Mathematical Induction

Proof (of the truth) of proposition $P(n)$ for all natural numbers n with $n \geq m$:

- basis: proof of $P(m)$
- induction hypothesis (IH): suppose that $P(k)$ is true for all k with $m \leq k \leq n$
- inductive step: proof of $P(n+1)$ using the induction hypothesis

- Weak induction: Induction hypothesis only supposes that $P(k)$ is true for $k = n$
- Strong induction: Induction hypothesis supposes that $P(k)$ is true for all $k \in \mathbb{N}_0$ with $m \leq k \leq n$
  - also: complete induction

Inductive Definition

A set M can be defined inductively by specifying

- basic elements that are contained in M
- construction rules of the form "Given some elements of M, another element of M can be constructed like this."

Structural Induction

Proof of statement for all elements of an inductively defined set

- basis: proof of the statement for the basic elements
- induction hypothesis (IH): suppose that the statement is true for some elements M
- inductive step: proof of the statement for elements constructed by applying a construction rule to M (one inductive step for each construction rule)

Definition (Leaves of a Binary Tree)

The number of leaves of a binary tree B, written $leaves(B)$, is defined as follows:

$$leaves(\square) = 1$$
$$leaves(\langle L, \bigcirc, R\rangle) = leaves(L) + leaves(R)$$

Definition (Inner Nodes of a Binary Tree)

The number of inner nodes of a binary tree B, written $inner(B)$, is defined as follows:

$$inner(\square) = 0$$
$$inner(\langle L, \bigcirc, R\rangle) = inner(L) + inner(R) + 1$$

Definition (Height of a Binary Tree)

The height of a binary tree B, written $height(B)$, is defined as follows:

$$height(\square) = 0$$
$$height(\langle L, \bigcirc, R\rangle) = \max\{height(L), height(R)\} + 1$$

Prove by structural induction:

Theorem

For all binary trees B: $leaves(B) \leq 2^{height(B)}$.

Sets

A set is an unordered collection of distinct objects.

- Specification of sets
  - explicit, listing all elements, e. g. $A = \{1, 2, 3\}$
  - implicit with set-builder notation, specifying a property characterizing all elements, e. g. $A = \{x \mid x \in \mathbb{N}_0 \text{ and } 1 \leq x \leq 3\}$, $B = \{n^2 \mid n \in \mathbb{N}_0\}$
  - implicit, as a sequence with dots, e. g. $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$
  - implicit with an inductive definition

- $\overline{A \cup B} = \overline{A} \cap \overline{B}$ and
- $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

Definition (Equinumerous Sets)

Two sets A and B have the same cardinality ($|A| = |B|$) if there exists a bijection from A to B.

Such sets are called equinumerous.

A set A is countably infinite if $|A| = |\mathbb{N}_0|$.

A set A is countable if $|A| \leq |\mathbb{N}_0|$.

Theorem (Cantor's Theorem)

For every set S it holds that $|S| < |\mathcal{P}(S)|$.

- Consider an arbitrary finite set of symbols (an alphabet) Σ.
- You can think of $\Sigma = \{0, 1\}$ as internally computers operate on binary representation.
- Let S be the set of all finite strings made from symbols in Σ.
- There are at most $|S|$ computer programs with this alphabet.
- There are at least $|\mathcal{P}(S)|$ problems with this alphabet.
  - every subset of S corresponds to a separate decision problem
- By Cantor's theorem $|S| < |\mathcal{P}(S)|$, so there are more problems than programs.

Definition (Relation)

Let $S_1, \dots, S_n$ be sets.

A relation over $S_1, \dots, S_n$ is a set $R \subseteq S_1 \times \dots \times S_n$.

The arity of R is n.

- A relation of arity n is a set of n-tuples.
- The set contains the tuples for which the informal property is true.

- reflexive: $(x, x) \in R$ for all $x \in S$
- irreflexive: $(x, x) \notin R$ for all $x \in S$
- symmetric: $(x, y) \in R$ iff $(y, x) \in R$
- asymmetric: if $(x, y) \in R$ then $(y, x) \notin R$
- antisymmetric: if $(x, y) \in R$ then $(y, x) \notin R$ or $x = y$
- transitive: if $(x, y) \in R$ and $(y, z) \in R$ then $(x, z) \in R$

Irreflexivity => not reflexiv

contrapos: reflexiv => not irreflexiv

assymetry => irreflexive
assym => antisym

Definition (Partition)

A partition of a set S is a set $P \subseteq \mathcal{P}(S)$ such that

- $X \neq \emptyset$ for all $X \in P$, not contain empty set
- $\bigcup_{X \in P} X = S$, and every element of S must be in at least 1 subset of partition
- $X \cap Y = \emptyset$ for all $X, Y \in P$ with $X \neq Y$, every element in at most 1 subset of partition

The elements of P are called the blocks of the partition.

For $e \in S$ we denote by $[e]_P$ the block $X \in P$ such that $e \in X$.

Definition (Relation induced by a partition)

Let S be a set and P be a partition of S.

The relation $\sim_P$ induced by P is the binary relation over S with

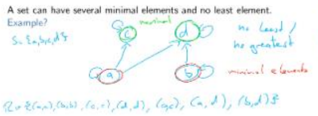
$$x \sim_P y \text{ iff } [x]_P = [y]_P.$$

- A relation is an equivalence relation if it is reflexive, symmetric and transitive.

- A partial order is reflexive, antisymmetric and transitive.
- With a total order $\preceq$ over S there are no elements $x, y \in S$ with $x \not\preceq y$ and $y \not\preceq x$. no incomparable elements
- If x is the greatest element of a set S, it is greater than every element: for all $y \in S$ it holds that $y \preceq x$.
- If x is a maximal element of set S then it is not smaller than any other element y: there is no $y \in S$ with $x \preceq y$ and $x \neq y$.

A set can have several minimal elements and no least element.
Example?

minimal element existiert =/=> least el. existiert



since c and d not comparable

Definition (Total relation)

A binary relation R over set S is total (or connex) if for all $x, y \in S$ at least one of xRy or yRx is true.

Definition (Total order)

A binary relation is a total order if it is total and a partial order.

