mathematical reasoning, proofs, logic

Mathematical Statements

The Concept of a Mathematical Statement

Definition 2.1: A matheamtical statement (proposition) is a statement that is ture or false in an exact, indisputable sense, according to the laws of mathematics.

Composition of Mathematical Statements

a 'and' b: both, a, b, must be true for the composition to be true

$S \Rightarrow T$ (implication): If S is true, then T is true.

The Concept of a Proof

The purpose of a proo is to demonstrate (or prove) a mathematical statement S.

Examples of Proofs

Claim: n is not prime $\Rightarrow 2^n - 1$ is not prime.

Proof. $n = ab, a > 1, a < n$. $2^{ab} - 1 = (2^a - 1) \sum_{i=0}^{b-1} 2^{ia}$

Examples of False Proofs

Not relevant for the exam, I guess.

Two Meanings of $\Longrightarrow$

(a) composed statements $S \Rightarrow T$. (b) derivation step in a proof. To avoid confusion, we use $\Longrightarrow$ for (b).

A standard proof pattern is a sequence of implications, each step denoted with $\Longrightarrow$. The justification must be clear - stated in accompanying text/line remark (or implicitly).

Proofs Using Several Implications

To prove $S \Rightarrow T$, one might must do: $S \Longrightarrow S_1, S \Longrightarrow S_2, S_1 \Longrightarrow S_3, S_1 \Longrightarrow S_4, S_2 \Longrightarrow S_5, S_3$ and $S_5 \Longrightarrow S_6, S_1$ and $S_4 \Longrightarrow S_7, S_6$ and $S_7 \Longrightarrow T$.

An Informal Understanding of the Proof Concept

Definition 2.2 (informal): A proof of a statement S is a sequence of simple, easily verifiable, consecutive steps. The proof starts from a set of axioms (things postulated to be true) and known (previously proved) facts. Each step corresponds to the application of a derivation rule to a few already proven statements, resulting in a newly proven statement, until the final step results in S.

Informal vs. Formal Proofs

Most proofs are quite informal. Benefits of formal proofs: Prevention of errors, Proof complexity and automatic verification, Precision and deeper understanding. The border between informal/formal proofs is fluent and varies accross scientific fields.

The Role of Logic

Not relevant here.

Proofs in this Course

Proof sketch/idea: non-obvious ideas are described, but not spelled out in detail with explicit references to all definitions etc.

Complete proof: use of every definition etc. explicit. Every step justified by stating the rule or definition applied.

Formal proof: Phrased in a given proof calculus.

A First Introduction to Propositional Logic

Not relevant here, later in great detail.

A First Introduction to Predicate Logic

Not relevant here, later in great detail.

Logical Formulas vs. Mathematical Statements

Not relevant here, later in great detail.

proof patterns

Composition of Implications

Definition 2.12: The proof step of composing implications is as follows: If $S \Rightarrow T$ and $T \Rightarrow U$ are both true, then $S \Rightarrow U$ is true.

Lemma 2.5: $(A \to B) \land (B \to C) \models A \to C$

Direct Proof of an Implication

Definition 2.13: Direct proof of $S \Rightarrow T$: assuming S, proving T under that assumption.

Indirect Proof of an Implication

Definition 2.14: Indirect proof of $S \Rightarrow T$: assuming T is false, proving S is false under that assumption.

Lemma 2.6: $\neg B \to \neg A \models A \to B$

Modus Ponens

Definition 2.15: A proof of statement S by modus ponens:

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

Lemma 2.7: $A \land (A \to B) \models B$

Case Distinction

Definition 2.16: A proof of statement S ba case distinction:

1. Fina finite list $R_1, \ldots, R_k$ of mathematical statements (cases)
2. Prove that one of the R_i is always true (one case occurs)
3. Prove $R_i \Rightarrow S$ for $i = 1, \ldots, k$

Lemma 2.8: $(A_1 \lor \ldots \lor A_k) \land (A_1 \to B) \land \ldots \land (A_k \to B) \models B$

Proof by Contradiction

Definition 2.17: A proof by contradiction of statement S:

1. Find a suitable mathematical statement T.
2. Prove that T is false.
3. Assume that S is false and prove (from this assumption) that T is true (a contradiction.

Lemma 2.9: $(\neg A \to \neg B) \land \neg B \models A$

Existence Proofs

Definition 2.18: Consider a set $\mathcal{X}$ of parameters and for each $x \in \mathcal{X}$ a statement denoted S_x. An existence proof is a proof of the statement that S_x is true for at least one $x \in \mathcal{X}$. An existence proof is constructive if it exhibits an a for which S_a is true, and otherwise it is non-constructive.

Existence Proofs vis the Pingeonhole Principle

Theorem 2.10: 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.

Proofs by Counterexample

Definition 2.19: Consider a set $\mathcal{X}$ of parameters and for each $x \in \mathcal{X}$ a statement denoted S_x. A proof by counterexample is a proof of the statement that S_x is not true for all $x \in \mathcal{X}$, by exhibiting an a (called counterexample) such that S_a is false.

Proofs by Induction

Definition:

1. *Base case:* Prove $P(0)$.
2. *Induction step:* Prove that for any arbitrary n we have $P(n) \Rightarrow P(n+1)$

Theorem 2.11: universe $\mathbb{N}$, arbitrary unary predicate P: $P(0) \land \forall n (P(n) \to P(n+1)) \Rightarrow \forall n P(n)$.

sets, relations, functions

introduction

Definition 3.1 (informal): The number of elements of a finite set A is called its cardinality and is denoted $|A|$.

Russell's Paradox

This shows flaws in Cantor's early definition of sets/set theory. Set theory was then based on more rigorous grounds. Zermelo-Fraenkel (ZF) set theory most wiedely considered set of axioms.

$R = \{A | A \notin A\}$ - set of sets, which are not elements of themselves. Zermelo's aximoatization: Fo rany set B and predicate P: $\{x \in B | P(x)\}$ is a set, P: $\{x | P(x)\}$ is not a set.

sets and operations on sets

The Set Concept

Universe of possible sets. Universe of objects (may be elements of sets). Both universes may be the same.

Binary predicate E: $E(x,y) = 1 \overset{\text{def}}{\Longleftrightarrow} x$ is an element of y - $x \in y$.

Set Equality and Constructing Sets From Sets

Definition 3.2 - axiom of extensionality: $A = B \overset{\text{def}}{\Longleftrightarrow} \forall x (x \in A \leftrightarrow x \in B)$

a is a set. Then, the set $\{a\}$ exists.

For finite liste of sets $a, b, c, \ldots$ Then, the set $\{a, b, c, \ldots\}$ exists.

Lemma 3.1: For any (sets) a and b, $\{a\} = \{b\} \Rightarrow a = b$.

If cardinality > 1, this does not hold. But we may considere ordered lists of objects, then this still holds. An (ordered) list of k objects $a_1, \ldots, a_k$ is denoted $(a_1, \ldots, a_k)$. Two lists of same length are equal if they agree in every component.

Subsets

Definition 3.3: A set A is a subset of the set B, denoted $A \subseteq B$, if every element of A is also an element of B. $A \subseteq B \overset{\text{def}}{\Longleftrightarrow} \forall x (x \in A \to x \in B)$.

Lemma 3.2: $A = B \Leftrightarrow (A \subseteq B) \land (B \subseteq A)$

Lemma 3.3: For any sets A, B, C: $A \subseteq B \land B \subseteq C \Rightarrow A \subseteq C$.

Union and Intersection

Definition 3.4: The union of two sets A and B is defined as $A \cup B \overset{der}{=} \{x | x \in A \lor x \in B\}$. And their intersection is defined as $A \cap B \overset{def}{=} \{x | x \in A \land x \in B\}$.

$\mathcal{A}$ non-empty set of sets. $\bigcup \mathcal{A} \overset{def}{=} \{x | x \in A$ for some $A \in \mathcal{A}\}$. Analogous for $\cap$.

Definition 3.5: The difference of sets B and A, denoted $B \backslash A$ is the set of elements of B without those that are elements of A: $B \backslash A \overset{def}{=} \{x \in B | x \notin A\}$.

Theorem 3.4:

- $A \cap A = A$ and $A \cup A = A$ (idempotence)
- $A \cap B = B \cap A$ and $A \cup B = B \cup A$ (commutativity)
- $A \cap (B \cap C) = (A \cap B) \cap C$ and $A \cup (B \cup C) = (A \cup B) \cup C$ (associativity)
- $A \cap (A \cup B) = A$ and $A \cup (A \cap B) = A$ (absorption)
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ (distributivity)
- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ (distributivity)
- $A \subseteq B \Leftrightarrow A \cap B = A \Leftrightarrow A \cup B = B$ (consistency)

The Empty Set

Definition 3.6: Set A is called empty if it contains not elements. $\forall x \neg (x \in A)$.

Lemma 3.5: There is only one empty set (which is often denoted as $\varnothing$ or $\{\}$).

Lemma 3.6: The empty set is a subset of every set, i.e., $\forall A (\varnothing \subseteq A)$.

Constructing Sets from the Empty Set

Note that $\{\varnothing\} \neq \varnothing$. We may construct various sets from $\varnothing$: $\varnothing, \{\varnothing\}, \{\{\varnothing\}\}, \{\{\{\varnothing\}\}\}$.

A Construction of the Natural Numbers

$\mathbf{0} \overset{\text{def}}{=} \varnothing, \mathbf{1} \overset{\text{def}}{=} \{\varnothing\}, \mathbf{2} \overset{\text{def}}{=} \{\{\varnothing\}\}, \ldots$ The successor of set $\mathbf{n}$ $(s(\mathbf{n}))$ is defined as $s(\mathbf{n}) \overset{\text{def}}{=} \mathbf{n} \cup \{\mathbf{n}\}$. We define addition as $\mathbf{m} + \mathbf{0} \overset{\text{def}}{=} \mathbf{m}$ and $\mathbf{m} + s(\mathbf{n}) \overset{\text{def}}{=} s(\mathbf{m} + \mathbf{n})$.

Power Set of a Set

Definition 3.7: The power set of a set A, denoted $\mathcal{P}(A)$, is the set of all subsets of A: $\mathcal{P}(A) \overset{\text{def}}{=} \{S | S \subseteq A\}$

If $|A| = k$. Then $|\mathcal{P}(A)| = 2^k$.

