Relations, Graphs and Boolean Algebras

Hans Cuypers

October 17, 2001

Abstract
Notes for the course Wiskunde 3 voor Technische Informatica, 2F820

Contents

1. Relations 2
   1.1. Definitions and Examples 2
   1.2. Product of Relations 4

2. Maps 7
   2.1. Maps 7
   2.2. Domain and Image 8

3. Graphs and Networks 12
   3.1. Graphs 12
   3.2. Euler and Hamilton Cycles 14
   3.3. Spanning Trees and Search Algorithms 16
   3.4. Networks 19

4. Order Relations 23
   4.1. Partial Orders 23
   4.2. Lattices 26

5. Boolean Algebras 31
   5.1. Definitions 31
   5.2. Examples from Logic and Electrical Engineering 34
   5.3. The Structure of Boolean Algebras 36

6. Morphisms 39
   6.1. Morphisms of Relations 39
   6.2. Continuous Maps 41
   6.3. Fixed Point Theorems 43
1. Relations

1.1. Definitions and Examples

A (binary) relation $R$ between the sets $A$ and $B$ is a subset of the cartesian product $A \times B$. If $(a,b) \in R$, we say $a$ is in relation $R$ to be $b$. We denote this by $aRb$. If $A = B$ we say $R$ is a relation on $A$.

Example 1.1 We give some examples:

- “Is the mother of” is a relation between the set of all females and the set of all people.
- “There is a train connection between” is a relation between the cities of the Netherlands.
- The identity relation “=” is a relation on a set $A$.
- divides $|$ is a relation on $\mathbb{N}$.
- “Greater than” $>$ or “less than” $<$ are relations on $\mathbb{R}$.
- $R = \{(0,0),(1,0),(2,1)\}$ is a relation between the sets $A = \{0,1,2\}$ and $B = \{0,1\}$.
- $R = \{(x,y) \in \mathbb{R}^2 | y = x^2\}$ is a relation on $\mathbb{R}$.
- Let $\Omega$ be a set, then “is a subset of” $\subseteq$ is a relation on the set $A$ of all subsets of $\Omega$.

Besides binary relations one can also consider $n$-ary relations with $n > 2$. An $n$-ary relation $R$ on the sets $A_1,\ldots,A_n$ is a subset of the cartesian product $A_1 \times \cdots \times A_n$.

Unless stated otherwise, a relation $R$ is assumed to be a binary relation.

Some relations have special properties:

Definition 1.2 Let $R$ be a relation on a set $A$. Then $R$ is called

- Reflexive if for all $x \in A$ we have $(x,x) \in R$;
- Irreflective if for all $x \in A$ we have $(x,x) \notin R$;
- Symmetric if for all $x,y \in A$ we have $xRy$ implies $yRx$;
- Antisymmetric if for all $x \neq y \in A$ we have if $xRy$ then $y \notRx$;
- Transitive if for all $x,y,z \in A$ we have that $xRy$ and $yRz$ implies $xRz$. 
Example 1.3 We consider some of the examples given above:

- “Is the mother of” is a relation on the set of all people. This relation is irreflexive, antisymmetric and not transitive.

- “There is a train connection between” is a symmetric and transitive relation.

- “=” is a reflexive, symmetric and transitive relation on a set $A$.

- divides $|$ is a reflexive and transitive relation on $\mathbb{N}$.

- “Greater than” $>$ or “less than” $<$ on $\mathbb{R}$ are irreflexive, antisymmetric and transitive.

- The relation $R = \{(x, y) \in \mathbb{R}^2 \mid y = x^2\}$ is not reflexive nor irreflexive.

As we noticed in the above example, “being equal” is a reflexive, symmetric and transitive relation on any set $A$. Relations having these three properties deserve some special attention.

Definition 1.4 A relation $R$ on a set $A$ is called an equivalence relation on $A$ if and only if it is reflexive, symmetric and transitive.

Example 1.5 Consider the plane $\mathbb{R}^2$ and in it the set $A$ of straight lines. We call two lines parallel in $A$ if and only if they are equal or do not intersect. Notice that two lines in $A$ are parallel if and only if their slope is equal. Being parallel defines an equivalence relation on the set $A$.

Example 1.6 Fix $n \in \mathbb{Z}$, $n \neq 0$, and consider the relation $R$ on $\mathbb{Z}$ by $aRb$ if and only if $a - b$ is divisible by $n$. We also write $a = b \mod n$. As we have seen in Chapter 2 of [3], this is indeed an equivalence relation.

