.5 Functions

D. (Function) A function $f\colon A \to B$ from a *domain* A to a *codomain* B is a relation from A to B with special properties (using the relation notation $a\ f\ b$):

1. $\forall a \in A\ \exists b \in B : a\ f\ b$
   (f is totally defined, "all x-es are assigned"),
2. $\forall a \in A\ \forall b, b' \in B : a\ f\ b \wedge a\ f\ b' \to b = b'$
   (f is well-defined, "unique assignment for each x").

D. (Set of all Functions) from $A \to B$ is denoted as B^A.

D. (Partial Function) is a relation on $A \times B$ such that condition 2. above holds.

D. (Function Equality) Two (partial) functions with common domain A and codomain B are *equal* if they are equal as relations (i.e., as sets).

Com. $f = g$ is equivalent to saying that the function values of f and g agree for all arguments (including, in case of partial functions, whether or not it is defined).

D. (Image of a Set) For a function $f : A \to B$ and a subset S of A, the *image* of S under f, denoted $f(S)$, is the set

$$f(S) := \{f(a) | a \in S\}.$$

D. (Image) The subset $f(A)$ of B is called the *image* (or *range*) of f and is also denoted $\text{im}(f)$.

D. (Preimage of a Set) For a subset T of B, the *inverse image* (or *preimage*) of T, denoted $f^{-1}(T)$, is the set of values in A that map into T:

$$f^{-1}(T) := \{a \in A | f(a) \in T\}.$$

D. (Function Properties) A function $f\colon A \to B$ is called

1. *injective* if $\forall a, b \in A\colon\ a \neq b \to f(a) \neq f(b)$
2. *surjective* if $\forall b \in B\ \exists a \in A\colon\ f(a) = b$
3. *bijective* if it is both injective and surjective.

D. (Inverse Function) For a bijective function f, the inverse (as a relation see Definition 3.12) is also a function and is called the inverse function of f, usually denoted as f^{-1}.

D. (Composition) The *composition* of a function $f\colon A \to B$ and a function $g\colon B \to C$, denoted by $g \circ f$ or simply gf, is defined by $(g \circ f)(a) = g(f(a))$.

L. Function composition is associative (since it's a relation)

$$(h \circ g) \circ f = h \circ (g \circ f).$$

3 Combinatorics and Counting

Enumeration of a set means *listing the lements in a systematic manner*, and *counting* means *computing the cardinality* of a set.

3.1 Basic Counting Principles

Addition Principle: The cardinality of the union of n disjoint finite sets $A_1, \ldots, A_n$ is equal to the sum of the cardinalities. $\forall i, j, 1 \leq i < j \leq n$:

$$A_i \cap A_j = \emptyset \Longrightarrow |A_1 \cup \cdots \cup A_n| = \sum_{i=1}^{n} |A_i|.$$

Multiplication Principle: It is an obvious fact that $|A \times B| = |A| \cdot |B|$ and more generally, or finite sets $A_1, \ldots, A_n$ the following holds:

$$|A_1 \times \cdots \times A_n| = \prod_{i=1}^{n} |A_i|.$$

Bijection Principle: If there is a bijection (or one-to-one correspondence) between the finite sets A and B, then $|A| = |B|$.

Inclusion-Exclusion Principle: Is a generalisation of the addition principle, when two or more sets are not (necessarily) disjoint.

For any finite sets $A_1, \ldots, A_n$,

$$|A_1 \cup \cdots \cup A_n| = \sum_{i=1}^{n} |A_i| - \sum_{1 \leq i_1 < i_2 \leq n} |A_{i_1} \cap A_{i_2}| + \sum_{1 \leq i_1 < i_2 < i_3 \leq n} |A_{i_1} \cap A_{i_2} \cap A_{i_3}| - \ldots + (-1)^{n-1} |A_1 \cap \cdots \cap A_n|$$

T. (Bonferroni inequalities) They state that if the alternating sum in the inclusion-exclusion principle is stopped after counting the sizes of the unions of k sets, then this is a lower or upper bound for $|A_1 \cup \cdots \cup A_n|$, depending on whether the next sign would be + or -, respectively, i.e., whether k is even or odd. For instance:

$$|A_1 \cup \cdots \cup A_n| \geq \sum_{i=1}^{n} |A_i| - \sum_{1 \leq i_1 < i_2 \leq n} |A_{i_1} \cap A_{i_2}|.$$

D. (Factorial) $k! := k \cdot (k-1) \cdot (k-2) \cdots 2 \cdot 1$

D. $n^{\underline{k}} := n(n-1)\cdots(n-k+1) = \prod_{i=0}^{k-1}(n-i) = \frac{n!}{(n-k)!}$

D. (Bin. Coeff.) $\binom{n}{k} := \frac{n^{\underline{k}}}{k!} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\cdots(n-k+1)}{k!}$

$\binom{n}{0} = \binom{n}{n} = 1$

for $k < 0$ and $k > n$ we define $\binom{n}{k} := 0$
Can be constructed using Pascal's Triangle with tip $\binom{0}{0}$.

Drawing Elements from a Set: The table shows the umber of possibilities for selecting k elements from a set of size n (the example set is $\{a, b, c\}$ thus $n = 3$; and $k = 2$):

	Ordered		Unordered	
	Number	Examples	Number	Examples
w. rep.	n^k	aa ab ac ba bb bc ca cb cc	$\binom{n+k-1}{k}$	aa ab ac bb bc cc
w.o. rep.	$n^{\underline{k}}$	ab ac ba bc ca cb	$\binom{n}{k}$	ab ac bc

---

Double-Counting Principle: Consider the problem of counting a subset S of $A \times B$ (wich can also be interpreted as a relation). We can count S in two different ways, either by determining for each $a \in A$ the number m_a of $b \in B$ such that $(a, b) \in S$, or by determining for each $b \in B$ the number n_b of $a \in A$ such that $(a, b) \in S$. Then

$$|S| = \sum_{a \in A} m_a = \sum_{b \in B} n_b$$

The easiest way to think about this principle is probably to consider the matrix representation of S (considered as a relation). Then m_a is the number of 1's in the row for a, and n_b is the number of 1's in the column for b.

Pigeonhole Principle: If a set of n objects is partitioned into $k < n$ sets, then at least one of these sets contains at least $\lceil \frac{n}{k} \rceil$ objects.