The Cartesian Product of Sets

Definition 3.8: The Cartesian product $A \times B$ of two sets A and B is the set of all ordered pairs with the first component from A and the second component from B: $A \times B \overset{\text{def}}{=} \{(a,b) | a \in A \land b \in B\}$.

$|A \times B| = |A| \cdot |B|$.

relations

the Relation Concept

Definition 3.9: 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 $A = B$, ρ is called a relation on A.

Insetad of $(a,b) \in \rho$ one usually write $a \rho b$ (and $(a,b) \notin \rho$: $a \not\rho b$).

Definition 3.10: For any set A, the identity relation on A, denoted id_A (or simply id), is the realation $\text{id}_A = \{(a,a) | a \in A\}$.

There are 2^{n^2} different relations on a set o fcardinality n.

The relation concept can be generalized from binary to k-ary relations. Such realtions play an important role in modeling relational databases.

Representing Relations

For finite sets A and B, ρ from A to B can be represented as a boolean $|A| \times |B|$ matrix M^ρ with rows and columns labeled by the elements of A and B respectively. For $a \in A$ and $b \in B$, $M^\rho_{ab} = 1 \overset{\text{def}}{\Longleftrightarrow} a \rho b$.

Alternatively, directed graph $G = (V, E)$ with $|A| + |B|$ vertices labeled by the elements of A and B. $(a,b) \in E \overset{\text{def}}{\Longleftrightarrow} a \rho b$. Such a graph may contain loops, for instance if ρ on some set.

Set Operations on Relations

...

The Inveres of a Relation

Definition 3.11: The inverse of a relation ρ from A to B is the relation $\hat{\rho}$ from B to A defined by $\hat{\rho} \overset{\text{def}}{=} \{(b,a) | (a,b) \in \rho\}$.

For all a, b we have $b \hat{\rho} a \Leftrightarrow a \rho b$. Alternative for $\hat{\rho}$ is ρ^{-1}.

Composition of Relations

Definition 3.12: ρ relation from A to B. σ relation from B to C. Then, the composition of ρ and σ, denoted $\rho \circ \sigma$ (or also $\rho\sigma$), is the relation from A to C defined by $\rho \circ \sigma \overset{\text{def}}{=} \{(a,c) | \exists b ((a,b) \in \rho \land (b,c) \in \sigma)\}$.

Lemma 3.7: The composition of relations is associative. $\rho \circ (\sigma \circ \phi) = (\rho \circ \sigma) \circ \phi$.

In matrix representation: Matrix multiplication with all entries > 1 set to 1. Graph representation: $a \rho \sigma c$ if and only if path from a to c.

Lemma 3.8: ρ form A to B. σ from B to C. $\widehat{\rho\sigma} = \hat{\sigma}\hat{\rho}$.

Special Properties of Relations

Definition 3.13: ρ on A is reflexive if $a \rho a$ is true for all $a \in A$: $id \subseteq \rho$.

Matrix representation: Diagonal only contains 1. Graph: All loops.

Definition 3.14: ρ on A irreflextive if $a \not\rho b$ for all $a \in A$. $\rho \cap id = \varnothing$.

Definition 3.15: ρ on A is symmetric if $a \rho b \Leftrightarrow b \rho a$ for all $a, b \in A$: $\rho = \hat{\rho}$.

Matrix representation: matrix symmetric. Graph: undirected graph.

Definition 3.16: ρ on A antisymmetric if $a\rho b \wedge b\rho a \Rightarrow a = b$ is true for all $a, b \in A$: $\rho \cap \hat{\rho} \subseteq id$.
Graph: no cycle of length 2.

Definition 3.17: ρ on A is transitive if $a\rho b \wedge b\rho c \Rightarrow a\rho c$ is true for all $a, b, c \in A$.

Lemma 3.9: ρ transitive if and only if $\rho^2 \subseteq \rho$.

Transitive Closure

$\rho^n \subseteq \rho$ for $n > 1$.

Definition 3.18: The transitive closure of a relation ρ on a set A, denoted ρ^*, is $\rho^* = \bigcup_{n \in \mathbb{N}\setminus\{0\}} \rho^n$.

Graph: $a\rho^k b$ if and only if walk of length k from a to b. Transitive closure is the reachability relation. $a\rho^* b$ if and only if thre is a path from a to b.

equivalence relations

Definition of Equivalence Relation

Definition 3.19: An equivalence relation is a relation on a set A that is reflexive, symmetric, and transitive.

Definition 3.20: For an equivalence relation θ on a set A and for $a \in A$, the set of elements of A that are equivalent to a is called the equivalence class of A and is denoted $[a]_\theta$: $[a]_\theta \stackrel{\text{def}}{=} \{b \in A | b\theta a\}$.

Lemma 3.10: The intersection of two equivalence realtions (on the same set) is an equivalence relation.

Equivalence Classes Form a Partition

Definition 3.21: A partition of a set A is a set of mutually disjoint subsets of A that cover A. $\{S_i | i \in \mathcal{I}\}$ of sets S_i satisfying $S_i \cap S_j = \varnothing$ for $i \neq j$ and $\bigcup_{i \in \mathcal{I}} S_i = A$.

Relation $\equiv$: Two elementents are $\equiv$-related if and only if they are in the same set of the partition.

Definition 3.22: The set of equivalence classes of an equivalence relation θ, denoted by $A/\theta \stackrel{\text{def}}{=} \{[a]_\theta | a \in A\}$ is called the quotient set of A by θ, or simply A modulo θ, or $A \bmod \theta$.

Theorem 3.11: The set A/θ of equivalence classes of an equivalence relation θ on A is a partition of A.

Example: Definition of the Rational Numbers

$A = \mathbb{Z} \times (\mathbb{Z}\setminus\{0\})$. We define $\sim$ with $(a,b) \sim (c,d) \stackrel{\text{def}}{\Longleftrightarrow} ad = bc$. It can be shown that $\sim$ is reflexive, symmetric, and transitive. To every equivalence class $[(a,b)]$ we associate the rational number a/b. Thus, $\mathbb{Q} \stackrel{\text{def}}{=} (\mathbb{Z} \times (\mathbb{Z}\setminus\{0\}))/\sim$.

partial order relations

Definition

Definition 3.23: A partial order (or simply order relation) on a set A is a relation that is reflexive, antisymmetric, and transitive. A set A together with a partial order $\preceq$ on A is called partially ordered set (or simply poset) and is denoted as $(A; \preceq)$.

$a \prec b \stackrel{\text{def}}{\Longleftrightarrow} a \preceq b \wedge a \neq b$.

Definition 3.24: For a poset $(A; \preceq)$, two elements a and b are called comparable if $a \preceq b$ or $b \preceq a$; otherwise, they are called incomparable.

Definition 3.25: If any two elements of a poset $(A; \preceq)$ are comparable, then A is called totally ordered (or linearly ordered) by $\preceq$.

Hasse Diagrams

Definition 3.26: 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 $c \prec b$.

Definition 3.27: The Hasse diagram of (finite) poset $(A; \preceq)$ is the directed graph whose vertices are labeled with the elements of A and where there is an edge from a to b if and only if b covers a.

It is usually drawn such that whenever $a \prec b$, b is places higher than a. Then, all arrows are directed upwards and can be omitted.

Combinations of Posets and the Lexicographic Order

Definition 3.28: For given posets $(A; \preceq)$ and $(B; \sqsubseteq)$, their direct produce denoted $(A; \preceq) \times (B; \sqsubseteq)$, is the set $A \times B$ with the relation $\leq$ (on $A \times B$) defined by $(a_1 b_1) \leq (a_2, b_2) \stackrel{\text{def}}{\Longleftrightarrow} a_1 \preceq a_2 \wedge b_1 \sqsubseteq b_2$.

Theorem 3.12: $(A; \preceq) \times (B; \sqsubseteq)$ is a partially ordered set.

Theorem 3.13: For given posets $(A; \preceq)$ and $(B; \sqsubseteq)$, the relation $\leq_{lex}$ defined on $A \times B$ by $(a_1, b_1) \leq_{lex} (a_2, b_2) \stackrel{\text{def}}{\Longleftrightarrow} a_1 \prec a_2 \vee (a_1 = a_2 \wedge b_1 \sqsubseteq b_2)$ is a partial order relation.
If both $(A; \preceq)$ and $(B; \sqsubseteq)$ are totally ordered, then so is $\leq_{lex}$.

Special Elements in Posets

Definition 3.29: $(A; \preceq)$ poset. $S \subseteq A$. Then:

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

Definition 3.30: A poset $(A; \preceq)$ is well-ordered if it is totally ordered and if every non-empty subset of A has a least element.
Note: eveyr totally ordered finite poset is well-ordered.

Meet, Join, and Lattices

Definition 3.31: Let $(A; \preceq)$ be a poset. If a and b 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 $a \vee b$.

Definition 3.32: A poset $(A; \preceq)$ in which every pair of elements has a meet and a join is called a lattice.

functions

Functins are a special type of relation.

Definition 3.33: A function $F : A \to B$ from a domain A to a codomain B is a relation from A to B with the special properties:

1. $\forall a \in A, \exists b \in B$: afb (F is totally defined)
2. $\forall a \in A, \forall b, b' \in B$: $(afb \wedge afb' \to b = b')$ (f is well-defined)

Definition 3.34: The set of all functions $A \to B$ is denoted B^A.

Definition 3.35: A partial function $A \to B$ is a relation from A to B such that condition 2. above holds.
Two (partial) functions with common domain A and codomain B are equal if they are equal as relations.

Definition 3.36: For a function $f : A \to B$ and a subset S of A, the image of S under f, dnoted $f(S)$, is the set $f(S) \stackrel{\text{def}}{=} \{f(a) | a \in S\}$.

Definition 3.37: The subset $f(A)$ of B is called the image (or range) of f and is also denoted $Im(f)$.

Definition 3.38: For a subset T of B, the preimage of T, denoted $f^{-1}(T)$, is the set of values in A that ap into T: $f^{-1}(T) \stackrel{\text{def}}{=} \{a \in A | f(a) \in T\}$

Definition 3.39: $f : A \to B$ is called

1. injective (or on-to-one/an injection) if for $a \neq b$, we have $f(a) \neq f(b)$
2. surjective (or onto) if $f(A) = B$ - for every $b \in B$, $b = f(a)$ for some $a \in A$
3. bijective (or a bijection) if it is both injective and surjective

Definition 3.40: For a bijective function f, the inverse is called the inverse function of f, usually denoted as f^{-1}.