Example 1.7 Let $\Pi$ be a partition of the set $A$, i.e., $\Pi$ is a set of nonempty subsets of $A$ such each element of $A$ is in a unique member of $\Pi$. In particular, the union of all members of $\Pi$ yields the whole set $A$ and any two members of $\Pi$ have empty intersection.

We define the relation $R_{\Pi}$ as follows: $a, b \in A$ are in relation $R_{\Pi}$ if and only if there is a subset $X$ of $A$ in $\Pi$ containing both $a$ and $b$. we check that the relation $R_{\Pi}$ is an equivalence relation on $A$.

- Reflexivity. Let $a \in A$. Then there is an $X \in \Pi$ containing $a$. Hence $a, a \in X$ and $aR_{\Pi}a$

- Symmetry. Let $aR_{\Pi}b$. then there is an $X \in \Pi$ with $a, b \in X$. But then also $b, a \in X$ and $bR_{\Pi}a$. 

3
Transitivity. If \( a, b, c \in A \) with \( aR_{\Pi}b \) and \( bR_{\Pi}c \), then there are \( X, Y \in \Pi \) with \( a, b \in X \) and \( b, c \in Y \). However, then \( b \) is in both \( X \) and \( Y \). But then, as \( \Pi \) partitions \( A \), we have \( X = Y \). So \( a, c \in X \) and \( aR_{\Pi}c \).

The following theorem implies that every equivalence relation on a set \( A \) can be obtained as a partition of the set \( A \). But before we can state the result we need some more terminology and notation.

Let \( R \) be a relation on a set \( A \). Then for each element \( a \in A \) we define by 
\[
[a]_R := \{ b \in A \mid aRb \}.
\]
(Sometimes this set is also denoted by \( R(a) \).)

**Lemma 1.8** If \( b \in [a]_R \), then \( [b]_R = [a]_R \).

**Proof.** Suppose \( b \in [a]_R \). Thus \( aRb \). If \( c \in [b]_R \), then \( bRc \) and, as \( aRb \), we have by transitivity \( aRc \). In particular, \( [b]_R \subseteq [a]_R \).

Since, by symmetry of \( R \), \( aRb \) implies \( bRa \) and hence \( a \in [b]_R \), we similarly get \( [a]_R \subseteq [b]_R \). \( \square \)

**Theorem 1.9** Let \( R \) be an equivalence relation on a set \( A \). Then the set of \( R \)-equivalence classes partitions the set \( A \).

**Proof.** Let \( \Pi_R \) be the set of \( R \)-equivalence classes. Then by reflexivity of \( R \) we find that each element \( a \in A \) is inside the class \( [a]_R \) of \( \Pi_R \).

If an element \( a \in A \) is in the classes \( [b]_R \) and \( [c]_R \) of \( \Pi \), then by the previous lemma we find \( [b]_R = [a]_R \) and \( [b]_R = [c]_R \). In particular \( [b]_R \) equals \( [c]_R \). Thus each element \( a \in A \) is inside a unique member of \( \Pi_R \), which therefore is a partition of \( A \). \( \square \)

### 1.2. Product of Relations

If \( R_1 \) and \( R_2 \) are two relations between a set \( A \) and a set \( B \), then we can form new relations between \( A \) and \( B \) by taking the intersection \( R_1 \cap R_2 \) or the union \( R_1 \cup R_2 \). Also the complement of \( R_2 \) in \( R_1 \), \( R_1 - R_2 \), is a new relation. Furthermore we can consider a relation \( R^\top \) (sometimes also denoted by \( R^{-1} \)) from \( B \) to \( A \) as the relation 
\[
\{(b, a) \in B \times A \mid (a, b) \in R\}.
\]

Another way of making new relations out of old ones is the following. If \( R_1 \) is a relation between \( A \) and \( B \) and \( R_2 \) is a relation between \( B \) and \( C \) then the composition or product \( R = R_1 \circ R_2 \) (sometimes denoted by \( R_2 \circ R_1 \)) is the relation between \( A \) and \( C \) defined by \( aRc \) if and only if there is a \( b \in B \) with \( aR_1b \) and \( bR_2c \).
Proposition 1.10 Suppose $R_1$ is a relation from $A$ to $B$, $R_2$ a relation from $B$ to $C$ and $R_3$ a relation from $C$ to $D$. Then $R_1 \ast (R_2 \ast R_3) = (R_1 \ast R_2) \ast R_3$.