Definition (Strict order)

A binary relation $\prec$ over set S is a strict order if $\prec$ is irreflexive, asymmetric and transitive.

partial are reflexive and strict are irreflexive

- As partial orders, a strict order does not automatically allow us to rank arbitrary two objects against each other.
- Example 1 (personal preferences):
  - "Pasta tastes better than potato."
  - "Rice tastes better than bread."
  - "Bread tastes better than potato."
  - "Rice tastes better than potato."
  - This definition of "tastes better than" is a strict order.
  - No ranking of pasta against rice or of pasta against bread.

- Example 2: $\subset$ relation for sets
- It doesn't work to simply require that the strict order is total. Why? because we cannot compare two objects that are the same

Definition (Trichotomy)

A binary relation R over set S is trichotomous if for all $x, y \in S$ exactly one of xRy, yRx or $x = y$ is true.

Definition (Strict total order)

A binary relation $\prec$ over S is a strict total order if $\prec$ is trichotomous and a strict order.

Definition (Least/greatest/minimal/maximal element of a set)

Let $\prec$ be a strict order over S.

An element $x \in S$ is the least element of S if for all $y \in S$ where $y \neq x$ it holds that $x \prec y$.

It is the greatest element of S if for all $y \in S$ where $y \neq x$, $y \prec x$.

Element $x \in S$ is a minimal element of S if there is no $y \in S$ with $y \prec x$.

It is a maximal element of S if there is no $y \in S$ with $x \prec y$.

$S'_1, \dots, S'_n$. Then $R \cup R'$ is a relation over $S_1 \cup S'_1, \dots, S_n \cup S'_n$.

- Let R and R' be relations over n sets. Then $R \cap R'$ is a relation.
  Over which sets? (x1,...,xn) in R intersection R'
  With the standard relations $\leq, =$ and $\geq$ for $\mathbb{N}_0$, relation $=$ corresponds to the intersection of $\leq$ and $\geq$.
- If R is a relation over $S_1, \dots, S_n$ then so is the complementary relation $\bar{R} = (S_1 \times \dots \times S_n) \setminus R$.

$R = \{(0,1), (1,0), (1,1)\}$ over $\{0,1\}$ then $\bar{R} = \{(0,0)\}$ | over $\{0,1,2\}$ $(0,2) \in \bar{R}$

The inverse relation of R is the relation $R^{-1} \subseteq B \times A$ given by $R^{-1} = \{(b, a) \mid (a, b) \in R\}$.

Definition (Composition of relations)

Let R_1 be a relation over A and B and R_2 be a relation over B and C.

The composition of R_1 and R_2 is the relation $R_2 \circ R_1$ with:

$$R_2 \circ R_1 = \{(a, c) \mid \text{there is a } b \in B \text{ with } (a, b) \in R_1 \text{ and } (b, c) \in R_2\}$$

Theorem (Associativity of composition)

Let $S_1, \dots, S_4$ be sets and R_1, R_2, R_3 relations with $R_i \subseteq S_i \times S_{i+1}$. Then

$$R_3 \circ (R_2 \circ R_1) = (R_3 \circ R_2) \circ R_1.$$

Definition (Transitive closure)

The transitive closure R^* of a relation R over set S is the smallest relation over S that is transitive and has R as a subset.

The transitive closure always exists. Why?

because SxS is trans and contains R
R* is the intersection of all R in SxS

Define the i-th power of a homogeneous relation R as

$$R^1 = R \quad \text{if } i = 1 \text{ and}$$
$$R^i = R \circ R^{i-1} \quad \text{for } i > 1$$

Theorem

Let R be a relation over set S. Then $R^* = \bigcup_{i=1}^{\infty} R^i$.

A binary relation R over sets A and B is functional if for every $a \in A$ there is at most one $b \in B$ with $(a, b) \in R$.

Definition (Partial function)

A partial function f from set A to set B (written $f : A \rightharpoonup B$) is given by a functional relation G over A and B.

Relation G is called the graph of f.

Definition (domain of definition, codomain, image)

Let $f : A \rightharpoonup B$ be a partial function.

Set A is called the domain of f, set B is its codomain

The domain of definition of f is the set $dom(f) = \{x \in A \mid \text{there is a } y \in B \text{ with } f(x) = y\}$.

The image (or range) of f is the set $img(f) = \{y \mid \text{there is an } x \in A \text{ with } f(x) = y\}$.

Definition (Total function)

A (total) function $f : A \to B$ from set A to set B is a partial function from A to B such that $f(x)$ is defined for all $x \in A$.

Definition (Composition of partial functions)

Let $f : A \rightharpoonup B$ and $g : B \rightharpoonup C$ be partial functions.

The composition of f and g is $g \circ f : A \rightharpoonup C$ with

$$(g \circ f)(x) = \begin{cases} g(f(x)) & \text{if } f \text{ is defined for } x \text{ and } g \text{ is defined for } f(x) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Corresponds to relation composition of the graphs.

Definition (Permutation)

Let S be a set. A bijection $\pi : S \to S$ is called a permutation of S.

One-line notation only lists the second row

A permutation is cyclic if it has a single k-cycle with $k > 1$.

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 4 & 1 & 5 \end{pmatrix}^{-1} = \begin{pmatrix} 3 & 2 & 4 & 1 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix} \quad (\pi\sigma)^{-1} = \sigma^{-1}\pi^{-1}$$

$x \cdot b = a$ has exactly one solution x in S, namely $x = a \cdot b^{-1}$.

We call $a \cdot b^{-1}$ the right-quotient of a by b and also write it as a/b.

$b \cdot x = a$ has exactly one solution x in S, namely $x = b^{-1} \cdot a$.

We call $b^{-1} \cdot a$ the left-quotient of a by b and also write it as $b \setminus a$.

Definition

A generating set of a group $G = (S, \circ)$ is a set $S' \subseteq S$ such that every $e \in S$ can be expressed as a combination (under $\circ$) of finitely many elements of S' and their inverses.

Empty product is identity by definition, so no need to have it in S'.

- For $n \geq 2$, S_n is generated by $\{(i \quad i+1) \mid i \in \{1, \dots, n-1\}\}$.
- For $n > 2$, S_n is generated by $\{(1 \quad 2), (1 \quad \dots \quad n)\}$.

Definition (Permutation Group)

A permutation group is a group $G = (S, \cdot)$, where S is a set of permutations of some set M and $\cdot$ is the composition of permutations in S.