3.2 Binomial Coefficients

L. (Symmetry of Pascal's Triangle) $\binom{n}{k} = \binom{n}{n-k}$

L. (Pascal's Identity) For $n > 0$, $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

T. (Binomial Theorem) For any real (or complex) numbers x and y and for every integer $n \geq 0$,

$$(x + y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k.$$

C. These equalities follow from the binomial theorem by setting $x = y = 1$ as well as $x = 1$ and $y = -1$.

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n \quad \text{and} \quad \sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0.$$

C. (Vandermonde's Identity) Let $k, m, n \geq 0$ be integers with $m + n > 0$. Then, $\binom{m+n}{k} = \sum_{i=0}^{k} \binom{m}{i}\binom{n}{k-i}$ In particular, $\binom{2n}{n} = \sum_{k=0}^{n} \binom{n}{k}^2$.

3.3 Countable and Uncountable Sets

A set that is countable can be *enumerated* (or listed) by a program (even though this would take an unbounded time), while an uncountable set can, in principle, not be enumerated.

D. (Cardinality, Countability)

(i) Two sets A and B *have the same cardinality*, denoted $A \sim B$, if there exists a bijection $A \to B$.

(ii) *The cardinality of B is at least the cardinality of A*, denoted $A \preceq B$, if $A \sim C$ for some subset $C \subseteq B$.

(iii) *B dominates A*, denoted $A \prec B$, if $A \preceq B$ and $A \nsim B$.

(iv) A set A is called *countable* if $A \preceq \mathbb{N}$, and *uncountable* otherwise.

Note that $A \preceq B$ is equivalent to the existence of an injective function $A \to B$ (namely the bijection $A \to C \subseteq B$).

L.

(i) The relation $\sim$ is an equivalence relation.

(ii) The relation $\preceq$ is transitive: $A \preceq B \wedge B \preceq C \implies A \preceq C$.

(iii) $A \subseteq B \implies A \preceq B$.

(iv) A subset of a countable set is also countable: $A \subseteq B \wedge B \preceq \mathbb{N} \implies A \preceq \mathbb{N}$.

(v) $A \preceq B \wedge B \preceq A \implies A \sim B$.

(vi) For two sets A and B, exactly one of $A \prec B$, $A \sim B$, and $B \prec A$ holds.

T. For finite sets A and B, we have $A \sim B$ if and only if $|A| = |B|$. A finite set has never the same cardinality as one of its proper subsets. Somehwat surprisingly, for infinite sets this is possible.

T. A set A is countable if and only if it is finite or if $A \sim \mathbb{N}$ (A is countably infinite).
Com. The theorem can be restated as follows: There is no cardinality level between finite and countably infinite. Or: if $A \prec \mathbb{N}$, then A is finite.

T. The set $\{0, 1\}^* := \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$ of finite binary sequences is countable.
Proof. Bijection to $\mathbb{N}$.

T. The set $\mathbb{N} \times \mathbb{N}$ $(= \mathbb{N}^2)$ of ordered pairs of naural numbers is countable.
Proof. Diagonalisation argument.

C. The Cartesian product $A \times B$ of two countable sets A and B is countable, i.e., $A \preceq \mathbb{N} \wedge B \preceq \mathbb{N} \implies A \times B \preceq \mathbb{N}$.

C. The rational numbers $\mathbb{Q}$ are countable.
Proof. Diagonalisation argument.

T. Let A and A_i for $i \in \mathbb{N}$ be countable sets.

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

(ii) The union $\cup_{i \in \mathbb{N}} A_i$ of a countable list $A_0, A_1, A_2, \ldots$ of countable sets is countable.

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

D. Let $\{0, 1\}^\infty$ denote the set of semi-infinite binary sequences. Note that semi-infinite means: bounded in one direction (left), unbounded in the other (right).

T. The set $\{0, 1\}^\infty$ is uncountable.
TODO: useful comment

L. If A is uncountable and $A \preceq B$, then B is uncountable, i.e.,

$$A \npreceq \mathbb{N} \wedge A \preceq B \implies B \npreceq \mathbb{N}.$$

In particular, if a subset of a set B is uncountable, then so is B.

L. If A is uncountable and B is countable, then $A - B$ is uncountable.

T. The set $\mathbb{R}$ of real numbers is uncountable.

T. The interval $[0, 1)$ of real numbers is uncountable.

4 Graph Theory

D. ((Simple) Graph) $G = (V, E)$ consists of a finite set V of vertices (Knoten) and a set $E \subseteq \{\{u, v\} \subseteq V | u \neq v\}$ of edges (Kanten).
An edge $\{u, v\}$ is said to *connect* the vertices u and v. Vertices connected by an edge are also called *adjacent* (or neighbors).

D. (Neighborhood) of a vertex v is the set

$$\Gamma(v) := \{u \in V | \{u, v\} \in E\}$$

D. (Directed Graph) $G = (V, E)$ consists of a finite set V of vertices and a set $E \subseteq V \times V$ of (directed) edges.

D. (In- and Out-Degree) The *in-degree* $\deg^-(v)$ of a vertex v is the number of edges entering v, and the *out-degree* $\deg^+(v)$ of v is the number of edges leaving v.

L. (Sum of Degrees) In a directed graph,

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|.$$

In an undirected graph,

$$\sum_{v \in V} \deg(v) = 2|E|.$$

D. (Subgraph) A graph $G = (V, E)$ is a *subgraph* of a graph $H = (V', E')$, sometimes denoted $G \sqsubseteq H$, if $V \subseteq V'$ and $E \subseteq E'$.

D. (Union) of two graphs $G = (V, E)$ and $H = (V', E')$ is the graph $G \cup H := (V \cup V', E \cup E')$.
D. (Complement) $\overline{G}$ of a graph $G = (V, E)$ is the graph $\overline{G} = (V, \overline{E})$ where $\overline{E}$ consists of all possible edges that are not in E.

D. (Bipartite) A graph $G = (V, E)$ is called *bipartite* if V can be split into two disjoint sets V_1 and V_2 of vertices $V = V_1 \cup V_2$, such that no edge connects two vertices in the same subset V_i $(i = 1, 2)$.

D. (Adjacency Matrix) $A_G = [a_{ij}]$ of an undirected graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$ is the binary $n \times n$ matrix where

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \in E \\ 0 & \text{otherwise.} \end{cases}$$

For a directed graph, the condition $\{v_i, v_j\} \in E$ must be replaced by $(v_i, v_j) \in E$.

D. (Graph Isomorphism) Two graphs $G = (V, E)$ and $H = (V', E')$ are *isomorphic*, denoted $G \cong H$ if there exists a bijection $\pi \colon V \leftarrow V'$ such that renaming the vertices of G according to π resulsts in H, i.e., if $\{u, v\} \in E \iff \{\pi(u), \pi(v)\} \in E'$.
For directed graphs the definition is similar, except that $\{u, v\}$ and $\{\pi(u), \pi(v)\}$ must be replaced by (u, v) and $(\pi(u), \pi(v))$, respectively.
Note that $\cong$ is an equivalence relation on the set of graphs.