Proof. Suppose $a \in A$ and $d \in D$ with $aR_1 \ast (R_2 \ast R_3)d$. Then we can find a $b \in B$ with $aR_1b$ and $b(R_2 \ast R_3)d$. But then there is also a $c$ with $bR_2c$ and $cR_3d$. For this $c$ we have $aR_1 \ast R_2c$ and $cR_3d$ and hence $a \ast (R_1 \ast R_2) \ast R_3d$.

Similarly, if $a \in A$ and $d \in D$ with $a\ast (R_1 \ast R_2) \ast R_3d$, then we can find a $c \in C$ with $a(R_1 \ast R_2)c$ and $cR_3d$. But then there is also a $b \in B$ with $aR_1b$ and $bR_2c$. For this $b$ we have $bR_2 \ast R_3c$ and $aR_1b$ and hence $a \ast R_1 \ast (R_2 \ast R_3)d$. □

Let $R$ be a relation on a set $A$ and denote by $I$ the identity relation on $A$, i.e., $I = \{(a, b) \in A \times B \mid a = b\}$. Then we easily check that $I \ast R = R \ast I = R$.

We can view a relation $R$ on a set $A$ as a directed graph $\Gamma_R$ with vertex set $A$ and edge set $R$. Then two vertices $a$ and $b$ are in relation $R^2 = R \ast R$, if and only if there is a $c \in A$ such that both $(a, c)$ and $(c, b) \in R$. Thus $aR^2b$ if and only if they are at distance 2 in the graph defined by $R$.

For $n > 1$, the $n$-th power $R^n$ of the relation $R$ is recursively defined by $R^1 = R$ and $R^n = R \ast R^{n-1}$. Two vertices $a$ and $b$ are in relation $R^n$ if and only if, inside $\Gamma_R$, there is a path from $a$ to $b$ of length $n$.

We notice that whenever $R$ is reflexive, we have $R \subseteq R^2$ and thus also $R \subseteq R^n$ for all $n \in \mathbb{N}$. Actually, $a$ and $b$ are then in relation $R^n$ if and only if they are at distance $\leq n$ in the graph $\Gamma_R$.

Notice that for any collection $\mathcal{C}$ of transitive (reflexive or symmetric) relations the intersection $\bigcap_{R \in \mathcal{C}} R$ is again transitive (reflexive or symmetric). The reflexive, symmetric or transitive closure of a relation $R$ on a set $A$ is the smallest reflexive, symmetric or transitive relation containing $R$. This means it is the intersection $\bigcap_{R' \in \mathcal{C}} R'$, where $\mathcal{C}$ is the collection of all reflexive, symmetric or transitive relations containing $R$. One easily checks that the reflexive closure of a relation $R$ equals the relation $I \cup R$ and the
symmetric closure equals $R \cup R^\top$. The transitive closure is a bit more complicated. It contains $R, R^2, \ldots$. In particular, it contains $\bigcup_{n>0} R^n$, and, as we will show below, is equal to it.

**Proposition 1.11** $\bigcup_{n>0} R^n$ is the transitive closure of the relation $R$.

*Proof.* Define $\bar{R} = \bigcup_{n>0} R^n$. We prove transitivity of $\bar{R}$. Let $a \bar{R} b$ and $b \bar{R} c$, then there are sequences $a_1 = a, \ldots, a_k = b$ and $b_1 = b, \ldots, b_l = c$ with $a_i Ra_{i+1}$ and $b_i Rb_{i+1}$. But then the sequence $c_1 = a_1 = a, \ldots, c_k = a_k = b_1, \ldots, c_{k+l-1} = b_l = c$ is a sequence from $a$ to $c$ with $c_i Rc_{i+1}$. Hence $a R^{k+l-2} c$ and $a \bar{R} c$. □

**Example 1.12** If we consider the whole world wide web as a set of documents, then we may consider two documents to be in a relation $R$ if there is a hyperlink from one document to the other.

The reflexive and transitive closure of the relation $R$ defines a partition of the web into independent subwebs.

**Example 1.13** Let $A$ be the set of railway stations in the Netherlands. Two stations $a$ and $b$ are in relation $R$ if there is a train running directly from $a$ to $b$.

If $\bar{R}$ denotes the transitive closure of $R$, then the railway stations in $[a]_{\bar{R}}$ are exactly those stations you can reach by train when starting in $a$.