Every permutation group is a subgroup of a symmetric group and every such subgroup is a permutation group.

Divisibility | over $\mathbb{N}_0$ is a partial order.

$a = qb + r$ and $0 \leq r < |b|$

- $a = 18$, $b = -5$ 18 = -3 * -5 + 3

Definition (Congruence modulo n)

For integer $n > 1$, two integers a and b are called congruent modulo n if $n \mid a - b$.

We write this as $a \equiv b \pmod{n}$.

$0 \equiv 5 \pmod 5$

Theorem

For integers a and b and integer $n > 1$ it holds that $a \equiv b \pmod{n}$ iff there are $q, q', r \in \mathbb{Z}$ with

$$a = qn + r$$
$$b = q'n + r.$$

Theorem

Congruence modulo n is an equivalence relation.

Theorem

Congruence modulo n is compatible with addition, subtraction, multiplication, translation, scaling and exponentiation, i. e. if $a \equiv b \pmod n$ and $a' \equiv b' \pmod n$ then

- $a + a' \equiv b + b' \pmod n$,
- $a - a' \equiv b - b' \pmod n$,
- $aa' \equiv bb' \pmod n$,
- $a + k \equiv b + k \pmod n$ for all $k \in \mathbb{Z}$,
- $ak \equiv bk \pmod n$ for all $k \in \mathbb{Z}$, and
- $a^k \equiv b^k \pmod n$ for all $k \in \mathbb{N}_0$.

Theorem (Fermat's Little Theorem)

If $a \in \mathbb{Z}$ is not a multiple of prime number p then $a^{p-1} \equiv 1 \pmod p$.

Find the remainder when dividing 4^{100000} by 67.

67 is prime and 4 is not a multiple of 67, so we can use the theorem.

By the theorem, $4^{66} \equiv 1 \pmod{67}$. How does this help?

Raise both sides to a higher power.
$100000/66 = 1515.\overline{15}$ $\rightarrow$ use 1515

$(4^{66})^{1515} \equiv 1^{1515} \pmod{67}$ iff
$4^{99990} \equiv 1 \pmod{67}$ iff
$4^{10}4^{99990} \equiv 4^{10} \pmod{67}$ iff (calculator)
$4^{100000} \equiv 26 \pmod{67}$

Definition (Graph)

A graph (also: undirected graph) is a pair $G = (V, E)$, where

- V is a finite set called the set of vertices, and
- $E \subseteq \{\{u, v\} \subseteq V \mid u \neq v\}$ is called the set of edges.

Definition (Directed Graph)

A directed graph (also: digraph) is a pair $G = (N, A)$, where

- N is a finite set called the set of nodes, and
- $A \subseteq N \times N$ is called the set of arcs.

- A directed graph (N, A) is essentially identical to (= contains the same information as) an arbitrary relation R_A over the finite set N: $u R_A v$ iff $(u, v) \in A$
- A graph (V, E) is essentially identical to an irreflexive symmetric relation R_E over the finite set V: $u R_E v$ iff $\{u, v\} \in E$

Definition (Graph Terminology)

Let (V, E) be a graph.

- u and v are the endpoints of the edge $\{u, v\} \in E$
- u and v are incident to the edge $\{u, v\} \in E$
- u and v are adjacent if $\{u, v\} \in E$
- the vertices adjacent with $v \in V$ are its neighbours neigh(v): neigh$(v) = \{w \in V \mid \{v, w\} \in E\}$
- the number of neighbours of $v \in V$ is its degree deg(v): deg$(v) = |$neigh$(v)|$

Definition (Directed Graph Terminology)

Let (N, A) be a directed graph.

- u is the tail and v is the head of the arc $(u, v) \in A$; we say (u, v) is an arc from u to v
- u and v are incident to the arc $(u, v) \in A$
- u is a predecessor of v and v is a successor of u if $(u, v) \in A$
- the predecessors and successor of v are written as pred$(v) = \{u \in N \mid (u, v) \in A\}$ and succ$(v) = \{w \in N \mid (v, w) \in A\}$
- the number of predecessors/successors of $v \in N$ is its indegree/outdegree: indeg$(v) = |$pred$(v)|$, outdeg$(v) = |$succ$(v)|$

Definition (undirected graph induced by a directed graph)

Let $G = (N, A)$ be a directed graph.
The (undirected) graph induced by G is the graph (N, E) with $E = \{\{u, v\} \mid (u, v) \in A, u \neq v\}$ "forgetting the orientation of the arcs"

- Why require $u \neq v$? no self-loops
- If $|N| = n$ and $|A| = m$, how many vertices and edges does the induced graph have? n, <= m, all self-loops disappear
- How does the answer change if G has no self-loops? then the induced graph again has <= m and >= m/2 edges, 2 different arcs between the same two points become the same edge in the undirected graph

Lemma (degree lemma for directed graphs)

Let (N, A) be a directed graph.
Then $\sum_{v \in N} \text{indeg}(v) = \sum_{v \in N} \text{outdeg}(v) = |A|$.

Intuitively: every arc contributes 1 to the indegree of one node and 1 to the outdegree of one node.

Lemma (degree lemma for undirected graphs)

Let (V, E) be a graph.
Then $\sum_{v \in V} \deg(v) = 2|E|$.

Intuitively: every edge contributes 1 to the degree of two vertices.

Corollary

Every graph has an even number of vertices with odd degree.

because sum of degrees = 2 |E| which is even, hence we need an even number of odd degree terms to arrive at an even total sum

Proof of degree lemma for directed graphs.

$$\begin{aligned}
\sum_{v \in N} \text{indeg}(v) &= \sum_{v \in N} |\text{pred}(v)| \\
&= \sum_{v \in N} |\{u \mid u \in N, (u, v) \in A\}| \quad \text{(all u s.t. (u,v) is an arc)} \\
&= \sum_{v \in N} |\{(u, v) \mid u \in N, (u, v) \in A\}| \\
&= \left|\bigcup_{v \in N} \{(u, v) \mid u \in N, (u, v) \in A\}\right| \\
&= |\{(u, v) \mid u \in N, v \in N, (u, v) \in A\}| \\
&= |A|.
\end{aligned}$$

$\sum_{v \in N} \text{outdeg}(v) = |A|$ is analogous.

We omit the proof for undirected graphs, which can be conducted similarly.