Trick: To check whether two graphs are isomorph: find the vertices of highest degree and label them, see if a same path is contained, or check if they are both bipartite, or check if the both have triangles or other shapes.

D. (Contained) A graph $G = (V, E)$ is *contained* in a graph $H = (V', E')$, denoted $G \preceq H$, if there exists a subgraph K of H that is isomorphic to G:

$$G \preceq H :\iff \exists K (G \cong K \wedge K \sqsubseteq H).$$

Note that $\preceq$ is a partial order relation on the set of graphs.

D. (Complete Graph) on n vertices, denoted K_n, is a simple graph with n vertices in which any pair of vertices is connected.
D. (Empty Graph) is the complement of the complete graph (with no edges).
D. ((m, n)-Mesh) is a graph $M_{m,n}$ on mn vertices with $V = \{(i, j) | 1 \leq i \leq m,\ 1 \leq j \leq n\}$ and $E = \{\{(i, j), (i', j')\} \mid (i = i' \wedge |j - j'| = 1) \vee (|i - i'| = 1 \wedge j = j')\}$.
Com. the concept of a mesh also extends to higher dimensions.
D. (d-Dimensional Hypercube) Q_d is a graph on $V = \{0, 1\}^d$ with $\{u, v\} \in E$ if and only if u and v differ in exactly one bit.
D. (Complete Bipartite Graph) $K_{m,n}$ is a graph on $m + n$ vertices obtained by taking two vertex subsets B and W (for black and white) of sizes m and n, respectively, and connecting each vertex in B with every vertex in W, i.e., $K_{m,n} = (V, E)$ with $V = B \cup W$, $B \cap W = \emptyset$, $|B| = m$, $|W| = n$, and $E = \{\{u, v\} | u \in B \wedge v \in W\}$.
L. Here some observations about the special graphs

$$P_2 \cong K_{2,1} \qquad C_4 \cong K_{2,2} \cong Q_2 \cong M_{2,2}$$

$$P_m \preceq C_n \text{ when } m < n \qquad P_m \not\preceq C_n \text{ for } m \geq n \qquad C_{2^d} \preceq Q_d$$

4.1 Paths and Cycles

D. (Various Paths and Cycles trough Graphs)
walk [Weg]: any edges or vertices
↳ *tour* [Tour]: if all edges are distinct
  ↳ *circuit* [Schleife]: if start and endpoint are the same
↳ *path* [Pfad]: if all vertices are distinct
  ($\implies$ edges must be distinct $\implies$ every path is a tour)
  A path P_n of length n consists of
  · $n + 1$ vertices: $V = \{v_0, \ldots, v_n\}$
  · n edges: $E = \{\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}\}$
  ↳ *cycle* [Kreis,Zyklus]: if starting and endpoint are identical
    ($\implies$ every cycle is a circuit)
    A cycle C_n of length n consists of
    · n vertices: $V = \{v_1, \ldots, v_n\}$
    · n edges: $E = \{\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}\}$
    ↳ *hamiltonian cycle*: if all vertices are visited

D. (Hamiltonian Graph) a (directed) graph with a *hamiltonian cycle*.

T. In a graph G with adjacency matrix A_G there exists a walk of length ℓ from vertex u to vertex v if and only if the entry at position (u, v) in $(A_G)^\ell$ is not zero. Actually, this entry

corresponds to the number of distinct walks of length ℓ from u to v.

D. (Connected, Components) An undirected graph G is *connected* if any two vertices are connected by a path. The maximal connected subgraphs of a graph G are called *components*.

L. The complete graph K_n is trivially hamiltonian. Hence, adding enough edges will always turn a graph into a Hamiltonian graph.
T. A graph $G = (V, E)$ for which

$$|V| \geq 3 \quad \text{and} \quad \deg(u) + \deg(v) \geq |V|$$

for every non-adjacent pair (u, v) of vertices (i.e., $\{u, v\} \notin E$), is Hamiltonian. In particular

$$\forall v \in V\colon\ \deg(v) \geq \frac{|V|}{2} \implies G \text{ is Hamiltonian.}$$

Proof. Is done by contradiction assuming a maximal non-Hamiltonian graph (adding one edge would lead to a Hamiltonian graph), and the conditions of above. Then one adds the edge, one formalises the hamiltonian cycle. Then by the pigeonhole principle one can formalise that there would exist another hamiltonian cycle which contradicts the maximal non-Hamiltonian assumption, and thus a graph satisfying the conditions must be hamiltonian.
T. The hypercube Q_d is Hamiltonian for $d \geq 2$.
D. Gray Code a hamiltonian cycle in a hypercube.
L. A bipartite graph can have a Hamiltonian cycle only if the two sets of vertices have equal size.
L. From the previous lemma it follows that the mesh $M_{m,n}$ has no hamiltonian cycle if both m and n are odd.

4.2 Trees

D. (Tree) is an undirected connected graph with no cycles.
D. (Forest) is an undirected graph with no cycles, i.e., the union of several trees with disjoint vertex sets .
D. (Leaf) is a vertex with degree 1.

L. A *tree* with $n \geq 2$ vertices has at least 2 leaves.

T. (Tree Properties) For a graph G with n vertices, the following statements are equivalent:

(a) G is a tree.
(b) G has $n - 1$ edges and no cycles.
(c) G has $n - 1$ edges and is connected.

D. (Spanning Tree) of a connected graph G is a subgraph of G which is a tree and contains all vertices of G.

D. (Rooted Tree) is a tree with a distiguished vertex, the *root*. There is a unique path from the root to every vertex v; its length is the *distance* of v from the root. The *height*, or *depth* of the tree is the maximal distance of a leaf from the root. The vertices on the path from the root to v are called *ancestors* of v. The ancestor which is a neighbor of v is called the *parent*, and v is called a *child* of the parent. A rooted tree is a *d-ary tree* if every vertex has at most d children.

4.3 Planar Graphs

Note that graph can always be embedded without crossings in three-dimensional space.

D. (Planarity) A graph is *planar* if it can be drawn in the plane with no edges crossing.

D. (Regions, Degree) A drawing of a planar graph divides the plane into disjoint *regions*, one of which is infinite. The *degree* of a region is the number of edges one encounters in a walk around the region's boundary. (An edge is counted twice if the edge is a bridge.)