**Exercise 1.1** Which of the following relations on the set $A = \{1, 2, 3, 4\}$ is reflexive, irreflexive, symmetric, antisymmetric or transitive?

i. $\{(1, 3), (2, 4), (3, 1), (4, 2)\}$;

ii. $\{(1, 3), (2, 4)\}$;

iii. $\{(1, 1), (2, 2), (3, 3), (4, 4), (1, 3), (2, 4), (3, 1), (4, 2)\}$;

iv. $\{(1, 1), (2, 2), (3, 3), (4, 4)\}$;

v. $\{(1, 1), (2, 2), (3, 3), (4, 4), (1, 2), (2, 3), (3, 4), (4, 3), (3, 2), (2, 1)\}$.

**Exercise 1.2** Compute for each of the relations in Exercise 1.1 the reflexive closure, the symmetric closure and the transitive closure.

**Exercise 1.3** Suppose $R$ is a reflexive and transitive relation on $A$. Show that $R^2 = R$.

**Exercise 1.4** Suppose $R_1$ and $R_2$ are two relations on a set $A$. Let $R$ be the product $R_1 * R_2$. Prove or disprove the following statements

i. If $R_1$ and $R_2$ are reflexive, then so is $R$. 

ii. If $R_1$ and $R_2$ are irreflexive, then so is $R$.

iii. If $R_1$ and $R_2$ are symmetric, then so is $R$.

iv. If $R_1$ and $R_2$ are antisymmetric, then so is $R$.

v. If $R_1$ and $R_2$ are transitive, then so is $R$.

2. Maps

2.1. Maps

Examples of maps are the well known functions $f : \mathbb{R} \to \mathbb{R}$ given by $f(x) = x^2$, $f(x) = \sin x$, or $f(x) = \frac{1}{x^2 + 1}$. We can view these maps as relations on $\mathbb{R}$. Indeed, the function $f : \mathbb{R} \to \mathbb{R}$ can be viewed as the relation $\{(x, y) \mid y = f(x)\}$. Maps are special relations:

**Definition 2.1** A relation $F$ from a set $A$ to a set $B$ is called a map or function from $A$ to $B$ if for each $a \in A$ there is one and only one $b \in B$ with $aFb$.

If $F$ is a map from $A$ to $B$, we write this as $F : A \to B$. Moreover, if $a \in A$ and $b \in B$ is the unique element with $aFb$, then we write $b = F(a)$.

A partial map $F$ from a set $A$ to a set $B$ is a relation with the property that for each $a \in A$ there is at most one $b$ with $aFb$. In other words, it is a map from a subset $A'$ of $A$ to $B$.

**Example 2.2** We have encountered numerous examples of maps. Below you will find some familiar ones.

i. polynomial functions like $f : \mathbb{R} \to \mathbb{R}$, with $f(x) = x^3$ for all $x$.

ii. goniometric functions like cos, sin and tan.

iii. $\sqrt{\cdot} : \mathbb{R}^+ \to \mathbb{R}$, taking square roots.

iv. $\ln : \mathbb{R}^+ \to \mathbb{R}$, the natural logarithm.

And some less familiar ones.

i. $\gcd : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $\lcm : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$.

ii. $\phi : \mathbb{N} \to \mathbb{N}$, where $\phi$ is the Euler $\phi$-function.

iii. $\text{mod} : \mathbb{Z} \to \mathbb{Z}/n\mathbb{Z}$, where $z \in \mathbb{Z}$ maps to $z \mod n$.

If $f : A \to B$ and $g : B \to C$, then we can consider the product $f \ast g$ as a relation from $A$ to $C$. We also use the notation $g \circ f$ and call it the composition of $f$ and $g$. 
Proposition 2.3  Let $f : A \to B$ and $g : B \to C$ be maps, then the composition $g \circ f$ is a map from $A$ to $C$.