One possible proof strategy that reuses the result we proved:

- Define directed graph (V, A) from the graph (V, E) by orienting each edge into an arc arbitrarily.
- Observe $\deg(v) = \text{indeg}(v) + \text{outdeg}(v)$, where deg refers to the graph and indeg/outdeg to the directed graph.
- Use the degree lemma for directed graphs:
$\sum_{v \in V} \deg(v) = \sum_{v \in V}(\text{indeg}(v) + \text{outdeg}(v)) = \sum_{v \in V} \text{indeg}(v) + \sum_{v \in V} \text{outdeg}(v) = |A| + |A| = 2|A| = 2|E|$

Definition (Walk)

A walk of length n in a graph (V, E) is a tuple $\langle v_0, v_1, \dots, v_n \rangle \in V^{n+1}$ s.t. $\{v_i, v_{i+1}\} \in E$ for all $0 \leq i < n$.

A walk of length n in a digraph (N, A) is a tuple $\langle v_0, v_1, \dots, v_n \rangle \in N^{n+1}$ s.t. $(v_i, v_{i+1}) \in A$ for all $0 \leq i < n$.

- The length of the walk (n) does not equal the length of the tuple (n+1)!
- The case $n = 0$ is allowed. single vertex is always a walk
- Vertices may repeat along a walk.

Definition

Let $\pi = \langle v_0, \dots, v_n \rangle$ be a walk in a graph or digraph G.

- We say π is a walk from v_0 to v_n.
- A walk with $v_i \neq v_j$ for all $0 \leq i < j \leq n$ is called a path.
- A walk of length 0 is called an empty walk/path.
- A walk with $v_0 = v_n$ is called a tour.
- A tour with $n \geq 1$ (digraphs) or $n \geq 3$ (graphs) and $v_i \neq v_j$ for all $1 \leq i < j \leq n$ is called a cycle.

Definition (successor and reachability)

Let G be a graph (digraph).
The successor relation S_G and reachability relation R_G are relations over the vertices/nodes of G defined as follows:

- $(u, v) \in S_G$ iff $\{u, v\}$ is an edge ((u, v) is an arc) of G
- $(u, v) \in R_G$ iff there exists a walk from u to v

If $(u, v) \in R_G$, we say that v is reachable from u.

Recall the n-fold composition R^n of a relation R over set S:

- $R^1 = R$
- $R^{n+1} = R \circ R^n$

also: $R^0 = \{(x, x) \mid x \in S\}$ (0-fold composition is identity relation)

Theorem

Let G be a graph or digraph. Then:
$(u, v) \in S_G^n$ iff there exists a walk of length n from u to v.
n-th power of the successor relation

Corollary

Let G be a graph or digraph. Then $R_G = \bigcup_{n=0}^{\infty} S_G^n$.

In other words, the reachability relation is the reflexive and transitive closure of the successor relation.

Theorem

Let G be a graph or digraph.
There exists a path from u to v iff there exists a walk from u to v.

Proof.
($\Rightarrow$): obvious because paths are special cases of walks

($\Leftarrow$): Proof by contradiction. Assume there exist u, v such that there exists a walk from u to v, but no path. Let $\pi = \langle w_0, \dots, w_n \rangle$ be such a counterexample walk of minimal length.
Because π is not a path, some vertex/node must repeat.
Select i and j with $i < j$ and $w_i = w_j$.
Then $\pi' = \langle w_0, \dots, w_i, w_{j+1}, \dots, w_n \rangle$ also is a walk from u to v.
If π' is a path, we have a contradiction.
If not, it is a shorter counterexample: also a contradiction. □

Theorem

For every graph G, the reachability relation R_G is an equivalence relation.

In directed graphs, this result does not hold (easy to see).

Definition (connected components, connected)

In a graph G, the equivalence classes of the reachability relation of G are called the connected components of G.

A graph is called connected if it has at most 1 connected component.

Remark: The graph $(\emptyset, \emptyset)$ has 0 connected components.
It is the only such graph.

Definition (weakly connected components, weakly connected)

In a digraph G, the equivalence classes of the reachability relation of the induced graph of G are called the weakly connected components of G.

A digraph is called weakly connected if it has at most 1 weakly connected component.

Definition (mutually reachable)

Let G be a graph or digraph.
Vertices/nodes u and v in G are called mutually reachable if v is reachable from u and u is reachable from v.
We write M_G for the mutual reachability relation of G

Theorem

For every digraph G, the mutual reachability relation M_G is an equivalence relation.

Definition (strongly connected components, strongly connected)

In a digraph G, the equivalence classes of the mutual reachability relation are called the strongly connected components of G.

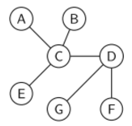
A digraph is called strongly connected if it has at most 1 strongly connected component.

Definition (acyclic, forest, DAG)

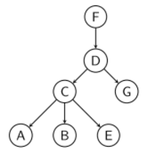
A graph or digraph G is called acyclic if there exists no cycle in G.
An acyclic graph is also called a forest.
An acyclic digraph is also called a DAG (directed acyclic graph).

Definition (tree)

A connected forest is called a tree.



tree graph



rooted tree with root F

Theorem

Let $G = (V, E)$ be a graph.
Then G is a tree iff there exists exactly one path from any vertex $u \in V$ to any vertex $v \in V$.

Definition

Let $G = (V, E)$ be a tree.
A leaf of G is a vertex $v \in V$ with $\deg(v) = 1$.

Theorem

Let $G = (V, E)$ be a tree with $|V| \geq 2$.
Then G has at least two leaves.

Theorem

Let $G = (V, E)$ be a tree with $V \neq \emptyset$.
Then $|E| = |V| - 1$.

Theorem

Let $G = (V, E)$ be a forest.
Let C be the set of connected components of G.
Then $|E| = |V| - |C|$.

This result generalizes the previous one.

Theorem

Let $G = (V, E)$ be a graph with $V \neq \emptyset$.
The following statements are equivalent:

1. *G is a tree.*
2. *G is acyclic and connected.*
3. *G is acyclic and $|E| = |V| - 1$.*
4. *G is connected and $|E| = |V| - 1$.*
5. *For all $u, v \in V$ there exists exactly one path from u to v.*

Definition (subgraph)

A subgraph of a graph (V, E) is a graph (V', E') with $V' \subseteq V$ and $E' \subseteq E$.