Definition 3.41: The composition of a function $f : A \to B$ and a function $g : B \to C$, denoted $g \circ f$ or simply gf, is defined by $(g \circ f)(a) = g(f(a))$.
Notice that this notation is ambigous. Because the order for notation is different than the one used for compositions or relations.

Lemma 3.14: Function composition is associative: $(h \circ g) \circ f = h \circ (g \circ f)$.

countable and uncountable sets

Countability of Sets

Definition 3.42:

- Two sets A, B are equinumerous ($A \sim B$) if there exists a bijection $A \to B$.
- The set B dominates the set A ($A \preceq B$) if $A \sim C$ for some subset $C \subseteq B$/an injection $A \to B$ exists.
- A set A is called countable if $A \preceq \mathbb{N}$, and uncountable otherwise.

Lemma 3.15: (i) - The relation $\preceq$ is transitive. & (ii) - $A \subseteq B \Rightarrow A \preceq B$.

Theorem 3.16 - Bernstein-Schröder theorem: $A \preceq B \wedge B \preceq A \Rightarrow A \sim B$.

Between Finite and Countably Infinite

For finite A, B: $A \sim B \Leftrightarrow |A| = |B|$.

Theorem 3.17: A set A is countable if and only if it is finite or if $A \sim \mathbb{N}$. ((Re)Phrased: There is no cardinality level between finite and countably infinite.)

Important Countable Sets

Theorem 3.18: The set $\{0,1\}^* \stackrel{\text{def}}{=} \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$ of finitte binary sequences is countable.
Proof: 1 at beginning - standard binary interpretation

Theorem 3.19: $\mathbb{N} \times \mathbb{N} (= \mathbb{N}^2)$ (set of ordered pairs of natural numbers) is countable.
Proof: $k + m = t - 1, m = n - \binom{t}{2}, t > 0$ (diagonals, bot to top)

Corollary 3.20: The Cartesian product $A \times B$ of two countable sets A and B is countable: $A \preceq \mathbb{N} \wedge B \preceq \mathbb{N} \Rightarrow A \times B \preceq \mathbb{N}$.

Corollary 3.21: The rational numbers $\mathbb{Q}$ are countable.

Theorem 3.22: A and A_i for $i \in \mathbb{N}$ be countable sets.

- For any $n \in \mathbb{N}$, the set A^n of n-tuples over A is countable.
- The union $\bigcup_{i \in \mathbb{N}} A_i$ of a countable list $A_0, A_1, \ldots$ of countable sets is countable.
- The set A^* of finite sequences of elements from A is countable.

Uncountability of $\{0,1\}^\infty$

Definition 3.43: $\{0,1\}^\infty$ set of semi-infinite binary sequences (or, equivalencly, the set of functions $\mathbb{N} \to \{0,1\}$.

Theorem 3.23: The set $\{0,1\}^*$ is uncountable.
Proof by Cantor's diagonalization argument.
Also note generally: $\mathbb{N} \prec \{0,1\}^\infty \sim \mathbb{R} \sim \mathbb{R} \times \mathbb{R} \prec \mathcal{P}(\mathbb{R})$.

Existence of Uncomputable Functions

Definition 3.44: A function $f : \mathbb{N} \to \{0,1\}$ is called computable if there is a program that, for every $n \in \mathbb{N}$, when given n as input, outputs $f(n)$.

Corollary 3.24: There are uncomputable function $\mathbb{N} \to \{0,1\}$.
One program: One function at most. Uncountably many functions. Only countably many programs (finite bit-strings).
Halting problem: Program with program as input. Uncomputable, whether terminates

number theory

introduction

Mathematical theory of the natural numbers. Integers are informally considered here. A formal treatment is beyond the scope of this course.

divisors and division

Divisors

Definition 4.1: For integers a and b we say that a divides b, denoted $a|b$, if there exists an integer c such that $b = ac$. In this case, a is called a divisor or b, and b is called a multiple of a. If $a \neq 0$ and a divisor exists, c is called the quotient when b is divided by a, and we write $c = \frac{b}{a}$ or $c = b/a$. We write $a \not| b$ if a does not divide b.

Division with Remainders

Theorem 4.1 - Euclid: For all integers a and $d \neq 0$ there exist unique integers q and r satisfying $a = dq + r$ and $0 \leq r < |d|$.
a: dividend, d: divisor, q: quotient, $r (= R_d(a) = a \bmod d)$: remainder

Gretest Common Divisors

Definition 4.2: For integers a and b (not both 0), an integer d is called a greatest common divisor of a and b if d divides both a and b and if every common divisor of a and b divides d: $d|a \wedge d|b \wedge \forall c((c|a \wedge c|b) \to c|d)$.
For integers two ggd: $\pm$. For other rings more.

Definition 4.3: For $a, b \in \mathbb{Z}$ (not both 0) one denotes the unique positive greatest common divisor by $gcd(a,b)$. If $gcd(a,b) = 1$, then a and b are relatively prime (teilerfremd).

Lemma 4.2: For any integers $, mn, q$ we have $gcd(m, n - qm) = gcd(m, n)$.
Implies: $gcd(m, R_m(n)) = gcd(m, n) \to$ Euclid's gcd-algorithm.

Definition 4.4: For $a, b \in \mathbb{Z}$, the ideal generated by a and b, denoted (a, b), is the set $(a,b) := \{ua + vb | u, v \in \mathbb{Z}\}$. Similarly, the ideal generated by a single integer a is $(a) := \{ua | u \in \mathbb{Z}\}$.

Lemma 4.3: For $a, b \in \mathbb{Z}$ there exists $d \in \mathbb{Z}$ such that $(a, b) = (d)$.

Lemma 4.4: Let $a, b \in \mathbb{Z}$ (not both 0). If $(a, b) = (d)$, then (d) is a greatest common divisor of a and b.

Corollary 4.5: For $a, b \in \mathbb{Z}$ (not both 0), there exist $u, v \in \mathbb{Z}$ such that $gcd(a, b) = ua + vb$.
To determine u, v, consider extended Euclid's algorithm for $gcd(a, b)$ (preferably) with $a > b$:

$$r_0 = a, s_0 = 1, t_0 = 1$$
$$r_1 = b, s_1 = 0, t_1 = 1$$
$$\ldots$$
$$r_{i+1} = r_{i-1} - q_i r_i (0 \leq r_{i+1} < |r_i|), \text{(defining } q_i)$$
$$s_{i+1} = s_{i-1} - q_i s_i, \qquad t_{i+1} = t_{i-1} + q_i t_i$$

Stop, when $r_{k+1} = 0$: $gcd(a, b) = r_k = as_k + bt_k$.

Least Common Multiples

Definition 4.5: The least common multiple l of two positive integers a and b, denoted $l = lcm(a, b)$, is the common multiple of a and b which divides every common multiple of a and b: $a|l \wedge b|l \wedge \forall m((a|m \wedge b|m) \to l|m)$.

factorization into primes

Not exam-relevant

some basic facts about primes

Not exam-relevant

congruences and modular arithmetics

Modular Congruences

Definition 4.8: For $a, b, m \in \mathbb{Z}$ with $m \geq 1$, we say that a is congruent to b modulo m if m divides $a - b$. We write $a \equiv b(\mod m)$ or simply $a \equiv_m b$: $a \equiv_m b \overset{\text{def}}{\iff} m|(a-b)$.

Lemma 4.13: For any $m \geq 1$, $\equiv_m$ is an equivalence relation.

$a \not\equiv_m b \Rightarrow a \neq b$.

Lemma 4.14: If $a \equiv_m b$ and $c \equiv_m d$, then $a + c \equiv_m b + d$ and $ac \equiv_m bd$.

Corollary 4.15: Let $f(x_1, ..., x_k)$ be a multip-variate polynomial in k variables with integer coefficients, and let $m \geq 1$. If $a_i \equiv_m b_i$ for $1 \leq i \leq k$, then: $f(a_1, ..., a_k) \equiv_m f(b_1, ..., b_k)$.

Modular Arithmetic

m equivalence classes of $\equiv_m$: $[0], [1], ..., [m-1]$. Each $[a]$ has a natural representative $R_m(a) \in [a]$ in $\mathbb{Z}_m$.

Lemma 4.16: For any $a, b, m \in \mathbb{Z}$ with $m \geq 1$: (i): $a \equiv_m R_m(a)$ & (ii): $a \equiv_m b \Leftrightarrow R_m(a) = R_m(b)$.

Corollary 4.17: Let $f(x_1, ..., x_k)$ be a multivariate polynomial in k variables with integer coefficients, and let $m \geq 1$. Then $R_m(f(a_1, ..., a_k)) = R_m(f(R_m(a_1), ..., R_m(a_k)))$.

Multiplicative Inverses

Lemma 4.18: The congruence equation $ax \equiv_m 1$ has a solution $x \in \mathbb{Z}_m$ if and only if $gcd(a, m) = 1$. The solution is unique.

Definition 4.9: 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 $x \equiv_m 1/a$.

Consider: $ax \equiv_m 1$. We must have $gcd(a, m) = 1$. Also, $gcd(a, m) = ua + vm$ (extended Euclid. Alg.). So, $1 \equiv_m ua + vm$ for some u, v: $1 \equiv_m ua$. Thus, $R_m(u) = x$.

The Chinese Remainder Theorem

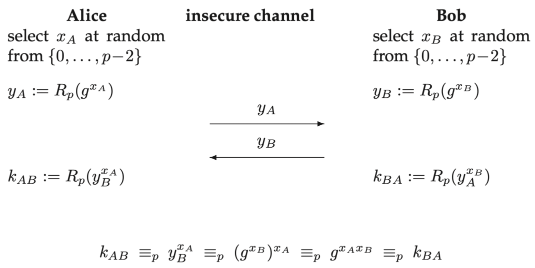
Theorem 4.19: Let $m_1, m_2, ..., m_r$ be pairwise relatively prime integers and let $M = \prod_{i=1}^{r} m_i$. For every list $a_1, ..., a_r$ with $0 \leq a_i < m_i$ for $1 \leq i \leq r$, the system of congruence equations

$$x \equiv_{m_1} a_1$$
$$x \equiv_{m_2} a_2$$
$$...$$
$$x \equiv_{m_r} a_r$$

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

Diffie-Hellman Key-Agreement

Diffie and Hellman proposed public-key encryption in a seminal 1976 paper. This solves the key distribution problem. The security of the Diffie-Hellman protocol is based on the asymetry in computation difficulty - it requires a one-way function, which is easy to compute in one direction but computationally very hard to invert. Specifically: $y = R_p(g^x)$ with p a very large prime (2048 bits for example). y is easily computable even if p, g, x are very large numbers. Computing x when given p, g, y is generally (believed to be) computationally infeasible. The prime p and the basis g are public parameters. The communicatino must be authenticated, but not secret.