Proof. Let $a \in A$, then $g(f(a))$ is an element in $C$ in relation $f \ast g$ with $a$. If $c \in C$ is an element in $C$ that is in relation $f \ast g$ with $a$, then there is a $b \in B$ with $afb$ and $bgc$. But then, as $f$ is a map, $b = f(a)$ and, as $g$ is a map, $c = g(b)$. Hence $c = g(b) = g(f(a))$. □

2.2. Domain and Image

Let $A$ and $B$ be two sets and $f : A \to B$ a map from $A$ to $B$. The set $A$ is called the domain of $f$. If $a \in A$, then the element $b = f(a)$ is called the image of $a$ under $f$. The subset of $B$ consisting of the images of the elements of $A$ under $f$ is called the image or range of $f$ and is denote by $\text{Im}(f)$. So

$$\text{Im}(f) = \{ b \in B \mid \text{there is an } a \in A \text{ with } b = f(a) \}.$$ 

If $A'$ is a subset of $A$, then the image of $A'$ under $f$ is the set $f(A') = \{ f(a) \mid a \in A' \}$. So, $\text{Im}(f) = f(A)$.

If $a \in A$ and $b = f(a)$, then the element $a$ is called a pre-image of $b$. Notice that $b$ can have more than one pre-image. Indeed if $f : \mathbb{R} \to \mathbb{R}$ is given by $f(x) = x^2$ for all $x \in \mathbb{R}$, then both $-2$ and $2$ are pre-images of $4$. The set of all pre-images of $b$ is denoted by $f^{-1}(b)$. So,

$$f^{-1}(b) = \{ a \in A \mid f(a) = b \}.$$ 

If $B'$ is a subset of $B$ the the pre-image of $B'$, denoted by $f^{-1}(B')$ is the set of elements $a$ from $A$ that are mapped to an element $b$ of $B'$. In particular,

$$f^{-1}(B') = \{ a \in A \mid f(a) \in B' \}.$$ 

Example 2.4  

i. Let $f : \mathbb{R} \to \mathbb{R}$ with $f(x) = x^2$ for all $x \in \mathbb{R}$. Then $f^{-1}([0, 4]) = [-2, 2]$.

ii. Consider the map mod 8 from $\mathbb{Z}$ to $\mathbb{Z}$. The inverse image of 3 is the set $3 \text{ mod } 8 = \{ \ldots, -5, 3, 11, \ldots \}$.

Theorem 2.5  Let $f : A \to B$ be a map.

- If $A' \subseteq A$, then $f^{-1}(f(A')) \supseteq A'$.
- If $B' \subseteq B$, then $f(f^{-1}(B')) \subseteq B'$.
Proof. Let \( a' \in A' \), then \( f(a') \in f(A') \) and hence \( a' \in f^{-1}(f(A')) \). Thus \( A' \subseteq f^{-1}(f(A')) \).

Let \( a \in f^{-1}(B') \), then \( f(a) \in B' \). Thus \( f(f^{-1}(B')) \subseteq B' \). \( \Box \)

Example 2.6 Let \( f : \mathbb{R} \to \mathbb{R} \) be defined by \( f(x) = x^2 \) for all \( x \in \mathbb{R} \). Then \( f^{-1}([0, 1]) \) equals \([−1, 1]\) and thus properly contains \([0, 1]\). Moreover, \( f(f^{-1}([-4, 4])) = [0, 4] \) which is properly contained in \([-4, 4]\). This shows that we can have strict inclusions in the above theorem.

Theorem 2.7 Let \( f : A \to B \) and \( g : B \to C \) be maps. Then \( \text{Im}(g \circ f) = g(f(A)) \subseteq \text{Im}(g) \).

Definition 2.8 A map \( f : A \to B \) is called surjective, if for every \( b \in B \) there is an \( a \in A \) with \( b = f(a) \). In other words if \( \text{Im}(f) = B \).

The map \( f \) is called injective if for each \( b \in B \), there is at most one \( a \) with \( f(a) = b \). So the pre-image of \( b \) is either empty or consist of a unique element. In other words, \( f \) is injective if for any elements \( a \) and \( a' \) from \( A \) we find that \( f(a) = f(a') \) implies \( a = a' \).

The map \( f \) is bijective if it is both injective and surjective. So, if for each \( b \in B \) there is a unique \( a \in A \) with \( f(a) = b \).

![A surjective map from A to B](image)

Example 2.9 i. The map \( \sin : \mathbb{R} \to \mathbb{R} \) is not surjective nor injective.

ii. The map \( \sin : [-\pi/2, \pi/2] \to \mathbb{R} \) is injective but not surjective.

iii. The map \( \sin : \mathbb{R} \to [-1, 1] \) is a surjective map. It is not injective.
An injective map from $A$ to $B$.

A bijective map from $A$ to $B$.
iv. The map \( \sin : \left[ -\pi/2, \pi/2 \right] \rightarrow [-1,1] \) is a bijective map.