A subgraph of a digraph (N, A) is a digraph (N', A') with $N' \subseteq N$ and $A' \subseteq A$.

Question: Can we choose V' and E' arbitrarily?
no: if I remove some of the vertices and keep edges that were attached...

The subgraph relationship defines a partial order on graphs (and on digraphs).

Definition (induced subgraph)

Let $G = (V, E)$ be a graph, and let $V' \subseteq V$.
The subgraph of G induced by V' is the graph (V', E') with $E' = \{\{u, v\} \in E \mid u, v \in V'\}$.
We say that G' is an induced subgraph of $G = (V, E)$ if G' is the subgraph of G induced by V' for any set of vertices $V' \subseteq V$.

completely analogous

Definition (induced subgraph)

Let $G = (N, A)$ be a digraph, and let $N' \subseteq N$.
The subgraph of G induced by N' is the digraph (N', A') with $A' = \{(u, v) \in A \mid u, v \in N'\}$.
We say that G' is an induced subgraph of $G = (N, A)$ if G' is the subgraph of G induced by N' for any set of nodes $N' \subseteq N$.

- They are the largest (in terms of the set of edges) subgraphs with any given set of vertices.
- A typical example are subgraphs induced by the connected components of a graph.
- The subgraphs induced by the connected components of a forest are trees.

- How many subgraphs does a graph (V, E) have?
- How many induced subgraph does a graph (V, E) have?

For the second question, the answer is $2^{|V|}$.
The first question is in general not easy to answer because vertices and edges of a subgraph cannot be chosen independently.

Example (subgraphs of a complete graph)

A complete graph with n vertices (i.e., with all possible $\binom{n}{2}$ edges) has $\sum_{k=0}^{n} \binom{n}{k} 2^{\binom{k}{2}}$ subgraphs. (Why?)
for $n = 10$: 1024 induced subgraphs, 35883905263781 subgraphs

Definition (Isomorphism)

Let $G = (V, E)$ and $G' = (V', E')$ be graphs.

An isomorphism from G to G' is a bijective function $\sigma : V \to V'$ such that for all $u, v \in V$:

$$\{u, v\} \in E \quad \text{iff} \quad \{\sigma(u), \sigma(v)\} \in E'.$$

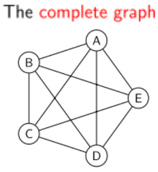
If there exists an isomorphism from G to G', we say that they are isomorphic, in symbols $G \cong G'$.

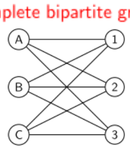
graph invariant,

- examples: number of vertices, number of edges, maximum/minimum degree, sorted sequence of all degrees, number of connected components
- Having a cycle of a given length is an invariant.
- An isomorphism σ between a graph G and itself is called an automorphism or symmetry of G.

rotation, reflection

The complete graph K_5 The complete bipartite graph $K_{3,3}$





they are the smallest non-planar graphs.
a graph is planar iff it does not contain K_5 or $K_{3,3}$.

Edge Contraction

We say that $G' = (V', E')$ can be obtained from graph $G = (V, E)$ by contracting the edge $\{u, v\} \in E$ if

- $V' = (V \setminus \{u, v\}) \cup \{uv\}$, where $uv \notin V$ is a new vertex
- $E' = \{e \in E \mid e \cap \{u, v\} = \emptyset\} \cup \{\{uv, w\} \mid \{u, w\} \in E \text{ or } \{v, w\} \in E\}$.

Definition (minor)

We say that a graph G' is a minor of a graph G if it can be obtained from G through a sequence of transformations of the following kind:

1. remove a vertex (of degree 0) from the graph
2. remove an edge from the graph
3. contract an edge in the graph

Notes:

- If we only allowed the first two transformations, we would obtain the regular subgraph relationship.
- It follows that every subgraph is a minor, but the opposite is not true in general.

Theorem (Wagner's Theorem)

A graph is planar iff it does not contain K_5 or $K_{3,3}$ as a minor.

Theorem (Graph minor theorem)

Let Π be a minor-hereditary properties of graphs.

Then there exists a finite set of forbidden minors $F(\Pi)$ such that the following result holds:

A graph has property Π iff it does not have any graph from $F(\Pi)$ as a minor.

- the forbidden minors for planarity are K_5 and $K_{3,3}$
- the (only) forbidden minor for acyclicity is K_3,

Theorem

Let S be a finite set with n elements, and let $k \in \{0, \dots, n\}$. Then S has $\binom{n}{k}$ subsets of size k, where

$$\binom{n}{0} = 1$$
$$\binom{n}{n} = 1$$
$$\rightarrow \binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \qquad \text{for all } n \geq 1, 0 < k < n$$

Closed-form solution:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Definition (binary tree)

A binary tree is inductively defined as a tuple of the following form:

- The empty tree () is a binary tree. Such a tree is called a leaf.
- If L and R are binary trees, then (L, R) is a binary tree. Such a tree is called an inner node with left child L and right child R.

(L, R) and (R, L) are different trees (unless $L = R$)

Theorem

There are $C(n)$ binary trees with $n + 1$ leaves, where

Catalan numbers

$$C(0) = 1$$
$$C(n) = \sum_{k=0}^{n-1} C(k)C(n-k-1) \qquad \text{for all } n \geq 1$$

Closed-form solution (without proof):

$$C(n) = \frac{1}{n+1}\binom{2n}{n}$$

Definition (Fibonacci series)

The Fibonacci series F is defined as follows:

$$F(0) = 0$$
$$F(1) = 1$$
$$F(n) = F(n-1) + F(n-2) \qquad \text{for all } n \geq 2$$

Definition (golden ratio)

The number

$$\varphi = \frac{1+\sqrt{5}}{2}$$

is called the golden ratio.