T. (Euler's formula) A plane drawing of a connected planar graph $G = (V, E)$ divides the plane into $r := |E| - |V| + 2$ regions.
L. For any connected planar graph $G = (V, E)$, the sum of the degrees of the regions is equal to $2|E|$.
T. Every connected planar graph $G = (V, E)$ with $|V| \geq 3$ satisfies

$$|E| \leq 3|V| - 6.$$

If G is bipartite, then the following stronger inequality holds:

$$|E| \leq 2|V| - 4.$$

C. K_n is planar $\iff n \leq 4$.

C. $K_{3,3}$ is not planar.

Planarity Preserving Operations
We can define three operations on a graph:

1. deletion of edges,
2. deletion of singleton vertices, and
3. merging neighboring vertices, i.e., deleting the edge between them, replacing the two vertices by a single vertex and maintaining all the edges from the two (merged) vertices.

L. If a sequence of these three operations is performed on a graph G and the resulting graph H is non-planar, then also G is non-planar.

D. (Polyhedron) is a solid bounded by a finite number of (plane) polygon faces. The vertices and edges of these polygons are the vertices and edges of the polyhedron. A polyhedron is *convex* if the straight line segment connecting any two points lies entirely within it. A polyhedron is *regular* if for some $m, n \geq 3$ each vertex meets exactly m faces (and hence m edges) and each face is a regular n-gon.

T. There are exactly five regular polyhedra, where (m, n) is either $(3,3), (3,4), (4,3), (3,5)$, or $(5,3)$.

	$\lvert V\rvert$		$\lvert F\rvert$	$\lvert E\rvert$	m = Faces met per Vertex	n = Edges of n-gons (Faces)
Tetraeder	4	-	4	6	3	3
Hexaeder	8	×	6	12	3	4
Oktaeder	6		8	12	4	3
Dodekaeder	20	×	12	30	3	5
Ikosaeder	12		20	30	5	3

Note that $-$ and $\times$ indicate the duality.

5 Number Theory

5.1 Divisors and Division

The integers $\mathbb{Z}$ are a special case of a mathematical strucutre called a *ring*, which will be discussed in Chapter 7. In this chapter we mention in a few places that concepts like divisors, greatest common divisors, ideals, etc. can be defined for any ring, not just for the integers.

D. (Divisibility) For $a, b \in \mathbb{Z}$ with $a \neq 0$ we define

$$a|b :\iff \exists c \in \mathbb{Z}\colon ac = b$$

(a=divisor, factor of b; c=quotient=$\frac{b}{2}$; b = mulitple of a)

Com. Note that every non-zero integer is a divisor of 0. Moreover, 1 and -1 are divisors of every integer.

D. (Euclid, Quotient Remanider)

$$\forall a, d \in \mathbb{Z}\ \ (d \neq 0)\ \ \exists!\, q, r \in \mathbb{Z}\colon\ a = dq + r\ \wedge\ 0 \leq r \leq |d|$$

(a=dividend; d=divis.; q=quot.; r=remain.=$R_d(a)$=$a \mod d$)

D. (A Greatest Common Divisor) $d \neq 0$; a, b not both 0

$$d := gcd(a,b) \overset{\neg(a=b=0)}{\iff} (d|a \wedge d|b)\ \wedge\ (\forall c \in \mathbb{Z} - \{0\}\colon c|a \wedge c|b \Rightarrow c|d)$$

D. (The Greatest Common Divisor) $= |\gcd(a,b)|$

Com. The concept of a greatest common divisior applies not only to $\mathbb{Z}$, but more general structures (e.g. polynomial rings). If d and d' are both greatest common divisors of a and b, then $d \mid d'$ and $d' \mid d$. For the integers $\mathbb{Z}$, this means that $d' = \pm d$, i.e., there are two greatest common divisors. (But for more general structures there can be more than two greatest common divisors.)

D. (Relatively Prime)

$$a \text{ and } b \text{ are relatively prime} \iff gcd(a,b) = 1$$

D. (Ideal) generated by a (and b)

$$(a,b) := \{ua + vb \mid u, v \in \mathbb{Z}\} \qquad (a) := \{ua \mid u \in \mathbb{Z}\}$$

L. Every ideal in $\mathbb{Z}$ can be generated by a single integer.

$$\forall a, b \in \mathbb{Z} \text{ (not both 0)}\colon (a,b) = (d) \Longrightarrow d = \pm \gcd(a,b)$$

C.

$$\forall a, b \in \mathbb{Z} \text{ (not both 0) } \exists u, v \in \mathbb{Z}\colon\ \gcd(a,b) = ua + vb$$

T. (Euclid's Extended GCD Algorithm) efficiently computes for given nonnegative integers a and b with $a \geq b$ (not both 0), the integers $d = \gcd(a,b)$, as well as u and v satisfying $ua + vb = \gcd(a,b)$.

Mostly it's used to compute multiplicative inverse mudulo an integer m, ie.e to compute for a given a the integer b such that $R_m(ab) = 1$.

Application

How to compute $d := \gcd(a,b)$ with $a \geq b$ as well as u and v satisfying $ua + vb = \gcd(a,b) =: d$.

	q	s_1	s_2	u_1	u_2	v_1	v_2
after init	$\lfloor \frac{a}{b} \rfloor$	a	b	1	0	0	1
after n-th loop	$\lfloor \frac{s_1}{s_2} \rfloor$	s_2	$s_1 - qs_2 =: R_{s_2}(s_1)$	u_2	$u_1 - qu_2$	v_2	$v_1 - qv_2$
term. if		$\underline{s_2}$	$\boxed{\leq 0}$	$\underline{\underline{u_2}}$		$\underline{\underline{v_2}}$	

Then the result is: $d := s_1$, $u = u_1$, $v = v_1$;

(You can just compute the values of the next line with the values from the previous line (Except q which is computed from the values of the current line, and is done last). Stop calculation the next line if the termination condition was met in the previous line.)

Example: $\gcd(789, 22)$

	q	s_1		s_2	u_1		u_2	v_1		v_2
after init	35	789	↙	22	1	↙	0	0	↙	1
after 1st loop	1	22	↙	19	0	↙	1	1	↙	-35
after 2nd loop	6	19	↙	3	1	↙	-1	-35	↙	36
after 3rd loop	3	3	↙	1	-1	↙	7	36	↙	-251
after 4th loop		$\underline{1}$		$\boxed{0}$	$\underline{7}$			$\underline{\underline{-251}}$		

5.2 Factorisation into Primes

D. (Prime) A positive integer $p > 1$ is called *prime* if the only positive divisors of p are 1 and p.

D. (Composite) An integer greater than 1 that is not a prime is called *composite*.

Com. This notion of having only trivial divisors extends to other rings, for example $\mathbb{R}[x]$. In such a general context, the property is called *irreducible* rather than *prime*.

The term *prime* is in general used for the property that if p divides a product of elements, then it divides at least one of them.