**Theorem 2.10 [Pigeonhole Principle]** Let \( f : A \rightarrow B \) be a map between two sets of size \( n \in \mathbb{N} \). Then \( f \) is injective if and only if it is surjective.

**Remark 2.11** The above result is called the pigeonhole principle because of the following. If one has \( n \) pigeons (the set \( A \)) and the same number of holes (the set \( B \)), then one pigeonhole is empty if and only if one of the other holes contains at least two pigeons.

**Example 2.12** The pigeonhole principle has been used in the proof of Fermat’s Little Theorem as given in Chapter 2 of [3].

If \( f : A \rightarrow B \) is a bijection, i.e., a bijective map, then for each \( b \in B \) we can find a unique \( a \in A \) with \( f(a) = b \). So, also the relation \( f^T = \{(b,a) \in B \times A \mid (a,b) \in f\} \) is a map. This map is called the inverse map of \( f \) and denoted by \( f^{-1} \).

**Proposition 2.13** Let \( f : A \rightarrow B \) be a bijection. Then for all \( a \in A \) and \( b \in B \) we have \( f^{-1}(f(a)) = a \) and \( f(f^{-1}(b)) = b \). In particular, \( f \) is the inverse of \( f^{-1} \).

**Proof.** Let \( a \in A \). Then \( f^{-1}(f(a)) = a \) by definition of \( f^{-1} \). If \( b \in B \), then, by surjectivity of \( f \), there is an \( a \in A \) with \( b = f(a) \). So, by the above, \( f(f^{-1}(b)) = f(f^{-1}(f(a))) = f(a) = b \).

**Theorem 2.14** Let \( f : A \rightarrow B \) and \( g : B \rightarrow C \) be two maps.

i. If \( f \) and \( g \) are surjective, then so is \( g \circ f \);

ii. If \( f \) and \( g \) are injective, then so is \( g \circ f \);

iii. If \( f \) and \( g \) are bijective, then so is \( g \circ f \).

**Proof.**

i. Let \( c \in C \). By surjectivity of \( g \) there is a \( b \in B \) with \( g(b) = c \). Moreover, since \( f \) is surjective, there is also an \( a \in A \) with \( f(a) = b \). In particular, \( g \circ f(a) = g(f(a)) = g(b) = c \). This proves \( g \circ f \) to be surjective.

ii. Let \( a, a' \in A \) with \( g \circ f(a) = g \circ f(a') \). Then \( g(f(a)) = g(f(a')) \) and by injectivity of \( g \) we find \( f(a) = f(a') \). Injectivity of \( f \) implies \( a = a' \). This shows that \( g \circ f \) is injective.

iii. (i) and (ii) imply (iii).
Lemma 2.15 Suppose $f : A \to B$ and $g : B \to C$ are bijective maps. Then the inverse of the map $g \circ f$ equals $f^{-1} \circ g^{-1}$.

Proof. $(f^{-1} \circ g^{-1})(g \circ f)(a) = f^{-1}(g^{-1}(g(f(a)))) = f^{-1}(f(a)) = a$. □

Exercise 2.1 Which of the following relations are maps from $A = \{1, 2, 3, 4\}$ to $A$?

i. $\{(1,3), (2,4), (3,1), (4,2)\}$;

ii. $\{(1,3)(2,4)\}$;

iii. $\{(1,1), (2,2), (3,3), (4,4), (1,3), (2,4), (3,1), (4,2)\}$;

iv. $\{(1,1), (2,2), (3,3), (4,4)\}$.

Exercise 2.2 Suppose $f$ and $g$ are maps from $\mathbb{R}$ to $\mathbb{R}$ defined by $f(x) = x^2$ and $g(x) = x + 1$ for all $x \in \mathbb{R}$. What is $g \circ f$ and what is $f \circ g$?

Exercise 2.3 Let $A = \{1, 2, 3, 4\}$ and $R_1 = \{(1,2), (1,3), (2,4), (2,2), (3,4), (4,3)\}$ and $R_2 = \{(1,1), (1,2), (3,1), (4,3), (4,4)\}$. Compute $R_1 \ast R_2$ and $R_2 \ast R_1$.