Algebra

introduction

Mathematical study of structures consting of a set and certain operations on the set. Goal: understanding such algebraic systems at the highest level of generality and abstraction.

Algebraic Structures

Definition 5.1: An operation on a set S is a function $S^n \to S$, where $n \geq 0$ is called the "arity" of the operation.

Operations with arity 1 and 2 are called unary and binary operations, respectively. An operation with 0 arity is called a constant.

Definition 5.2: An algebra (or algebraic strucutre or Ω-algebra) is a pair $\langle S; \Omega \rangle$ where S is a set (the carrier of the algebra) and $\Omega = (\omega_1, ..., \omega_n)$ is a list of operations on S.

monoids and groups

We consider one binary (and possible one unary and one nullary) operation.

Neutral Element

Definition 5.3: A left [right] neutral element (or identity element) of an algebra $\langle S; * \rangle$ is an element $e \in S$ such that $e * a = a$ [$a * e = a$] for all $a \in S$. If $e * a = a * e = a$ for all $a \in S$, then e is simply called neutral element.

Lemma 5.1: If $\langle S; * \rangle$ has both a left and a right neutral element, then they are equal. In particular $\langle S; * \rangle$ can have at most one neutral element.

Associativity and Monoids

Definition 5.4: A binary operation $*$ on a set S is associative if $a * (b * c) = (a * b) * c$ for all $a, b, c \in S$.

Addition and multiplication are associate operations in $\mathbb{Z}, \mathbb{N}, \mathbb{Q}, \mathbb{R}, \mathbb{Z}_m$.

Definition 5.5: A monoid is an algebra $\langle M; *, e \rangle$ where $*$ is associative and e is the neutral element.

$\mathbb{Z}, \mathbb{N}, \mathbb{Q}, \mathbb{R}, \mathbb{Z}_m$ with addition (neutral element 0) and multiplication (neutral element 1) respectively are monoids.

Inverses and Groups

Definition 5.6: A left [right] inverse element of an element a in an algebra $\langle S; *, e \rangle$ with neutral element e is an element $b \in S$ such that $b * a = e$ [$a * b = e$]. If $b * a = a * b = e$, then b is simply called an inverse of a.

Lemma 5.2: In a monoid $\langle M; *, e \rangle$, if $a \in M$ has a left and a right inverse, then they are equal. In particular, a has at most one inverse.

Definition 5.7: A group is an algebra $\langle G; *, \hat{}, e \rangle$ satisfying the follwoing axioms:

1. $*$ is associative
2. e is a neutral element
3. Every $a \in G$ has an inverse element $\hat{a}$.

For addition (+): inverse $-a$, neutral element 0. For multiplication: inverse a^{-1} or $1/a$, neutral element: 1.

We have $\langle \mathbb{N}; + \rangle$, $\langle \mathbb{Z}; + \rangle$, $\langle \mathbb{Q}; + \rangle$, $\langle \mathbb{Q} \backslash \{0\}; \cdot \rangle$, $\langle \mathbb{R}; + \rangle$, $\langle \mathbb{R} \backslash \{0\}; \cdot \rangle$, $\langle \mathbb{Z}_m; \oplus \rangle$.

Definition 5.8: A group $\langle G; * \rangle$ (or monoid) is called commutative or abelian if $a * b = b * a$ for all $a, b \in G$.

Lemma 5.3:

1. $\hat{\hat{a}} = a$
2. $\widehat{a * b} = \hat{b} * \hat{a}$
3. Left cancellation law: $a * b = a * c \Rightarrow b = c$
4. Right cancellation law: $b * a = c * a \Rightarrow b = c$
5. $a * x = b$ [$x * a = b$] has a solution for any a and b

(Nonn)minimality of the Group Axioms

The above aximos may be simplified. Replace **G2** with **G2'** ($a * e = a$) and **G3** with **G3'** ($\hat{a} * a = e$). Then, **G1**, **G2'**, **G3'** imply **G2** and **G3**.

Some Examples of Groups

Examples irrelevant.

the structure of groups

Direct Products of Groups

Definition 5.9: The direct product of n groups $\langle G_1; *_1 \rangle$, ..., $\langle G_n; *_n \rangle$ is the algebra $\langle G_1 \times G_2 \times ... \times G_n; \star \rangle$, where the operation $\star$ is component wise: $(a_1, ..., a_n) \star (b_1, ..., b_n) = (a_1 *_1 b_1, ..., a_n *_n b_n)$.

Lemma 5.4: $\langle G_1 \times ... \times G_n; \star \rangle$ is a group, where the neutral element and the inversion operation are component-wise in the respective groups.

Group Homomorphisms

Definition 5.10: For two groups $\langle G; *, \hat{}, e \rangle$ and $\langle H; \star, \sim, e' \rangle$, a function $\psi : G \to H$ is called a group homomorphism if, for all a and b, $\psi(a * b) = \psi(a) \star \psi(b)$. If ψ is a bijection from G to H, then it is called an isomorphism, and we say that G and H are isomorphic and write $G \simeq H$.

Lemma 5.5: A group homomorphism ψ from $\langle G; *, \hat{}, e \rangle$ to $\langle H; \star, \sim, e' \rangle$ satisfies (i) $\psi(e) = e'$ and (ii) $\psi(\hat{a}) = \widetilde{\psi(a)}$ for all a.

Subgroups

Definition 5.11: 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., if 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 $a \in H$.

The Order of Group Elements and of a Group

Definition 5.12: G a group. $a \in G$. The order of a, denoted $ord(a)$, is the least $m \geq 1$ such that $a^m = e$, if such an m exists, and $ord(a)$ is said to be infinite otherwise, written $ord(a) = \infty$.

If $ord(a) = 2$ for some a: $a^{-1} = a$. (self-inverse)

Lemma 5.6: In a finite group G, every element has a finite order.

Definition 5.13: For a finite group G, $|G|$ is called the order of G.

Cyclic Groups

Definition 5.14: For a group G and $a \in G$, the group generated by a, denoted $\langle a \rangle$ is defined as $\langle a \rangle \overset{\text{def}}{=} \{a^n | n \in \mathbb{Z}\}$.

$\langle a \rangle$ is the smallest subgroup of G containing $a \in G$.

Definition 5.15: A group $G = \langle g \rangle$ generated by an element $g \in G$ is called cyclic, and g is called a generator of G.

There may be multiple generators. g^{-1} is always a generator too.

Theorem 5.7: A cyclic group of order n is isomorphic to $\langle \mathbb{Z}_n; \oplus \rangle$ (and hence abelian).

Application: Diffie-Hellman for General Groups

Was described before for $\mathbb{Z}_p^*$ (for notation see below). Works as well in any cyclic group $G = \langle g \rangle$ for which computing x from g^x is computationally infeasible.

Also, elliptic curves are an important class of cyclic groups used in cryptography.

The Order of Subgroups

Theorem 5.8 - Lagrange: Let G be a finite group and let H be a subgroup of G. Then the order of H divides the order of G.

Corollary 5.9: For a finite group G, the order of every element divides the group order, i.e., $ord(a)$ divides $|G|$ for every $a \in G$.

Corollary 5.10: Let G be a finite group. Then $a^{|G|} = e$ for every $a \in G$.

Corollary 5.11: Every group of prime order is cyclic, and in such a group every element except the neutral element is a generator.

The Group $\mathbb{Z}_m^*$ and Euler's Function

Definition 5.16: $\mathbb{Z}_m^* \overset{\text{def}}{=} \{a \in \mathbb{Z}_m | gcd(a, m) = 1\}$.

That so that we have a group. Because $a \in \mathbb{Z}_m$ has a multiplicative inverse if and only if $gcd(a, m) = 1$.

Definition 5.17: The Euler function $\varphi : \mathbb{Z}^+ \to \mathbb{Z}^+$ is defined as the cardinality of $\mathbb{Z}_m^*$: $\varphi(m) = |\mathbb{Z}_m^*|$.

If p is prime: $\mathbb{Z}_p^* = \{1, ..., p-1\} = \mathbb{Z}_p \backslash \{0\}$. Hence, $\varphi(p) = p - 1$.

Lemma 5.12: If the prime factorization of m is $m = \prod_{i=1}^{r} p_i^{e_i}$, then $\varphi(m) = \prod_{i=1}^{r} (p_i - 1) p_i^{e_i - 1}$.

Theorem 5.13: $\langle \mathbb{Z}_m^*; \odot, ^{-1}, 1 \rangle$ is a group.

Corollary 5.14 - Fermat, Euler: For all $m \geq 2$ and all a with $gcd(a, m) = 1$: $a^{\varphi(m)} \equiv_m 1$. In particular, for every prime p and every a not divisible by p: $a^{p-1} \equiv_p 1$.

Theorem 5.15: The group $\mathbb{Z}_m^*$ is cyclic if and only if $m = 2$, $m = 4$, $m = p^e$, $m = 2p^e$, where p is an odd prime and $e \geq 1$.

RSA public-key encryption

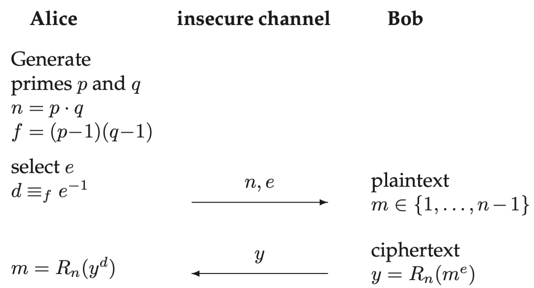
e-th Roots in a Group

Theorem 5.16: G some finite group. $e \in \mathbb{Z}$ relatively prime to $|G|$. The function $x \mapsto x^e$ is a bijection and the (unique) 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 $|G|$: $ed \equiv_{|G|} 1$.