For the integers, these two concepts are equivalent. The next lemma states one direction of this equivalence.

L. If p is a prime which divides the product $x_1 x_2 \cdots x_n$ of some integers $x_1, \ldots, x_n$, then p divides one of them, i.e. $p \mid x_i$ for some $i \in \{1, \ldots, n\}$.

$$p \text{ is prime } \wedge\, p|x_1 x_2 \cdots x_n \Longrightarrow \exists i \in \{1, \ldots, n\}\colon p|x_i$$

T. (Fundamental Theorem of Arithmetic) Every positive integer can be written uniquely (up to the order in which factors are listed) as the product of primes.

T. $\sqrt{n}$ is irrational unless n is a square ($n = c^2$, for $c \in \mathbb{Z}$).

D. (Least Common Multiple) l of two positive integers a and b, denoted $l = \text{lcm}(a,b)$, is the common multiple of a and b which divides every common multiple of a and b, i.e., $a \mid l, b \mid l, l > 0$, and

$$(l = \text{lcm}(a,b)) \qquad a \mid l' \wedge b \mid l' \Longrightarrow l \mid l'.$$

Building GCD and LCM of Prime Factorisation of a and b

$$a = \prod_i p_i^{e_i} \qquad b = \prod_i p_i^{f_i}$$

$$\gcd(a,b) = \prod_i p_i^{\min(e_i, f_i)} \qquad \text{lcm}(a,b) = \prod_i p_i^{\max(e_i, f_i)}.$$

It is easy to see that:

$$\gcd(a,b) \cdot \text{lcm}(a,b) = ab.$$

One can easily derive the following rules from the prime factorisation.

$$even + even = even \qquad even \cdot even = even$$
$$even + odd = odd \qquad even \cdot odd = even$$
$$odd + odd = even \qquad odd \cdot odd = odd$$

5.3 Congruences and Modular Arithmetic

D. (Congruence) For $a, b, m \in \mathbb{Z}$, $m \geq 1$

$$a \equiv b \pmod m :\iff a \equiv_m b :\iff m|(a-b)$$

"a is congruent to b modulo m"

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

Com. There are m equivalence classes, namely $[0], [1], \ldots, [m-1]$. Each equivalence class $[a]$ has a natural representative $R_m(a) \in [a]$ in the set $\mathbb{Z}_m := 0, 1, \ldots, m-1$ of remainders modulo m.

L. (Compatibility with Arithmetic Operations)

$$(a \equiv_m b)\ \wedge\ (c \equiv_m d)\ \Longrightarrow\ (a + c \equiv_m b + d)\ \wedge\ (ac \equiv_m bd)$$

C. Let $f(x_1, \ldots, x_k)$ be a multi-variate polynomial in k variables with integer coefficients, and let $m \geq 1$. If $a_1 \equiv_m b_i$ for $1 \leq i \leq k$, then $f(a_1, \ldots, a_k) \equiv_m f(b_1, \ldots, b_k)$.

Good to know: If an equality holds over the integers, then it must also hold over any modulus m. In other words,

$$a = b \implies a \equiv_m b$$

The implication can be turned around and can be used to prove the inequality of two numbers a and b:

$$a \not\equiv_m b \implies a \neq b$$

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

(i) $a \equiv_m R_m(a)$.

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

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

(i) $R_m(a + b) = R_m(R_m(a) + R_m(b))$.

(ii) $R_m(ab) = R_m(R_m(a) \cdot R_m(b))$.

L. $(\exists x \in \mathbb{Z}_m \colon\ ax \equiv_m 1) \iff \gcd(a, m) = 1$
The solution x is unique (use Euclid's GCD).
D. (Multiplicative Inverse) If $\gcd(a, m) = 1$, the unique solution $x \in \mathbb{Z}_m$ to the congruence equation $ax \equiv_m 1$ is called the *multiplicative inverse of a modulo m*. One also uses the notation $x \equiv_m a^{-1}$ or $x \equiv_m 1/a$.

T. (Chinese Remainder Theorem, CRT) Let $m_1, m_2, \ldots, m_r$ be pairwise relatively prime integers and let $M = \prod_{i=1}^{r} m_i$. For every list $a_1, \ldots, a_r$ with $0 \leq a_i < m_i$ for $1 \leq i \leq r$, the system of congruence equations

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

for x has a unique solution x satisfying $0 \leq x < M$.

Application of CRT Compute all $M_i = M/m_i$. Now since $\gcd(M_i, m_i) = 1$ there exists an N_i satisfying

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

Determine all N_i by trying (e.g. increasing i from 0 to m_i). Then determine the solution x by computing

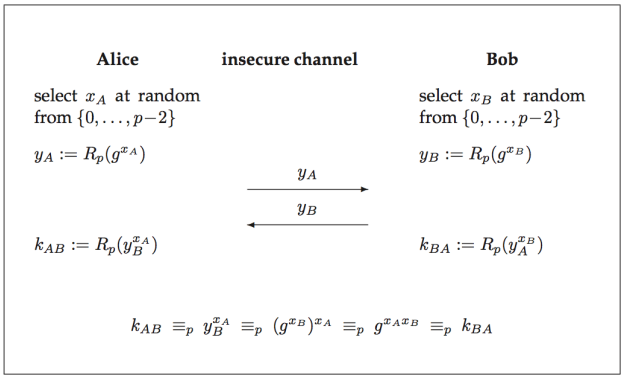
$$x = R_M\left(\sum_{i=1}^{r} a_i M_i N_i\right).$$

5.4 Diffie-Hellman Key-Agreement Protocol

Color Analogy: It's hard to mix two colors together. However, it's almost impossible to find out the two colors that were mixed to obtain a color, since there are many ways to mix a color.

Lock: A one-way function that is easy to compute in one direction and hard to compute in the other direction (e.g. discrete logarithm, $a^x \equiv_k b$).

Steps: Assuming you're A, and you want to establish a shared secret key with B without meeting him except through the network:



1.)	Agree publicly on a starting color.	Agree publicly on a prime modulus p and a generator g. You may also choose any other finite cyclic group.
2.)	Randomly select a private color (and keep it).	Select your secret x_A from the carrier set.
3.)	Mix the private and public color	Raise the generator g to your secret x_A to get y_A, $y_A = g^{x_A}$.
4.1)	Send the mixture to the other guy B.	Send the computed y_A to the other guy B.
4.1)	Receive the color of the other guy B.	Receive the computed y_B of the other guy B.
5.)	Add your private color to the received mixture of B. This will build the shared secret color.	Take the received y_B and raise it to your secret x_A. This will yield you the shared private key k_{AB}.