Exercise 2.4 Which of the following maps is injective, surjective or bijective?

i. $f : \mathbb{R} \to \mathbb{R}, f(x) = x^2$ for all $x \in \mathbb{R}$.

ii. $f : \mathbb{R} \to \mathbb{R}_{\geq 0}, f(x) = x^2$ for all $x \in \mathbb{R}$.

iii. $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}, f(x) = x^2$ for all $x \in \mathbb{R}$.

3. Graphs and Networks

3.1. Graphs

As we noticed before, a directed graph with vertex set $A$ and edge set $R$ is nothing else than a binary relation $R$ on a set $A$. An ordinary graph with vertex set $A$ has as edges subsets of size two from $A$. A symmetric and irreflexive relation $R$ is often identified with the ordinary graph with edges $\{a,b\}$ where $(a,b) \in R$.

Many questions about relations can be phrased in terms of graphs. In this section we will discuss some of them. Although many of the definitions and result can also be stated for directed graphs, we will restrict our attention to ordinary graphs. We start with introducing some graph theoretical notation.
Let $\Gamma = (V, E)$ be an ordinary graph with vertex set $V$ and edge set $E$. A walk in $\Gamma$ is a sequence $(v_1, \ldots, v_n)$ in $V$ such that $\{v_i, v_{i+1}\} \in E$. A path in $\Gamma$ is a sequence $(v_1, \ldots, v_n)$ in $V$ such that $\{v_i, v_{i+1}\} \in E$ and $v_i \neq v_{i+2}$. The length of the path $(v_1, \ldots, v_n)$ equals $n - 1$. The point $v_1$ is the starting point of the path and $v_n$ is the end point. A shortest path from $a$ to $b$ is a path from $a$ to $b$ of minimal length. The distance between two points equals the length of a shortest path between them. If there is no such path, the distance is set to infinity. The graph $\Gamma$ is called connected whenever for every two vertices $a$ and $b$ of $\Gamma$ there is a path with starting point $a$ and end point $b$. A cycle is a path $(v_1, \ldots, v_n)$ with the same starting and end point so, $v_1 = v_n$. A connected graph without cycles is called a tree. The degree $\deg(v)$ of a vertex $v \in V$ equals the cardinality of the set $\Gamma_v = \{w \in V \mid \{v, w\} \in E\}$. A tree contains vertices of degree 1. These special vertices are called the leaves of the tree.

**Exercise 3.1** A graph $\Gamma$ is called regular of degree $d$ if all its vertices have degree $d$. Show that in a finite regular graph $\Gamma = (V, E)$ of degree $d$ we have

$$|V| \cdot d = 2 \cdot |E|.$$ 

**Exercise 3.2** A tree $\Gamma$ is called binary if all its vertices either have degree $\leq 3$ or are leaves and one vertex, called the root and denoted by $r$, is a vertex of degree 2.

Let $\Gamma$ be a finite binary tree. Fix a vertex $r$ (called root) of $\Gamma$. By $l$ we denote the number of leaves of $\Gamma$ and by $d$ the maximal distance of a vertex $v$ in $\Gamma$ to $r$. Prove that $l \leq 2^d$.

For sorting algorithms based on comparison of two elements we can make a decision tree. For example, for sorting the three elements $\{a, b, c\}$ the decision tree looks as follows:

How many leaves does such a decision tree have when sorting $n$ elements? Suppose $r$ is the starting point of the decision tree. Can you give a lower bound on $d$? What
does this mean for the complexity of sorting algorithms based on comparison of two elements?

3.2. Euler and Hamilton Cycles

Example 3.1 One of the most famous problems in graph theory is the problem of the bridges of Königsberg (now called Kaliningrad). The Prussian city of Königsberg was divided into 4 parts by a river, one of these parts being an island in the river. The 4 regions of the city were connected by 6 bridges. On sunny Sundays the people of Königsberg used to go walking along the river and over the bridges. The question is, is there a walk using all the bridges once, that brings the pedestrian back to its starting point. The problem was solved by Leonard Euler, who showed that such a walk is not possible.

A cycle in a graph is called an Euler cycle if it contains all edges of the graph once.

The following result is due to Euler.

Theorem 3.2 A finite connected graph contains an Euler cycle if and only if all vertices have even degree.

Proof. Suppose Γ is a graph admitting an Euler cycle E. Suppose v is a vertex of the graph. Then inside the Euler cycle E, we see v just as often as an end point of an edge from E as a starting point. Hence the number of edges on v is even.