$|G|$ known, d computable from $ed \equiv_{|G|} 1$ with the extended Euclidean algorithm. No general method is known for computing e-th roots in a group G without knowing its order.

Description of RSA

We consider $\mathbb{Z}_n^*$ with $n = pq$, p and q being two sufficiently large secret primes. Then: $|\mathbb{Z}_n^*| = \varphi(n) = (p-1)(q-1)$. The order can only be managably computed if the (secret) prime factors p and q of n are known.



The (public) encryption transformation is defined by $m \mapsto y = R_n(m^e)$. The (secret) decryption transformation is defined by $y \mapsto m = R_n(y^d)$. d can be computed according to $ed \equiv_{(p-1)(q-1)} 1$.
That is the naive approach (being deterministic etc.). The message m is usually a short-term encryption key.

On the Security of RSA

First, it is widely believed that computing e-th roots modulo n is computationally equivalent to factoring n/large integers - but not definitely known. Without a major breakthrough and processor speed developing as predicted, a 2048-bit modulus seems secure for another 15 years. Larger modulo are secure much longer.
Note that RSA is only (believed to be) secure if the communication channel is authenticated. If an adversary can interfere with the data traffic, it can just provide its own keys to both parties and 'mediate' to listen. This is usually solved with publick-key certificates signed by a trusted authority.
Also, the message must be randomized for RSA to be secure. Otherwise, an adversary could simply encrypt messages itself and comparing them with the encrypted messages. For a small message space this allows to break the system.

Digital Signatures

Signature can only be created by the entity knowing the secrt key. Can be verified by anyone knowing the public key. Message: m. $z = m||h(m)$ (h introduces redundancy), $z \in \mathbb{Z}_n$. Signature $s = R_n(z^d)$. Verification: checking $R_n(s^e) = m||h(m)$.

rings and fields

Now: two binary operations, usually called addition and multiplication.

Definition of a Ring

Definition 5.18: A ring $\langle R; +, -, 0, \cdot, 1\rangle$ is an algebra for which

1. $\langle R; +, -, 0\rangle$ is a commutative group
2. $\langle R; \cdot, 1\rangle$ is a monoid
3. $a(b+c) = (ab) + (ac)$ and $(b+c)a = (ba) + (ca)$ for all $a, b, c \in R$.

Commutative ring: multiplication is commutative ($ab = ba$).
Lemma 5.17: For any ring $\langle R; +, -, 0, \cdot, 1\rangle$, and for all $a, b \in R$:

1. $0a = a0 = 0$
2. $(-a)b = -(ab)$
3. $(-a)(-b) = ab$
4. R non-trivial $\Rightarrow 1 \neq 0$

Definition 5.19: The characteristic of a ring is the order of 1 in he additive group if it is finite, and otherwise the characteristic is defined to be 0 (not infinite).

Unts and the Multiplicative Group of a Ring

Definition 5.20: An element u of a ring R is called a unit if u is invertible: $uv = vu = 1$ for some $v \in R$. The set of units of R is denoted by R^*.
Lemma 5.18: For a ring R, R^* is a multiplicative group (the group of units of R).

Divisors

Definition 5.21: For $a, b \in R$ with $a \neq 0$ we say that a divides b, denoted $a|b$, if there exists $c \in R$ such that $b = ac$. In this case, a is called a divisors of b and b is called a multiple of a.
All non-zero elements divise 0. $1/-1$ divise every element.
Lemma 5.19: In any commutative ring:

- $a|b$ and $b|c \Rightarrow a|c$ (transitivity of $|$)
- $a|b \Rightarrow a|bc$ for all c
- $a|b$ and $a|c \Rightarrow a|(b+c)$

Definition 5.22: For ring elements a and b (not both 0), a ring element d is called a greatest common divisor of a and b if d divides both a and b and if every common divisor of a and b divides d: $d|a \wedge d|b \wedge \forall c((c|a \wedge c|b) \to c|d)$.

Zeordivisors and Integral Domains

Definition 5.23: An element $a \neq 0$ of a commutative ring R is called a zerodivisor if $ab = 0$ for some $b \neq 0$ in R.
Definition 5.24: An integral domain is a (nontrivial, $1 \neq 0$) commutative ring without zerodivisors: $\forall a \forall b(ab = 0 \to a = 0 \vee b = 0)$.
Lemma 5.20: In an integral domain, if $a|b$, then c with $b = ac$ is unique (denoted $c = \frac{b}{a}$ or $c = b/a$ and called quotient)

Polynomial Rings

Definition 5.25: A polynomial $a(x)$ over a commutative ring R in the indeterminate x is a formal expression of the form $a(x) = a_d x^d + a_{d-1}x^{d-1} + ... + a_1 x + a_0 = \sum_{i=0}^{d} a_i x^i$ for some non-negative integer d, with $a_i \in R$. The degree of $a(x)$, denoted $deg(a(x))$, is the greatest i for which $a_i \neq 0$. The special polynomial 0 is defined to have degree "minus infinity". Let $R[x]$ denote the set of polynomials (ni x) over R.
Actually better to understand polynomials as finite lists $(a_0, a_1, ..., a_{d-1}, a_d)$. Addition: $a(x) + b(x) = \sum_{i=0}^{max(d,d')}(a_i + b_i)x^i$. Multiplication: as usual. Degree of product at most sum of degrees. If R integral domain, exactly sum.
Theorem 5.21: For any commutative ring R, $R[x]$ is a commutative ring.
Lemma 5.22: (i) If D is an integral domain, then so is $D[x]$. (ii) The units of $D[x]$ are the constant polynomials that are units of D: $D[x]^* = D^*$.

Fields

Definition 5.26: A field is a nontrivial commutative ring F in which every nonzero element is a unit. ($F^* = F\backslash\{0\}$).
F is a field if and only if $\langle F\backslash\{0\}; \cdot, ^{-1}, 1\rangle$ is an abelian group.
Theorem 5.23: $\mathbb{Z}_p$ is a field if and only if p is prime.
Theorem 5.24: A field is an integral domain.

polynomials over a field

F field. $F[x]$ ring. - as F commutative, also $F[x]$ commutative.

Factorization and Irreducible Polynomials

Definition 5.27: A polynomial $a(x) \in F[x]$ is called monic if the leadin coefficient is 1.
Definition 5.28: A polynomial $a(x) \in F[x]$ with degree at least 1 is called irreducible if it is divisible only by constant polynomials and by constant multiples of $a(x)$.

- Polynomial of degree 1: always irreducible.
- Polynomial of degree 2: irreducible of product of two polynomials of degree 1.
- Polynomial of degree 3: irreducible or at least one factor of degree 1.
- Polynomial of degree 4: irreducible or a factor of degree 1 or an irreducible factor of degree 2.

Definition 5.29: The monic polynomial $g(x)$ of largest degree such that $g(x)|a(x)$ and $g(x)|b(x)$ is called the greatest common divisor of $a(x)$ and $b(x)$, denoted $gcd(a(x), b(x))$.

The Division Property in $F[x]$

Theorem 5.25: F a field. For any $a(x)$ and $b(x) \neq 0$ in $F[x]$ there exists a unique $q(x)$ (the quotient) and a unique $r(x)$ (the remainder) such that $a(x) = b(x) \cdot q(x) + r(x)$ and $deg(r(x)) < deg(b(x))$.
$r(x)$ denoted by $R_{b(x)}(a(x))$.

Analogies Between $\mathbb{Z}$ and $F[x]$, Euclidean Domains

Not exam relevant!

Definition 5.30: In an integral domain, a and b are called associates ($a \sim b$) if $a = ub$ for some unit u.
Definition 5.31: In an integral domain, a non-unit $p \in D\backslash\{0\}$ is irreducible if, whenever $p = ab$, then either a or b is a unit. (p only divisible by units/associates)
Units in $\mathbb{Z}$: $1, -1$. Units in $F[x]$: non-zero constant polynomials.
$a \in D$ on associate distinguished. For $\mathbb{Z}$: $|a|$. For $a(x) \in F[x]$: monic polynomial associated with $a(x)$. Only considering distinguished associates for $\mathbb{Z}$: usual notion of primes.
Lemma 5.26: $a \sim b \Leftrightarrow a|b \wedge b|a$
Definition 5.32: A Euclidean domain is an integral domain D together with a so-called degree function d: $D\backslash\{0\} \to \mathbb{N}$ such that:

1. For every a and $b \neq 0$ in D: exists q, r such that $a = bq + r$ and $d(r) < d(b)$ or $r = 0$.
2. For all nonzero $a, b \in D$: $d(a) \leq d(ab)$.

$\mathbb{Z}[i]$ (Gaussian integers) are Euclidean domain with absolte value as degree.
Theorem 5.27: In a Euclidean domain every element can be factored uniquely (up to taking associates) into irreducible elements.

Polynomials as Functions

Polynomial Evaluation

For a ring R, $a(x) \in R[x]$ can be interpreted as a function $R \to R$ by defining evaluation of $a(x)$ at $\alpha \in R$ in the usual manner. This defines $R \to R : \alpha \mapsto a(\alpha)$.
Lemma 5.28: Polynomial evaluation is compatible with the ring operations:

- $c(x) = a(x) + b(x) \Rightarrow c(\alpha) = a(\alpha) + b(\alpha)$ for any α
- $c(x) = a(x) \cdot b(x) \Rightarrow c(\alpha) = a(\alpha) \cdot b(\alpha)$ for any α

Roots

Definition 5.33: Let $a(x) \in R[x]$. An element $\alpha \in R$ for which $a(\alpha) = 0$ is called a root of $a(x)$.
Lemma 5.29: For a field F, $\alpha \in F$ is a root of $a(x)$ if and only if $x - \alpha$ divides $a(x)$.
Corollary 5.30: A polynomial $a(x)$ of degree 2 or 3 over a field F is irreducible if and only if it has no roots.
Theorem 5.31: For a field F, a nonzero polynomial $a(x) \in F[x]$ of degree d has at most d roots.