$$\psi = -\frac{1}{\varphi}$$

Theorem

$$F(n) = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right) = \frac{1}{\sqrt{5}}(\varphi^n - \psi^n) \qquad \text{for all } n \geq 0$$

$$\psi = 1 - \varphi$$

$$\psi = \frac{1-\sqrt{5}}{2} = \frac{1+1-1-\sqrt{5}}{2} = \frac{2-(1+\sqrt{5})}{2} = \frac{2}{2} - \frac{1+\sqrt{5}}{2} = 1 - \varphi$$

$$\varphi^2 = \varphi + 1$$

$$\varphi^2 = \left(\frac{1+\sqrt{5}}{2}\right)^2 = \frac{1}{4}(1+\sqrt{5})^2 = \frac{1}{4}(1+2\sqrt{5}+5) = \frac{1}{4}(2+2\sqrt{5}+4) = \frac{1}{4}(2+2\sqrt{5}) + \frac{4}{4} = \frac{1}{2}(1+\sqrt{5}) + 1 = \varphi + 1$$

$$\psi^2 = \psi + 1$$

$$\psi^2 = (1-\varphi)^2 = 1 - 2\varphi + \varphi^2 = 1 - 2\varphi + \varphi + 1 = 1 - \varphi + 1 = (1-\varphi) + 1 = \psi + 1$$

Definition (power series)

Let $(a_n)_{n \in \mathbb{N}_0}$ be a sequence of real numbers.
The power series with coefficients (a_n) is the (possibly partial) function $g : \mathbb{R} \to \mathbb{R}$ defined by

$$g(x) = \sum_{n=0}^{\infty} a_n x^n \qquad \text{for all } x \in \mathbb{R}.$$

Definition (generating function)

Let $f : \mathbb{N}_0 \to \mathbb{R}$ be a function over the natural numbers.
The generating function for f is the power series with coefficients $(f(n))_{n \in \mathbb{N}_0}$.

Idea: partial fraction decomposition, i.e., find a, b, α, β such that $h(x) = \frac{a}{1-\alpha x} + \frac{b}{1-\beta x}$.

Definition (O, Ω, Θ)

Let $g : \mathbb{R}_0^+ \to \mathbb{R}$ be a function.
The sets of functions $O(g), \Omega(g), \Theta(g)$ are defined as follows:

- $O(g) = \{f : \mathbb{R}_0^+ \to \mathbb{R} \mid \text{there exist } C, n_0 \in \mathbb{R} \text{ s.t. } |f(n)| \leq C \cdot g(n) \text{ for all } n \geq n_0\}$
- $\Omega(g) = \{f : \mathbb{R}_0^+ \to \mathbb{R} \mid \text{there exist } C, n_0 \in \mathbb{R} \text{ s.t. } |f(n)| \geq C \cdot g(n) \text{ for all } n \geq n_0\}$
- $\Theta(g) = O(g) \cap \Omega(g)$

1. Construct A smaller inputs of size n/B.
2. Recursively solve these inputs using the same algorithm.
3. Compute the result from the recursively computed results.

If 1.+3. take time $f(n)$, the overall run-time for $n > C$ can be expressed as $T(n) = A \cdot T(n/B) + f(n)$.

we have n/2 sets

- Mergesort: $A = 2$, $B = 2$, $f(n) = \Theta(n)$
- Binary Search: $A = 1$, $B = 2$, $f(n) = \Theta(1)$

Theorem

Let $A \geq 1, B \geq 1$, and let T satisfy the divide-and-conquer recurrence $T(n) = A \cdot T(n/B) + f(n)$. Then:

- *If $f(n) = O(n^{\log_B A - \varepsilon})$ for some $\varepsilon > 0$, then $T(n) = \Theta(n^{\log_B A})$.*
- *If $f(n) = \Theta(n^{\log_B A})$, then $T(n) = \Theta(n^{\log_B A} \log_2 n)$.*
- *If $f(n) = \Omega(n^{\log_B A + \varepsilon})$ for some $\varepsilon > 0$, then $T(n) = \Theta(f(n))$.*

Definition (Syntax of Propositional Logic)

Let A be a set of atomic propositions. The set of propositional formulas (over A) is inductively defined as follows:

- Every atom $a \in A$ is a propositional formula over A.
- If φ is a propositional formula over A, then so is its negation $\neg\varphi$.
- If φ and ψ are propositional formulas over A, then so is the conjunction $(\varphi \wedge \psi)$.
- If φ and ψ are propositional formulas over A, then so is the disjunction $(\varphi \vee \psi)$.

Definition (Semantics of Propositional Logic)

A truth assignment (or interpretation) for a set of atomic propositions A is a function $\mathcal{I} : A \to \{0, 1\}$.

A propositional formula φ (over A) holds under $\mathcal{I}$ (written as $\mathcal{I} \models \varphi$) according to the following definition:

I satisfies phi

$\mathcal{I} \models a$	iff	$\mathcal{I}(a) = 1$	(for $a \in A$)
$\mathcal{I} \models \neg\varphi$	iff	not $\mathcal{I} \models \varphi$	
$\mathcal{I} \models (\varphi \wedge \psi)$	iff	$\mathcal{I} \models \varphi$ and $\mathcal{I} \models \psi$	
$\mathcal{I} \models (\varphi \vee \psi)$	iff	$\mathcal{I} \models \varphi$ or $\mathcal{I} \models \psi$	

$\mathcal{I} \models \varphi$ we also say $\mathcal{I}$ is a model of φ

$A = \{\text{DrinkBeer}, \text{EatFish}, \text{EatIceCream}\}$

$\mathcal{I} = \{\text{DrinkBeer} \mapsto 1, \text{EatFish} \mapsto 0, \text{EatIceCream} \mapsto 1\}$

$\varphi = (\neg\text{DrinkBeer} \to \text{EatFish})$

This means that if we want to prove $\mathcal{I} \models \varphi$, it is sufficient to prove

$$\mathcal{I} \models \neg\neg\text{DrinkBeer}$$

or to prove

$$\mathcal{I} \models \text{EatFish}.$$

Proof that $\mathcal{I} \models (\neg\text{DrinkBeer} \to \text{EatFish})$:

1. We have $\mathcal{I} \models \text{DrinkBeer}$ (uses defn. of $\models$ for atomic props. and fact $\mathcal{I}(\text{DrinkBeer}) = 1$).
2. From (1), we get $\mathcal{I} \not\models \neg\text{DrinkBeer}$ (uses defn. of $\models$ for negations).
3. From (2), we get $\mathcal{I} \models \neg\neg\text{DrinkBeer}$ (uses defn. of $\models$ for negations).
4. From (3), we get $\mathcal{I} \models (\neg\neg\text{DrinkBeer} \vee \psi)$ for all formulas ψ, in particular $\mathcal{I} \models (\neg\neg\text{DrinkBeer} \vee \text{EatFish})$ (uses defn. of $\models$ for disjunctions).
5. From (4), we get $\mathcal{I} \models (\neg\text{DrinkBeer} \to \text{EatFish})$ (uses defn. of "$\to$"). □

Definition (Equivalence of Propositional Formulas)

Two propositional formulas φ and ψ over A are (logically) equivalent ($\varphi \equiv \psi$) if for all interpretations $\mathcal{I}$ for A it is true that $\mathcal{I} \models \varphi$ if and only if $\mathcal{I} \models \psi$.