6 Logic

6.1 Elementary Concepts in Logic

D. (Proof System) is a quadruple $\Pi = (\mathcal{S}, \mathcal{P}, \tau, \phi)$, where

$\mathcal{S}$ is the set of (syntactic representations of) mathematical statements. Every statement $s \in \mathcal{S}$ is either true or false.

$\mathcal{P}$ is the set of proofs. Proofs $p \in \mathcal{P}$ are also syntactic objects, for example strings over some alphabet.

τ The function $\tau \colon \mathcal{S} \to \{0, 1\}$ assigning to each $s \in \mathcal{S}$ its truth value $\tau(s)$ can be called the *truth function*. This function defines the meaning, called the *semantics*, of the objects in $\mathcal{S}$.

ϕ A proof p for a statement s is relative to a *verification function* $\phi \colon \mathcal{S} \times \mathcal{P} \to \{0, 1\}$, where $\phi(s, p) = 1$ means that p is a valid proof for the statement s in the proof system. Hence, the function ϕ defines what is a proof $p \in \mathcal{P}$ for a statement $s \in \mathcal{S}$.

D. (Soundness) A proof system Π is *sound* (korrekt) if no false statement has a proof, i.e., if for all $s \in \mathcal{S}$ for which there exists $p \in \mathcal{P}$ with $\phi(s, p) = 1$, we have $\tau(s) = 1$. This means

$$\forall s \in \mathcal{S}\ \forall p \in \mathcal{P} \colon \quad \phi(s, p) = 1 \implies \tau(s) = 1$$

D. (Completeness) A proof system Π is *complete* if every true statement has a proof, i.e., if for all $s \in \mathcal{S}$ with $\tau(s) = 1$, there exists $p \in \mathcal{P}$, with $\phi(s, p) = 1$.

$$\forall s \in \mathcal{S} \colon \quad \tau(s) = 1 \implies \exists p \in \mathcal{P}\ \phi(s, p) = 1$$

In addition, one requires that the function ϕ is *efficiently computable* (for some notion of efficiency) and that every true statement has a reasonably short proof.

D. (Syntax) of a logic defines an alphabet (of allowed symbols) and specifies which strings (over the alphabet) are syntactically correct formulas.

D. (Structure) A formulla generally contains certain variable parts which are not determined (by the formula) and can take on values in certain domains. A particular choice of these variable parts is caled a *structure*.
D. (Suitable Structure) A structure is *suitable* for a formula F if all variable elements of F are defined (i.e., fixed), i.e., if it makes the formula true or false. (It may also define *more* variables than the ones appearing in F).
Important: Note that for a structure to be suitable it also needs to define the *free* variables (i.e., assign them some value!).

D. (Semantics) The *sematics* of a logic is a function σ assigning to each formula F and each structure $\mathcal{A}$ suitable for F a truth value $\sigma(F, \mathcal{A})$ in $0, 1$.

D. (Model) A formula F (or set M of formulas) is called *satisfiable* if there exists a model F for M, and *unsatisfiable* otherwise. The symbol $\bot$ is used for an unsatisfiable formula.

D. (Tautology) A formula F is called a *tautology* or *valid* if it is true for every suitable structure. The symbol $\top$ is used for a tautology.

D. (Logical Consequence) A formula G is a *logical consequence* of a formula F (or a set M of formulas) denoted $F \models G$ (or $M \models G$), if every structure suitable for both F (or M) and G, which is a model for F (for M), is also a model for G.
D. (Equivalence) Two formulas F and G are *equivalent*, denoted $F \equiv G$ (or also $F \iff G$), if every structure suitable for both F and G yields the same truth value for F and G, i.e., if each is logical consequence of the other: $F \equiv G :\iff F \models G$ and $G \models F$.

6.2 Logical Calculi

D. (Derivation Rule) is a rule for deriving a formula from a set of formulas (called the precondition). We write $\{F_1, \ldots, F_k\} \vdash_R G$ if G can be derived from the set $\{F_1, \ldots, F_k\}$ by rule R.
D. (Calculus) A (logical) *calculus* K is a finite set of derivation rules: $K = \{R_1, \ldots, R_m\}$.
D. (Derivation) A *derivation of a formula* G from a set M of formulas in a calculus K is a finite sequence (of some length n) of applications of rules in K, leading to G.
More precisely, we have $M_0 = M$, $M_i := M_{i-1} \cup \{G_i\}$ for some $R_i \in K$, and where $G_n = G$. We write $M \vdash_K G$ if there is a derivation of G from M in the calculus K.
D. (Correctness) A derivation rule R is *correct* if for every set M of formulas and every formula F

$$M \vdash_R F \Longrightarrow M \models F$$

In other words: A rule is correct, if it's derivated formula is always true when it's preconditions are met. As with the implication we need this "$\leq$" relation $0 \leq 1$ or $0 \leq 0$ for all rows.

D. (Soundness) A calculus K is *sound* or *correct* if for every set M of formulas and every formula F, if F can be derived from M then F is also a logical consequence of M:

$$M \vdash_K F \Longrightarrow M \models F$$

D. (Completeness) A calculus K is *complete* if for every M and F, if F is a logical consequence of M, then F can also be derived from M:

$$M \models F \Longrightarrow M \vdash_K F$$

One writes $\vdash_K F$ if F can be derived from the empty set of formulas.
L. If $F \vdash_K G$ for a sound calculus, then $\models (F \to G)$.

Example: A calculus that is not sound If a calculus has one rule, which is not correct, like $\{A \vee B\} \vdash A \wedge B$ then the calculus is not sound (or not correct).
Example: A calculus that is complete, but not sound The calculus $K := \{R\}$ with only one rule $\emptyset \vdash_R F$. In this calculus, we can derive any formula from the empty set, however it is not correct. For example deriving $\emptyset \vdash A \wedge B$ is not correct.
Example: A calculus that is sound, but not complete:
Let K consist of the following two rules.

$$\{A \wedge B\} \vdash_{R_1} A \qquad \{A, A \to B\} \vdash_{R_2} B$$

As we can see the Rules R_1 and R_2 are correct. However, the calculus does not allow to derive from $B \wedge A$ the statement $A \wedge B$, even though $A \wedge B \models B \wedge A$ ($B \wedge A$ is a logical consequence of $A \wedge B$). Hence, there exists a set of formulas, where te calculus does not allow to derive all its logical consequences from it. Therefore the calculus is not complete.

Anwendung der Regeln Beim Anwenden von Regeln muss man immer angeben, welche Regel man anwendet, welche Formeln man als welches Argument benutzt, und dann nummeriert man die entstehende Formel (so kann man sie später wieder als Argument benutzen).