Polynomial Interpolation

Lemma 5.32: A polynomial $a(x) \in F[x]$ of degree at most d is uniquely determined by any $d+1$ values of $a(x)$.

finite fields

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

$a(x) \equiv_{m(x)} b(x) \stackrel{\text{def}}{\Longleftrightarrow} m(x)|(a(x) - b(x))$
Lemma 5.33: Congruence modulo $m(x)$ is an equivalence relation on $F[x]$, and each equivalence class has a unique representative of degree less than $deg(m(x))$.
Definition 5.34: Let $m(x)$ be a polynomial of degree d over F. Then $F[x]_{m(x)} \stackrel{\text{def}}{=} \{a(x) \in F[x] | deg(a(x)) < d\}$.
Lemma 5.34: Let F be a finite field with q elements and let $m(x)$ be a polynomial of degree d over F. Then $|F[x]_{m(x)}| = q^d$.
Lemma 5.35: $F[x]_{m(x)}$ is a ring with respect to addition and multiplication modulo $m(x)$.
Lemma 5.36: The congruence equation $a(x)b(x) \equiv_{m(x)} 1$ (for a given $a(x)$) has a solution $b(x) \in F[x]_{m(x)}$ if and only if $gcd(a(x), m(x)) = 1$. The solution is unique. In other words, $F[x]^*_{m(x)} = \{a(x) \in F[x]_{m(x)} | gcd(a(x), m(x)) = 1\}$.

Constructing Extension Fields

Theorem 5.37: The ring $F[x]_{m(x)}$ is a field if and only if $m(x)$ is irreducible.
One can show that $\mathbb{R}_{m(x)}$ is isomorphic to $\mathbb{C}$ for every irreducible polynomial of degree 2 over $\mathbb{R}$.
There are not irreducible polynomials of higher degree than 2 over $\mathbb{R}$.
There are not irreducible polynomials of degree > 1 over $\mathbb{C}$.

Some Facts About Finite Fields

Theorem 5.38: For every prime p and every $d \geq 1$ there exists an irreducible polynomial of degree d in $GF(p)[x]$. In particular, there exists a finite field with p^d elements.
Theorem 5.39: There exists a finite field with q elements if and only if q is a power of a prime. Moreover, any two finite fields of the same size q are isomorphic.
Theorem 5.40: The multiplicative group of every finite field $GF(q)$ is cyclic.
Multiplicative group of $GF(q)$ has order $q-1$ and $\varphi(q-1)$ generators.

Application: Error-Correcting Codes

On application of finite fields in CS.

Definition of Error-Correcting Codes

Two problems: erased data & errors in data. Second more severe as unkonwn.
Definition 5.35: A (n, k)-encoding function E for some alphabet $\mathcal{A}$ is an injective function that maps a list $(a_0, ..., a_{k-1}) \in \mathcal{A}^k$ of k (information) symbols to a list $(c_0, ..., c_{n-1}) \in \mathcal{A}^n$ of $n > k$ (encoded) symbols in $\mathcal{A}$, called codeword: $E : \mathcal{A}^k \to \mathcal{A}^n : (a_0, ..., a_{k-1}) \mapsto E((a_0, ..., a_{k-1})) = (c_0, ..., c_{n-1})$.
$\mathcal{C} = Im(e) = \{E((a_0, ..., a_{k-1})) | a_0, ..., a_{k-1} \in \mathcal{A}\}$ is called an error-correcting code.
Definition 5.36: An (n, k)-error-correcting code over the alphabet $\mathcal{A}$ with $|\mathcal{A}| = q$ is a subset of $\mathcal{A}^n$ of cardinality q^k.
Definition 5.37: The Hamming distance between two strings of equal length over a finite alphabet $\mathcal{A}$ is the number of positions at which two strings differ.
Definition 5.38: The minimum distance of an error-correcting code $\mathcal{C}$, denoted $d_{\min}(\mathcal{C})$, is the minimum of the Hamming distance between any two codewords.

Decoding

Definition 5.39: A decoding function D for an (n, k)-encoding function is a function $D : \mathcal{A}^n \to \mathcal{A}^k$.
Such a function (should be efficiently computable) takes an arbitrary list $(r_0, ..., r_{n-1}) \in \mathcal{A}^n$ and decodes it to the most plausible information vectors $(a_0, ..., a_{k-1})$.
Definition 5.40: A decoding function D is t-error correcting for encoding function E if for any $(a_0, ..., a_{k-1})$: $D((r_0, ..., r_{n-1})) = (a_0, ..., a_{k-1})$ for any $(r_0, ..., r_{n-1})$ with Hamming distance at most t from $E((a_0, ..., a_{k-1}))$.
A code $\mathcal{C}$ is t-error correcting if there exists E and D with $\mathcal{C} = Im(E)$ where D is t-error correcting.
Theorem 5.41: A code $\mathcal{C}$ with minimum distance d is t-error correcting if and only if $d \geq 2t + 1$.

Codes based on Polynomial Evaluation

Theorem 5.42: Let $\mathcal{A} = GF(q)$ and let $\alpha_0, ..., \alpha_{n-1}$ be arbitrary distinct elements of $GF(q)$. Consider the encoding function $E((a_0, ..., a_{k-1})) = (a(\alpha_0), ..., a(\alpha_n - 1))$,

where $a(x)$ is the polynomial $a(x) = a_{k-1}x^{k-1} + ... + a_1 x + a_0$. This code has minimum distance $n - k + 1$.
An (n, k)-code over $GF(2^d)$ can be interpreted as a binary (dn, dk)-code over $GF(2)$. Minimum distance of the binary code $\geq$ original code.

Logic

introduction

Not relevant.

proof systems

Definition

Syntactic objects defined as finite strings over some alphabet. Alphabet Σ. Σ^* set of finite strings over Σ.
Consider statements of certain type & proofs of statements for this type.
Now, fixed statement type. $\mathcal{S} \subseteq \Sigma^*$, set of syntactic representations of mathematical statements of that type. $\mathcal{P} \subseteq \Sigma^*$, set of syntactic representations of proof strings.
$\tau : \mathcal{S} \to \{0, 1\}$ Truth function assigns truth value. Defines semantics.
Proof $p \in \mathcal{P}$ either valid or invalud for some $s \in \mathcal{S}$: $\phi : \mathcal{S} \times \mathcal{P} \to \{0, 1\}$ (1 meaning valid proof for s).
Without loss of generality one can consider $\mathcal{S} = \mathcal{P} = \{0, 1\}^*$. With syntactically wrong statements as false statements.
Definition 6.1: A proof system is a quadruple $\Pi = (\mathcal{S}, \mathcal{P}, \tau, \phi)$.
ϕ has to be efficiently computable for Π to be of any use.
Definition 6.2: A proof system Π is sound if not false statement has a proof: for all $s \in \mathcal{S}$: if $\phi(s, p) = 1$ for some $p \in \mathcal{P} \Rightarrow \tau(s) = 1$.
Definition 6.3: A proof system Π is complete if every true statement has a proof: for all $s \in \mathcal{S}$ with $\tau(s) = 1 \Rightarrow p \in \mathcal{P}$ with $\phi(s, p) = 1$ exists.

Examples

A proof system with efficient verification for the existence of Hamiltonian cycles in graphs exists - just providing a cycle. However, no reaonable sound and complete proof system for the non-existence of Hamiltonian cycles is known to exists.
Now, consider primality. For some number not be be prime, a simple (verifiable) proof is providing a non-trivial divisor. Proving that some number is prime, however, is harder. A proof consists of (1) $p_1, ..., p_k$ distinct prime factors of $n-1$, (2) recursive proof of primality for each $p_1, ..., p_k$, (3) a generator g of the group $\mathbb{Z}_p^*$. For understanding remember that the multiplicative group of any finite field is cyclic and has a generator g.

Discussion

- Proof verification must be efficient. Proof generation generally is not efficient. Requires ingenuity and insight.
- A proof system is always restricted to a certain type of mathematical statement.
- The proof verification method of logic (checking a sequence of rule applications) is only a special case.
- Existence of proof system for certain statement type does not imply existence for negated statement (at least with efficient verification)

Proof Systems in Theoretical Computer Science

$\mathcal{S} = \mathcal{P} = \{0, 1\}^*$. $L \subseteq \{0, 1\}^*$ with $L := \{s | \tau(s) = 1\}$. Hence, L also defines predicate τ.
L: formal language. Problem: prove that s in language: $s \in L$. Proof for $s \in L$: witness w.
Consider W bounded by plynomial in the length of s & ϕ computable in polynomial time in the length of s. NP: Class of languages for which such a polynomial-time computable verification function exists.
Proof system of interest: probabilistically checkable proofs
Interactive proofs: Proof is a protocol/interaction between prover / verifier. Accepts exponentially small probability of verifier accepting proof for a flase statement. Justification

- statements provable, not provable conventionally
- zero-knowledge proofs (verifier can not proof itself)
- relevance for block-chain systems etc.

elementary general concepts in logic

The General Goal of Logic

A goal of logic is to provide a specific proof system Π for which a very large class of matheamtical statements can be expressed as an element of $\mathcal{S}$.
Never, all possible math. statements included. Self-referential statements usually not allowed.
$s \in \mathcal{S}$ consiste of one or more formulas. Proof: sequence of syntactic steps, called derivation or a deduction (step: applying one allowed role). Set of all allowed rules: Calculus.

Syntax, Semantics, Interpretaion, Model

Definition 6.4: The syntax of a logic defines an alphabet Λ (of allowed symbols) and specifies which strings Λ^* are formulas.
Definition 6.5: The semantics of a logic defines (among other things, see below) a function $free$ which assigns to each formula $F = (f_1, f_2, ..., f_k) \in \Lambda^*$ a subset $free(F) \subseteq \{1, ..., k\}$ of the indices. If $i \in free(F)$, then the symbol f_i is said to occur $free$ in F.
Definition 6.6: An interpretation consists of a set $\mathcal{Z} \subseteq \Lambda$ of symbols of Λ, a domain (a set of possible values) for each symbol in $\mathcal{Z}$, and a function that assigns to each symbol in $\mathcal{Z}$ a value in its associated domain.
Definition 6.7: An interpretaion is suitable for a formula F if it assigns a value to all symbols $\beta \in \Lambda$ occuring free in F.
Definition 6.8: The semantics of a logic also defines a function σ assigning to each formula F, and each interpretation $\mathcal{A}$ suitable for F, a truth value $\sigma(F, \mathcal{A})$ in $\{0, 1\}$. In threatments of logic one often writes $\mathcal{A}(F)$, which is called the truth value of F under interpretation $\mathcal{A}$.
Definition 6.9: A (suitable) interpretation $\mathcal{A}$ for which a formula F is true is called a model for F, and one also write $\mathcal{A} \models F$. For a set M of formulas, a (suitable) interpretation for which all formulas in M are true is called a model for M, denoted $\mathcal{A} \models M$.