(absorption)

$(\varphi \wedge (\varphi \vee \psi)) \equiv \varphi$

$(\varphi \vee (\varphi \wedge \psi)) \equiv \varphi$

(tautology rules)

$(\varphi \vee \psi) \equiv \varphi$ if φ tautology

$(\varphi \wedge \psi) \equiv \psi$ if φ tautology

(unsatisfiability rules)

$(\varphi \vee \psi) \equiv \psi$ if φ unsatisfiable

$(\varphi \wedge \psi) \equiv \varphi$ if φ unsatisfiable

- Placement of parentheses for a conjunction of conjunctions does not influence whether an interpretation is a model.
- ditto for disjunctions of disjunctions
  - $\neg$ binds more strongly than $\wedge$
  - $\wedge$ binds more strongly than $\vee$
  - $\vee$ binds more strongly than $\to$ or $\leftrightarrow$

- $\bigwedge_{\varphi \in \emptyset} \varphi$ is a tautology.
- $\bigvee_{\varphi \in \emptyset} \varphi$ is unsatisfiable.
- $\bigwedge_{\varphi \in \{\chi\}} \varphi = \bigvee_{\varphi \in \{\chi\}} \varphi = \chi$
  - A literal is an atomic proposition or the negation of an atomic proposition (e. g., A and $\neg$A).
  - A clause is a disjunction of literals
  - A monomial is a conjunction of literals

The terms clause and monomial are also used for the corner case with only one literal.

- $((P \vee \neg Q) \wedge P)$ is neither literal nor clause nor monomial
- $\neg P$ is a literal, a clause and a monomial
- $(P \to Q)$ is neither literal nor clause nor monomial (but $(\neg P \vee Q)$ is a clause!)
- $(P \vee P)$ is a clause, but not a literal or monomial
- $\neg\neg P$ is neither literal nor clause nor monomial

Definition (Conjunctive Normal Form)

A formula is in conjunctive normal form (CNF) if it is a conjunction of clauses, i. e., if it has the form

$$\bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} L_{ij}$$

with $n, m_i > 0$ (for $1 \le i \le n$), where the L_{ij} are literals.

Definition (Disjunctive Normal Form)

A formula is in disjunctive normal form (DNF) if it is a disjunction of monomials, i. e., if it has the form

$$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} L_{ij}$$

with $n, m_i > 0$ (for $1 \le i \le n$), where the L_{ij} are literals.

- $((P \vee \neg Q) \to P)$ not CNF, not DNF
- P CNF or DNF, we can think of it as conjunction or disjunction of 1 element

P $\wedge$ Q is another example which is both: CNF and DNF

Algorithm to Construct CNF

1. Replace abbreviations $\to$ and $\leftrightarrow$ by their definitions (($\to$)-elimination and ($\leftrightarrow$)-elimination).
   $\rightsquigarrow$ formula structure: only $\vee$, $\wedge$, $\neg$
2. Move negations inside using De Morgan and double negation.
   $\rightsquigarrow$ formula structure: only $\vee$, $\wedge$, literals
3. Distribute $\vee$ over $\wedge$ with distributivity (strictly speaking also with commutativity).
   $\rightsquigarrow$ formula structure: CNF
4. optionally: Simplify the formula at the end or at intermediate steps (e. g., with idempotence).

Note: For DNF, distribute $\wedge$ over $\vee$ instead.

Definition (Model for Knowledge Base)

Let KB be a knowledge base over A, i. e., a set of propositional formulas over A.

A truth assignment $\mathcal{I}$ for A is a model for KB (written: $\mathcal{I} \models$ KB) if $\mathcal{I}$ is a model for every formula $\varphi \in$ KB.

Definition (Logical Consequence)

Let KB be a set of formulas and φ a formula.

We say that KB logically implies φ (written as KB $\models \varphi$) if all models of KB are also models of φ.

also: KB logically entails φ, φ logically follows from KB, φ is a logical consequence of KB

$\models$ is "overloaded": KB $\models \varphi$ vs. $\mathcal{I} \models \varphi$.

KB implies phi

I is a model of phi

unsatisfiable KB implies everything
empty KB is tautology

Let φ = DrinkBeer and

KB = {(¬DrinkBeer → EatFish),
((EatFish ∧ DrinkBeer) → ¬EatIceCream),
((EatIceCream ∨ ¬DrinkBeer) → ¬EatFish)}.

Show: KB $\models$ (implies) φ

Proof sketch.

Proof by contradiction: assume $\mathcal{I} \models$ KB, but $\mathcal{I} \not\models$ DrinkBeer.
Then it follows that $\mathcal{I} \models \neg$DrinkBeer.
Because $\mathcal{I}$ is a model of KB, we also have
$\mathcal{I} \models (\neg\text{DrinkBeer} \to \text{EatFish})$ and thus $\mathcal{I} \models$ EatFish. (Why?)
With an analogous argumentation starting from
$\mathcal{I} \models ((\text{EatIceCream} \vee \neg\text{DrinkBeer}) \to \neg\text{EatFish})$
we get $\mathcal{I} \models \neg$EatFish and thus $\mathcal{I} \not\models$ EatFish. $\rightsquigarrow$ Contradiction!