6.3 Propositional Logic

D. (Syntax,Atomic Formula, Formula) An *atomic formula* is of the form A_i with $i \in \mathbb{N}$. A *formula* is defined inductively: An atomic formula is a formula, and if F and G are formulas, then also $\neg F$, $\neg G$, $(F \wedge G)$, and $(F \vee G)$ are formulas.

D. (Semantics) For a set M of atomic formulas, a *truth assignment* is a function $\mathcal{A} \colon M \to \{0, 1\}$. Let $\widehat{M}$ be the set of formulas built from atomic formulas in M. We extend the domain of $\mathcal{A}$ to $\widehat{M}$ as follows:

$$\mathcal{A}((F \wedge G)) = 1 \text{ if and only if } \mathcal{A}(F) = 1 \text{ and } \mathcal{A}(G) = 1$$
$$\mathcal{A}((F \vee G)) = 1 \text{ if and only if } \mathcal{A}(F) = 1 \text{ or } \mathcal{A}(G) = 1$$
$$\mathcal{A}(\neg F) = 1 \text{ if and only if } \mathcal{A}(F) = 0$$

Example: Extend the propositional logic with the symbol $\oplus$ for the exlusive or ($A \oplus B$ is exactly then true, when either A or B is true, but not both).
Syntax: For all formulas F and G, $(F \oplus G)$ is also a formula.
Semantics: $\mathcal{A}((F \oplus G)) = 1$ if and only if $\mathcal{A}(F) = 1$ or $\mathcal{A}(G) = 1$, but not both.

Sprachliche Bedeutung A kommt nur, wenn auch B kommt. Bedeutet dasselbe wie A ist nur dann wahr, wenn auch B wahr ist. Das heisst nicht genau dann wenn. Das heisst mehr "$A \leq B$". Das heisst $A \to B$.

L. 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)

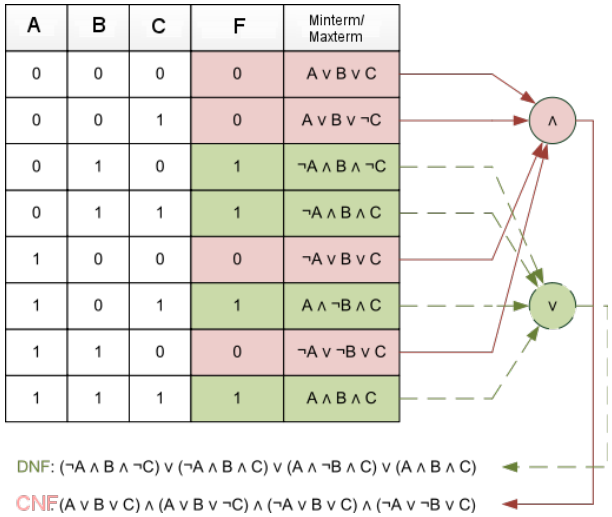
3) $(F \wedge G) \wedge H \equiv F \wedge (G \wedge H)$ and
$(F \vee G) \vee H \equiv F \vee (G \vee H)$ (associativity)
4) $F \wedge (F \vee G) \equiv F$ and $F \vee (F \wedge G) \equiv F$ (absorption)
5) $F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H)$ and
$F \vee (G \wedge H) \equiv (F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

D. (Literal) is an atomic formula or the negation of an antomic formula

D. (Conjunctive Normal Form (CNF)) A formula F is in CNF if it is a conjunction of disjunctions of literals, i.e., if it is of the form $\bigwedge_i \bigvee_j (\neg) x_{ij}$.
The CNF of a Formula can be obtained from its truth table as follows: For every row where $F = 0$, build the disjunction of the inverse of every literal (i.e., take $\neg A_i$ if $A_i = 1$ and take A_i if $A_i = 0$) and then conjunct the disjunctions.

D. (Disjunctive Normal Form (DNF)) A formula F is in DNF if it is a disjunction of conjunction of literals, i.e., if it is of the form $\bigvee_i \bigwedge_j x_{ij}$.
From a truth table it can be obtained as follows: For every row where $F = 1$, build the conunction of every literal (i.e., take A_i if $A_i = 1$, or $\neg A_i$ if $A_i = 0$) and then disjunct the conunctions.

T. Every formula is equivalent to a formula in CNF and also to a formula in DNF.



D. (Clause) is a set of literals.

D. (Set of Clauses associated to a Formula)

$$F = (L_{11} \vee \cdots \vee L_{1m_1}) \wedge \cdots \wedge (L_{n1} \vee \cdots \vee L_{nm_n})$$

in CNF, denoted as $\mathcal{K}(F)$, is the set

$$\mathcal{K}(F) := \{\{L_{11}, \ldots, L_{1m_1}\}, \ldots, \{L_{n1}, \ldots, L_{nm_n}\}\}$$

D. (Set of Clauses Associated with a Set of Formulas) $M = \{F_1, \ldots, F_k\}$ qis the union of their clause sets: $\mathcal{K}(M) := \cup_{i=1}^{k} \mathcal{K}(F_i)$

D. (Resolvent) A clause K is a *resolvent* of claues K_1 and K_2 if there is a literal L such that $L \in K_1$, $\neg L \in K_2$, and

$$K = (K_1 - \{L\}) \cup (K_2 - \{\neg L\})$$

Given a set $\mathcal{K}$ of clauses, a resolution step takes two clauses $K_1 \in \mathcal{K}$ and $K_2 \in \mathcal{K}$, computes a resolvent K, and adds K to $\mathcal{K}$. One can also write the resolution rule as

$$\{K_1, K_2\} \vdash_{\text{res}} K,$$

where the previous equation must be satisfied. The resolution calculus, denoted Res, consist of a single rule: $Res = \{res\}$.

L. The resolution calculus is sound, i.e., if $\mathcal{K} \vdash_{Res} K$ then $\mathcal{K} \models K$.

T. A set M of formulas is unsatisfiable iff $\mathcal{K}(M) \vdash_{Res} \emptyset$.

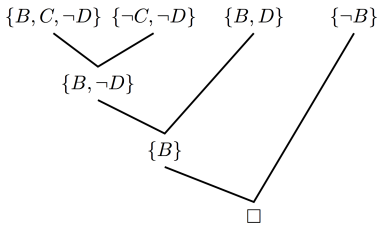
Example: Show that

$$G = (\neg B \wedge \neg C \wedge D) \vee (\neg B \wedge \neg D) \vee (C \wedge D) \vee B$$

is a tautology.
This is true when $\neg G$ is unsatisfiable. Since G is in DNF we can just turn it into DNF with de Morgan's rules:

$$\neg G = (B \vee C \vee \neg D) \wedge (B \vee D) \wedge (\neg C \vee \neg D) \wedge \neg B$$

Now, since we have the CNF we can just apply the resolution calculus on the set of literals to derive the empty set, in order to show that $\neg G$ is unsatisfiable. Hence, G is a tautology.