Connection to Proof Systems

Often logic is treated informally, but there are two options to foramlize logic:

- Formulas and interpretations are formas objects. A statement is a pair $(F, \mathcal{A})$. Then, σ corresponds to τ.
- Formulas are formal objects. Statements only refer to general formula (tautology, (un)satisfiable, logical consequence, ...). Foramlization of interpretations is not necessary. (Usual approach, also here.)

Satisfiability, Taugology, Consequence, Equivalence

Definition 6.10: A formula F (or a st M of formulas) is called satisfiable if there exists a model for F (or M), and unsatisfiable otherwise. $\perp$ is used for unsatisfiable formulas.
Definition 6.11: A formula F is called a tautology or valid if it is true for every suitable interpretaion. $\top$ is used for a tautology.
Definition 6.12: A formula G is a logical consequence of a formula F (or a set of formulas), denoted $F \models G$ or $M \models G$ if every interpretation suitable for both F (or M) and G, which is is a model for F (for M), is a model for G.
Definition 2.7: $F \models G \overset{\text{def}}{\Longleftrightarrow}$ all suitable truth assignments to symbols in F, G: value of G must be 1 if value of F is 1.
Definition 6.13: Two formulas F and G are equivalent ($F \equiv G$), if every interpretation suitable for both F and G yields the same truth value for F and G: $F \equiv G \overset{\text{def}}{\Longleftrightarrow} F \models G$ and $G \models F$.
Definition 2.6: In propositional logic, formulas $F \equiv G$ if same function (truth values equal for all truth assignments).
The empty set M correcponds to a tautology.
Definition 6.14: If F is a tautology, one also writes $\models F$.

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

Definition 6.15: If F and G are formulas, then also $\neg F$, $(F \wedge G)$ (conjunction), and $(F \vee G)$ (disjunction) are formulas.
Outermost parentheses and parentheses not needed because of associativity can be dropped. $F \to G$ stands for $\neg F \vee G$. $F \leftrightarrow G$ stands for $(F \wedge G) \vee (\neg F \wedge \neg G)$.
Definition 6.16:

- $\mathcal{A}(F \wedge G) = 1 \overset{\text{def}}{\Longleftrightarrow} \mathcal{A}(F) = 1$ and $\mathcal{A}(G) = 1$
- $\mathcal{A}(F \vee G) = 1 \overset{\text{def}}{\Longleftrightarrow} \mathcal{A}(F) = 1$ or $\mathcal{A}(G) = 1$
- $\mathcal{A}(\neg F) = 1 \overset{\text{def}}{\Longleftrightarrow} \mathcal{A}(F) = 0$

Lemma 6.1: For any formulas F, G, H:

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 rule)
9. $F \vee \top \equiv \top$ and $F \wedge \top \equiv F$ (tautology rules)
10. $F \vee \perp \equiv F$ and $F \wedge \perp \equiv \perp$ (unsatisfiability rules)
11. $F \vee \neg F \equiv \top$ and $F \wedge \neg F \equiv \perp$

Logical Consequence vs. Unsatisfiability

Lemma 6.2 and 2.2: A formula F is a tautology if and only if $\neg F$ is unsatisfiable.
Lemma 6.3 and 2.3: The following three statements are equivalent:

1. $\{F_1, F_2, ..., F_k\} \models G$
2. $(F_1 \wedge F_2 \wedge ... \wedge F_k) \to G$ is tautology
3. $\{F_1, F_2, ..., F_k, \neg G\}$ is unsatisfiable

Theorem and Theories

Four types of statements.

1. Theorem in an axiomatically defined theory.
2. Statements about a formula/a set of formulas.
3. $\mathcal{A} \models F$ for a given interpretation $\mathcal{A}$ and formula F
4. Statements about a logic (calculus being sound, ...)

For the first: Set T of formulas, formulas called axioms of the theory. Any F with $T \models F$ called theorem in theory T.

Extension from Chapter 2

Formulas may be understood as functions. In function tables, one can describe (or define) the value of a formula for all viable interpretations. The concept of function tables is especially useful for propositional logic, where the domain is finite.

logical calculi

Introduction

Proof of a theorem should be a puely syntactic derivation consisting of simple and easily verifiable steps. Step: Derivation of new syntactic object by application of a derivation/inference rule.
Set of rules for manipulation formulas: Calculus.

Hilbert-Style Calculi

Most intuitive type of calculus: Formulas are manipulated.
Definition 6.17: A derivation/inference rule is a rule fo rderiving a formula from a set of formulas (precondition/premises). We write $\{F_1, ..., F_k\} \vdash_R G$ if G can be derived from the set $\{F_1, ..., F_k\}$ by rule R.
Derivation purely syntactic concept.
Definition 6.18: The application of a derivation rule R to a set M of formulas means:

1. Select a subset N of M.
2. For the place-holders in R: specify formulas that appear in N such that $N \vdash_R G$ for a formula G.
3. Add G to the set M ($M \cup \{G\}$).

Definition 6.19: A (logical) calculus K is a finite ste of derivation rules: $K = \{R_1, ..., R_m\}$.
Definition 6.20: A derivation of a formula G from a set M offormulas 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 $1 \leq i \leq n$, where $N \vdash_{R_j} G_i$ for some $N \subseteq M_{i-1}$ and some $R_j \in K$, and where
- $G_n = G$

We write $M \vdash_K G$ if a derivation of G from M exists in K.

Soundness and Completeness of a Calculus

Definition 6.21: A derivation rule R is correct if for every set M of formulas and every formula F: $M \vdash_F \Rightarrow M \models F$.
Definition 6.22: A calculus K is sound/correct if for every set M of formulas and every formula F: $M \vdash_K F \Rightarrow M \models F$. And K is complete if for every M and F: $M \models F \Rightarrow M \vdash_K F$.
K is sound and complete if $M \vdash_K F \Leftrightarrow M \models F$.

Derivation from Assumptions

Lemma 6.4: If $\{F_1, ..., F_k\} \vdash_K G$ holds for a sound calculus, then: $\models ((F_1 \wedge ... \wedge F_k) \to G)$.
For a given calculus one can also prove new derivation rules. A proof pattern may be captured as a new rule.

connection to Proof Systems

Not relevant.

propositional logic

Syntax

Definition 6.23: An atomic formula is a symbol of the form A_i with $i \in \mathbb{N}$. A formula is defined as follows:

- An atomic formula is a formula.
- F and G formulas $\Rightarrow \neg F$, $(F \wedge G)$, $(F \vee G)$ are formulas

Semantics

In propositional logic, the free symbols of a formula are all the atomic formulas.
Definition 6.24: For a set Z of atomic formulas, an interpretation $\mathcal{A}$ (called truth assignment) is a function $\mathcal{A} : Z \to \{0, 1\}$. $\mathcal{A}$ is suitable for F if Z contains all atomic formulas appearing in F. The sematntics is defined by $\mathcal{A}(F) = \mathcal{A}(A_i)$ for any atomic formula $F = A_i$ and:

- $\mathcal{A}((F \wedge G)) = 1 \overset{\text{def}}{\iff} \mathcal{A}(F) = 1$ and $\mathcal{A}(G) = 1$
- $\mathcal{A}((F \vee G)) = 1 \overset{\text{def}}{\iff} \mathcal{A}(F) = 1$ or $\mathcal{A}(G) = 1$
- $\mathcal{A}(\neg F) = 1 \overset{\text{def}}{\iff} \mathcal{A}(F) = 0$

Normal Forms

Definition 6.25: A literal is an atomic formula or the negatio of an atomic formula.

Definition 6.26: A formula F is in conjunctive normal form (CNF) if it is a conjunction of disjunctions of liters, i.e., if it is of the form $F = (L_{11} \vee ... \vee L_{1m_1}) \wedge ... \wedge (F_{n1} \vee ... \vee L_{nm_n})$ for some literals L_{ij}.

Definition 6.27: A formula F is in disjunctive normal form (DNF) if it is a disjunction of conjunctions of literals, i.e., if it is of the form $F = (L_{11} \wedge ... \wedge L_{1m_1}) \vee ... \vee (L_{n1} \wedge ... \wedge L_{nm_n})$.

Theorem 6.5: Every formula is equivalent to a formula in CNF to a formula in DNF.

Some Derivation Rules

Not a calculus, just some rules. All equivalences (Lemma 6.1 and more) can be stated as rules: $\neg\neg F \vdash F$, $F \wedge G \vdash G \wedge F$, $\neg(F \vee G) \vdash \neg F \wedge \neg G$. Furthermore:

- $F \wedge G \vdash F$ and $F \wedge G \vdash G$
- $\{F, G\} \vdash F \wedge G$
- $F \vdash F \vee G$ and $F \vdash G \vee F$
- $\{F, F \rightarrow G\} \vdash G$
- $\{F \vee G, F \rightarrow H, G \rightarrow H\} \vdash H$

Also: $\vdash F \vee \neg F$ and $\vdash \neg(F \leftrightarrow \neg F)$.

The Resolution Calculus for Propositional Logic

Used to prove unsatisfiability of a set M of formulas. Also allows proofs of tautologies and logical consequences.

All formulas must be given in CNF. Work with equivalent objects:

Definition 6.28: A clause is a set of literals.

Definition 6.29: The set of clauses associated to a formula $F = (L_{11} \vee ... \vee L_{1m_1}) \wedge ... \wedge (L_{n1} \vee ... \vee L_{nm_n})$ in CNF, denoted as $\mathcal{K}(F)$ is the set $\mathcal{K}(F) \overset{\text{def}}{=} \{\{L_{11}, ..., L_{1m_1}\}, ..., \{L_{n1}, ..., L_{nm_n}\}\}$. The set of clauses associated with a set $M = \{F_1, ..., F_k\}$ of formulas is the union of their clauses: $\mathcal{K}(M) \overset{\text{def}}{=} \bigcup_{i=1}^{k} \mathcal{K}(F_i)$.

Clause is satisfied by an interpretation if some literal evaluates to true. Clauses stand for the disjunction of their literals. $\mathcal{K}(M)$ is satisfied by an interpretation if every clause in $\mathcal{K}(M)$ is satisfied by it. Sets of clauses stand for the conjunction of their clauses.