Theorem (Deduction Theorem)

KB $\cup \{\varphi\} \models \psi$ *iff* KB $\models (\varphi \to \psi)$

German: Deduktionssatz

Theorem (Contraposition Theorem)

KB $\cup \{\varphi\} \models \neg\psi$ *iff* KB $\cup \{\psi\} \models \neg\varphi$

swap the roles of phi and psi

German: Kontrapositionssatz

Theorem (Contradiction Theorem)

KB $\cup \{\varphi\}$ *is unsatisfiable iff* KB $\models \neg\varphi$

- Inference rules have the form $\frac{\varphi_1, \ldots, \varphi_k}{\psi}$

Definition (Derivation)

A derivation or proof of a formula φ from a knowledge base KB is a sequence of formulas $\psi_1, \ldots, \psi_k$ with

- $\psi_k = \varphi$ and
- for all $i \in \{1, \ldots, k\}$: hence psi_i must be deducible or given
  - $\psi_i \in$ KB, or
  - ψ_i is the result of the application of an inference rule to elements from $\{\psi_1, \ldots, \psi_{i-1}\}$.

Definition (Correctness and Completeness of a Calculus)

We write KB $\vdash_C \varphi$ if there is a derivation of φ from KB in calculus C.
(If calculus C is clear from context, also only KB $\vdash \varphi$.)

A calculus C is correct if for all KB and φ
KB $\vdash_C \varphi$ implies KB $\models \varphi$.

A calculus C is complete if for all KB and φ
KB $\models \varphi$ implies KB $\vdash_C \varphi$.

Definition (Refutation-Completeness)

A calculus C is refutation-complete if KB $\vdash_C \square$ for all unsatisfiable KB.

- **Widerlegungsvollständigkeit:**
  - Der RK ist widerlegungsvollständig. D.h., ist die zu untersuchende Formelmenge widersprüchlich, so findet man den Widerspruch mit einer endlichen Anzahl von Resolutionsschritten.

- Contradiction theorem: KB $\cup \{\varphi\}$ is unsatisfiable iff KB $\models \neg\varphi$
- This implies that KB $\models \varphi$ iff KB $\cup \{\neg\varphi\}$ is unsatisfiable.

called resolution rule:

$$\frac{C_1 \cup \{X\},\; C_2 \cup \{\neg X\}}{C_1 \cup C_2},$$

where C_1 and C_2 are (possibly empty) clauses and X is an atomic proposition.

- X and $\neg X$ are the resolution literals,
- $C_1 \cup \{X\}$ and $C_2 \cup \{\neg X\}$ are the parent clauses, and
- $C_1 \cup C_2$ is the resolvent.

Definition (Proof by Resolution)

A proof by resolution of a clause D from a knowledge base Δ is a sequence of clauses $C_1, \ldots, C_n$ with

- $C_n = D$ and
- for all $i \in \{1, \ldots, n\}$:
  - $C_i \in \Delta$, or
  - C_i is resolvent of two clauses from $\{C_1, \ldots, C_{i-1}\}$.

If there is a proof of D by resolution from Δ, we say that D can be derived with resolution from Δ and write $\Delta \vdash_R D$.

1. Reduce logical consequence to unsatisfiability.
2. Transform knowledge base into clause form (CNF).
3. Derive empty clause $\square$ with resolution.

Step 1: Reduce logical consequence to unsatisfiability.
KB $\models (R \vee S)$ iff KB $\cup \{\neg(R \vee S)\}$ is unsatisfiable.

Definition (Signature)

A signature (of predicate logic) is a 4-tuple $\mathcal{S} = \langle \mathcal{V}, \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ consisting of the following four disjoint sets:

- a finite or countable set $\mathcal{V}$ of variable symbols
- a finite or countable set $\mathcal{C}$ of constant symbols
- a finite or countable set $\mathcal{F}$ of function symbols
- a finite or countable set $\mathcal{P}$ of predicate symbols (or relation symbols)

Every function symbol $\mathsf{f} \in \mathcal{F}$ and predicate symbol $\mathsf{P} \in \mathcal{P}$ has an associated arity $ar(\mathsf{f}), ar(\mathsf{P}) \in \mathbb{N}_1$ (number of arguments).

Definition (Term)

Let $\mathcal{S} = \langle \mathcal{V}, \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ be a signature.
A term (over $\mathcal{S}$) is inductively constructed according to the following rules:

- Every variable symbol $v \in \mathcal{V}$ is a term.
- Every constant symbol $c \in \mathcal{C}$ is a term.
- If $t_1, \ldots, t_k$ are terms and $\mathsf{f} \in \mathcal{F}$ is a function symbol with arity k, then $\mathsf{f}(t_1, \ldots, t_k)$ is a term.

Definition (Formula)

For a signature $\mathcal{S} = \langle \mathcal{V}, \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ the set of predicate logic formulas (over $\mathcal{S}$) is inductively defined as follows:

- If $t_1, \ldots, t_k$ are terms (over $\mathcal{S}$) and $\mathsf{P} \in \mathcal{P}$ is a k-ary predicate symbol, then the atomic formula (or the atom) $\mathsf{P}(t_1, \ldots, t_k)$ is a formula over $\mathcal{S}$.
- If t_1 and t_2 are terms (over $\mathcal{S}$), then the identity $(t_1 = t_2)$ is a formula over $\mathcal{S}$.
- If $x \in \mathcal{V}$ is a variable symbol and φ a formula over $\mathcal{S}$, then the universal quantification $\forall x\,\varphi$ and the existential quantification $\exists x\,\varphi$ are formulas over $\mathcal{S}$.

- If φ is a formula over $\mathcal{S}$, then so is its negation $\neg\varphi$.
- If φ and ψ are formulas over $\mathcal{S}$, then so are the conjunction $(\varphi \wedge \psi)$ and the disjunction $(\varphi \vee \psi)$.

Definition (Interpretation, Variable Assignment)

An interpretation (for $\mathcal{S}$) is a pair $\mathcal{I} = \langle U, \cdot^{\mathcal{I}} \rangle$ of:

- a non-empty set U called the universe and
- a function $\cdot^{\mathcal{I}}$ that assigns a meaning to the constant, function, and predicate symbols:
  - $c^{\mathcal{I}} \in U$ for constant symbols $c \in \mathcal{C}$
  - $\mathsf{f}^{\mathcal{I}} : U^k \to U$ for k-ary function symbols $\mathsf{f} \in \mathcal{F}$
  - $\mathsf{P}^{\mathcal{I}} \subseteq U^k$ for k-ary predicate symbols $\mathsf{P} \in \mathcal{P}$

A variable assignment (for $\mathcal{S}$ and universe U) is a function $\alpha : \mathcal{V} \to U$. maps variable to objects