6.4 Predicate Logic

D. (Syntax)

- A *variable* is of the form x_i with $i \in \mathbb{N}$.
- A *function symbol* is of the form $f_i^{(k)}$ with $i, k \in \mathbb{N}$, where k denotes the number of arguments of the function. Function symbolds for $k = 0$ are called *constants*.
- A *predicate symbol* is of the form $P_i^{(k)}$ with $i, k \in \mathbb{N}$, where k denotes the number of arguments of the predicate.
- A *term* is defined inductively:
  - A variable is a term, and if $t_1, \ldots, t_k$ are terms, then $P_i^{(k)}(t_1, \ldots, t_k)$ is a formula, called an *atomic* formula.
  - If F and G are formulas, then also $\neg F$, $(F \wedge G)$, and $(F \vee G)$ are formulas.
  - $\forall x_i\ F$ and $\exists x_i\ F$ are also formulas.

D. (Bounded and Free Variables) Every occurrence of a variable in a formula is either *bound* or *free*. If a variable x occurs in a (sub-)formula of the form $\forall x\ G$ or $\exists x\ G$ then it is bound, otherwise it is free.

D. (Closed Formula) A formula is *closed* if it contains no free variables.

D. (Substitution of Free Variables) For a formula F a variable x and a term t, $F[x/t]$ denotes the formula obtained from F by substituting every free occurrence of x by t.

D. (Structure) A *structure* is a tuple $\mathcal{A} = (U, \phi, \psi, \xi)$ where

- U is a non-empty set, the so-called *universe*,
- ϕ is a function assigning to each funcition symbol (in a certain subset of all function symbols) a function, where for a k-ary function symbol f, $\phi(f)$ is a function $U^k \to 0, 1$, and where

- ξ is a function assigning to each variable symbol (in a certain subset of all variable symbols) a value in U.

D. (Semantics) For a structure $\mathcal{A} = (U, \phi, \psi, \xi)$, we define the value (in U) of terms and the truth value of formulas under that structure.

- The value $\mathcal{A}(t)$ of a term t is defined recursively as follows:
  - If t is a variable, then $\mathcal{A}(t) = \xi(t)$.
  - If t is of the form $f(t_1, \ldots, t_k)$ for some terms $t_1, \ldots, t_k$ and a k-ary function symbol f, then
$$\mathcal{A}(t) = \psi(f)(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_k)).$$
- The truth value of a formula F is defined recursively as follows:
  -

L. Any equivalence that holds in propositional logic also holds in predicate logic. Moreover, for any formulas F, G and H, where H does not contain the variable x, we have

1) $\neg(\forall x\ F) \equiv \exists x\ \neg F$
2) $\neg(\exists x\ F) \equiv \forall x\ \neg F$
3) $(\forall x\ F) \wedge (\forall x\ G) \equiv \forall x\ (F \wedge G)$
4) $(\exists x\ F) \vee (\exists x\ G) \equiv \exists x\ (F \vee G)$
5) $\forall x\ \forall y\ F \equiv \forall y\ \forall x\ F$
6) $\exists x\ \exists y\ F \equiv \exists y\ \exists x\ F$
7) $(\forall x\ F) \wedge H \equiv \forall x\ (F \wedge H)$
8) $(\forall x\ F) \vee H \equiv \forall x\ (F \vee H)$
9) $(\exists x\ F) \wedge H \equiv \exists x\ (F \wedge H)$
10) $(\exists x\ F) \vee H \equiv \exists x\ (F \vee H)$

L. If one replaces a subformula G of a formula F by an equivalent (to G) formula H, then the resulting formula is equivalent to F.

L. For a formula G in which x occurs only free and in which y does not occur
$$\forall x\ G \equiv \forall y\ G[x/y]$$
$$\exists x\ G \equiv \exists y\ G[x/y]$$

D. (Rectified Form) By appropriately renaming quantified variables one can transform any formula into an equivalent formula in which no variable appears both as a bound and free variable and such that all variables appearing after the quantifiers are distinct. Such a formula is said to be in *rectified* form.

D. (Prenex Form) A formula of the form
$$Q_1 x_1 Q_2 x_2 \cdots Q_n x_n\ G$$
where Q_i are arbitrary quantifiers ($\forall$ or $\exists$) and G is a formula free of quantifiers, is said to be in *prenex form*.

Example: Put $F = \neg\forall x\ \exists y\ P(x,y) \wedge \forall x\ Q(y,x)$ into prenex form.
Rename y in the right part, since it's a free variable there to avoid name collisions.
$$F = \neg\forall x\ \exists y\ P(x,y) \wedge \forall x\ Q(z,x)$$
Push the $\neg$ inside on the right side in order to have just quantifiers in the front.
$$F = \exists x\ \forall y\ \neg P(x,y) \wedge \forall x\ Q(z,x)$$
Rename x on the right side in order to avoid name collisions.
$$F = \exists x\ \forall y\ \neg P(x,y) \wedge \forall a\ Q(z,a)$$
Move the quantifiers to the front.
$$F = \exists x\ \forall y\ \forall a\ \neg P(x,y) \wedge Q(z,a)$$
F is now in prenex form. z is the only free varible x, y and a are bound variables. It doesn't matter whethere we move $\forall a$ to the front or to the end of the sequence of quantifiers.

T. The following formula is a tautology:
$$F := \neg\exists x\ \forall y\ (P(y,x) \leftrightarrow \neg P(y,y))$$
This means every suitable structure is a model for F.
Proof. The formula can be manipulated to say
$$\forall x\ \exists y (P(y,x) \leftrightarrow P(y,y))$$
which can be always satisfied by choosing $y := x$.
C. There exists no set that contains all sets S that do not contain themselves, i.e., $\{S | S \notin S\}$ is not a set.
C. The set $\{0,1\}^\infty$ is not countable.
C. There are functions $\mathbb{N} \to \{0,1\}$ that are not computed by any program.
Example: Here we give an example for $\neg\exists x\ \forall y\ (P(y,x) \leftrightarrow \neg P(y,y))$
Let

- $U^{\mathcal{A}}$ be the set of all men of Zuerich
- $P^{\mathcal{A}}(x,y) = 1$, if y shaves x

In this case there does not exist a man x, such that for all men y (including x), x shaves y if and only if y does not shave itself. In the case of x this would mean "x shaves x if and only if x does not shave itself x" which cannot be, therefore no such x exists.