Empty clause unsatisfiable. Empty set of clauses is tautology.

Definition 6.30: A clause K is resolvent of clauses K_1 and K_2 if there is a literal L such that $L \in K_1, \neg L \in K_2$, and $K = (K_1 \backslash \{L\}) \cup (K_2 \backslash \{\neg L\})$.

One can not perform two steps at once!

The resolution rule: $\{K_1, K_2\} \vdash_{\text{res}} K$. The resolution calculus: Res = {res}.

Lemma 6.6: Resolution calculus is sound: $\mathcal{K} \vdash_{\text{Res}} K \Rightarrow \mathcal{K} \models K$.

Theorem 6.7: A set M of formulas is unsatisfiable if and only if $\mathcal{K}(M) \vdash_{\text{Res}} \emptyset$.

predicate logic

Syntax

Definition 6.31:

- variable symbol is of the form x_i with $i \in \mathbb{N}$
- function symbol is of the form $f_i^{(k)}$ with $i, k \in \mathbb{N}$, where k denotes the number of argumets of the function. $k = 0$: Constant.
- 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.
- term is defined inductively: A variable is a term, and if $t_1, ..., t_k$ are terms, then $f_i^{(k)}(t_1, ..., t_k)$ is a term. $k = 0$: no parentheses
- formula is defined inductively:
  - For any i and k, if $t_1, ..., t_k$ are terms, then $P_i^{(k)}(t_1, ..., t_k)$ is a (atomic) formula.
  - If F and G are formulas, then $\neg F$, $(F \wedge G)$, $(F \vee G)$ are formulas.
  - If F is a formula, then, for any i, $\forall x_i F$ and $\exists x_i F$ are formulas.

$\forall$ is the universal quantifier. $\exists$ is the existential quantifier. One can depict such a formula as a tree. For function symbols (f, g, h) number of arguments usually implicit. For predicate symbols (P, Q, R) number of arguments usually implicit. $x, y, z, u, v, w, k, m, n$ as variable instead of x_i.

Free Variables and Variable Substitution

Definition 6.32: Every occurrence of a variable in a formula is either bound or free. If x occurs in a s(sub-)formula of the form $\forall xG$ or $\exists xG$, then it is bound - otherwise free. Formula F is called closed if it contains no free variables.

Definition 6.33: Formula F, variable x, term t: $F[x/t]$ denotes the formula obtained from F by substituting every free occurrence of x by t.

Semantics

In predicate logic, the free symbols fo a formula are all predicate symbols, all function symbols, and al occurrences of free variables.

Definition 6.34: An interpretation or structure is a tuple $\mathcal{A} = (U, \phi, \psi, \zeta)$, where

- U is a non-empty universe.
- ϕ is a function assigning to each function symbol (in a certain subset of all function symbols) a function, where for a k-ary function symbol f, $\phi(f)$ is a function $U^k \rightarrow U$.
- ψ is a function assigning to each predicate symbol (in a certain subset of all predicate symbols) a function, where for a k-ary predicate symbol P, $\psi(P)$ is a function $U^k \rightarrow \{0, 1\}$. (implies definition 2.10)
- ζ is a function assigning to each variable symbol (in a certain subset of all variable symbols) a value in U.

Notational convenience: $f^{\mathcal{A}}$ instead of $\phi(f)$, $P^{\mathcal{A}}$ instead of $\psi(P)$, $x^{\mathcal{A}}$ instead of $\zeta(x)$, $U^{\mathcal{A}}$ insetad of U.

Definition 6.35: An interpretation (structure) $\mathcal{A}$ is suitable for a formula F if it defines all function symbols, predicate symbols, and freely occuring variables of F.

Definition 6.36: For an interpretation $\mathcal{A} = (U, \phi, \psi, \zeta)$, 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:
  - If t is a variable ($t = x_i$): $\mathcal{A}(t) = \zeta(x_i)$.
  - If t is of the form $f(t_1, ..., t_k)$ for term $t_1, ..., t_k$ and a k-ary function symbol f, then $\mathcal{A}(t) = \phi(f)(\mathcal{A}(t_1), ..., \mathcal{A}(t_k))$.
- Teh truth value of a formula F is defined recursively by Def. 6.16 and:
  - If F is of the form $F = P(t_1, ..., t_k)$ for terms $t_1, ..., t_k$ and a k-ary predicate symbol P, then $\mathcal{A}(F) = \psi(P)(\mathcal{A}(t_1), ..., \mathcal{A}(t_k))$.
  - If F is of the form $\forall xG$ or $\exists xG$, then $\mathcal{A}_{[x\rightarrow u]}$ for some $u \in U$ be the same structure as $\mathcal{A}$ except that $\zeta(x)$ is overwritten by u:

    $\mathcal{A}(\forall xG) = \begin{cases} 1, \mathcal{A}_{[x\rightarrow U]}(G)=1 \text{ for all } u \in U \\ 0, \text{else} \end{cases}$

    $\mathcal{A}(\exists xG) = \begin{cases} 1, \mathcal{A}_{[x\rightarrow U]}(G)=1 \text{ for some } u \in U \\ 0, \text{else} \end{cases}$

This defines $\sigma(F, \mathcal{A})$ of Def. 6.8.

Predicate Logic with Equality

= is usually not usually allowed. But one can extend the syntax and semantics of predicate logic to include the equality symbol "=".

Some Basic Equivalences Involving Quantifiers

Lemma 6.8: For any formulas F, G, H (x not free in H):

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

Useful rules (2.4.8):

- $\exists x(P(x) \wedge Q(x)) \models \exists xP(x) \wedge \exists xQ(x)$
- $\exists y\forall xP(x, y) \models \forall x\exists yP(x, y)$

Lemma 6.9: If one replaces a sub-formula G of a formula F by an equivalent (to G) formula H, then the resulting formula is equivalent to F.

Substitution of Bound Variables

Lemma 6.10: For a formula G in which y does not occur, we have $\forall xG \equiv \forall yG[x/y]$ and $\exists xG \equiv \exists yG[x/y]$.

Definition 6.37: A formula in which no variable occurs both as a bound and as a free variable and in which all variables appearing after the quantifiers are distinct is said to be in rectified form.

And formula can be expressed in rectified form.

Universal Instantiation

Lemma 6.11: For any formula F and any term t we have $\forall xF \models F[x/t]$.

Normal Forms

Definition 6.38: A formula of the form $Q_1x_1\ Q_2x_2\ ...\ Q_nx_n\ G$ where Q_i are arbitary quantifiers and G is a formula free of quantifiers, is said to be in prenex form.

Theorem 6.12: For every formula there is an equivalent formula in prenex form.

For Skolem normal form one also removes all $\exists$ quantifiers. Then, only equivalence regarding satisfiability is guaranteed.

An Example Theorem and its Interpretations

Theorem 6.13: $\neg\exists x\forall y(P(y, x) \leftrightarrow \neg P(y, y))$.

Corollary 6.14: There exists no set that contains all sets S that do not contain themselves. (Russel's paradox.)

Barber paradox

Corollary 6.15: The set $\{0, 1\}^\infty$ is uncountable.

Corollary 6.16: Tehre are uncomputable function $\mathbb{N} \rightarrow \{0, 1\}$.

Corollary 6.17: The function $\mathbb{N} \rightarrow \{0, 1\}$ assigning to each $y \in \mathbb{N}$ the complement of what programm y outputs on input y, is uncomputable.

beyond predicate logic

Predicate logic is naturally limited. For instance, $\forall x\exists y$ corresponds to the existence of a function f for all x. But in predicatelogic we can not write $\exists f$.

Alternatively, in $\forall w\forall x\exists y\exists zP(w, x, y, z)$, y, z depend on w, x. In predicate logic it can not be expressed that y may only depend on w and z may only depend on x.

Addition

inverses mod m

mod 3: 2:2 mod 4: 3:3 mod 5: 2:3,4:4 mod 6: 5:5 mod 7: 2:4,3:5,6:6 mod 8: 3:3,5:5,7:7 mod 9: 2:5,3:3,4:7,8:8 mod 10: 3:7,9:9 mod 11: 2:6,3:4,5:9,7:8,10:10 mod 12: 5:5,7:7,11:11 mod 13: 2:7,3:9,4:10,5:8,6:11,12:12 mod 14: 3:5,9:11,13:13 mod 15: 2:8,4:4,7:13,11:11,14:14 mod 16: 3:11,5:13,7:7,9:9,16:16 mod 17: 2:9,3:6,4:13,5:7,8:15,10:12,11:14,16:16 mod 18: 5:11,7:13,17:17 mod 19: 2:10,3:13,4:5,6:16,7:11,8:12,9:17,14:15,18:18 mod 20: 3:7,9:9,11:11,13:17:19:19

irreducible polynomials

$GF(2)[x]$: 10, 11, 111, 1101, 10011, 11001, 11111, 100101, 101001, 101111, 110111, 111011, 111101, 1000011, 1001001, 1010111, 1011011, 1100001, 1100111, 1101101, 1110011, 1110101 $GF(3)[x]$: 10, 11, 12, 101, 112, 122, 1021, 1022, 1102, 1112, 1121, 1201, 1211, 1222, 10012, 10022, 10102, 10111, 10121, 10202, 11002, 11021, 111001, 11111, 11122, 11222, 12002, 12011, 12112, 12121, 12212

$GF(4)[x]$: 10, 11, 12, 13, 112, 113, 121, 122, 131, 133, 1002, 1003, 1011, 1021, 1031, 1101, 1112, 1113, 1123, 1132, 1201, 1213, 1222, 1232, 1233, 1301, 1312, 1322, 1323, 1333 $GF(5)[x]$: 10, 11, 12, 13, 14, 102, 103, 111, 112, 123, 124, 133, 134, 141, 142, 1011, 1014, 1021, 1024, 1032, 1033, 1042, 1043, 1101, 1102, 1113, 1114, 1131, 1134, 1141, 1143, 1201, 1203, 1213, 1214, 1222, 1223, 1242, 1244, 1302, 1304, 1311, 1312, 1322, 1323, 1341, 1343, 1403, 1404, 1411, 1412, 1431, 1434, 1442, 1444

$GF(7)[x]$: 10, 11, 12, 13, 14, 15, 16, 101, 102, 104, 113, 114, 116, 122, 123, 125, 131, 135, 136, 141, 145, 146, 152, 153, 155, 163, 164, 166

ad