Now suppose all vertices of the finite graph Γ have even degree. We start a path of Γ in the vertex v. Each time we arrive in a vertex u, there are only an odd number of edges on u in the path. Except when we arrive back in v. If the path $P_1$ constructed in this way is not an Euler cycle yet, there is at least one edge, e say, in Γ not visited yet. We may even assume, by connectedness of Γ, that this edge contains a vertex w from $P_1$. Now we make a path $P_2$ starting in w containing e. As soon as, in the process of constructing this path, we hit on a vertex of $P_1$, then, as we only have met an odd number of edges on this vertex, there has to be an edge not in $P_1$ and not yet part of $P_2$. So, the path $P_2$ may be constructed in such a way that it has no edges in common.
with $P_1$. The paths $P_1$ and $P_2$ can be combined to a path $P$ (with more edges than $P_1$) in the following way. Start in $v$, follow the path $P_1$ until one reaches $w$, then follow the path $P_2$ completely, and finally continue from $w$ over the path $P_1$ to end in $v$. Repeating this process will eventually yield an Euler cycle in $\Gamma$. \qed

Notice that the above not only proves the theorem, but also describes an algorithm to find an Euler cycle.

A Hamilton cycle in a graph $\Gamma$ is a cycle in the graph containing every vertex exactly ones. It does not seem to be possible to given a characterization of graphs admitting Hamilton cycles, similar to Theorem 3.2. However, the following result due to Ore, gives a partial result in this direction.

**Theorem 3.3** If in the finite graph $\Gamma$ on $v$ vertices for every two nonadjacent vertices $a$ and $b$ we have $\deg(a) + \deg(b) \geq v$, then $\Gamma$ contains a Hamilton cycle.

**Proof.** Suppose $\Gamma$ is a finite graph on $v$ vertices and for any two nonadjacent vertices $a$ and $b$ we have $\deg(a) + \deg(b) \geq v$. Suppose $\Gamma$ contains no Hamilton cycle. We add edges to $\Gamma$ as long as possible, but we avoid producing a Hamilton cycle. Since the complete graph on $v$ vertices admits a Hamilton cycle, we end up with a graph $\Gamma_0$ in which there is no Hamilton cycle, but addition of any new edges produces one. Notice that, also in $\Gamma_0$, we still have that for any two nonadjacent vertices $a$ and $b$ the sum $\deg(a) + \deg(b)$ is greater or equal than $v$.

Now suppose $a, b$ are nonadjacent vertices. Then after adding the edge $\{a, b\}$ to $\Gamma_0$ we obtain a Hamilton cycle $a = a_0, a_1, \ldots, a_v = b$ in $\Gamma_0$. Since $\deg(a) + \deg(b) \geq v$, there are two vertices $a_i$ and $a_{i+1}$ with $b$ adjacent to $a_i$ and $a$ adjacent to $a_{i+1}$. However, then consider the path $a = a_1, \ldots, a_i, b = a_v, a_{v-1}, \ldots, a_{i+1}, a$ in $\Gamma_0$. This is a Hamilton cycle and contradicts our assumption on $\Gamma_0$. Thus $\Gamma$ has to have a Hamilton cycle. \qed
Exercise 3.3 How can one use Theorem 3.2 to solve the problem of the Königsberger bridges, see 3.1?

Exercise 3.4 An Euler path is a path from a vertex $a$ to a vertex $b$ in a graph containing all edges of the graph just ones. Show that a finite connected graph $\Gamma$ contains an Euler path from $a$ to $b$ if and only if $a$ and $b$ are the only vertices of odd degree in $\Gamma$.

Exercise 3.5 How does Theorem 3.2 generalize to directed graphs?

Exercise 3.6 Can one find an Euler cycle in the cube? And a Hamilton cycle?

Exercise 3.7 Can one find an Euler cycle in the following graph? And a Hamilton cycle?

3.3. Spanning Trees and Search Algorithms

Example 3.4 A search engine at the university wants to search all the web servers on campus. These servers are connected by cables within an intranet. To do its search the search engine does not want to put too much load on the connections between the different web servers. In fact, it wants to use as as few connections as possible. If we represent the computer network as a graph with the web servers (and search engine) as vertices, two vertices connected, if and only if there is a cable connecting them, then we can phrase the problem as follows: find a connected subgraph on all the vertices with as few edges as possible.

Trees are graphs with the least number of vertices as follows from the following theorem.
Theorem 3.5 Let $\Gamma$ be a finite connected graph on $v$ vertices. Then $\Gamma$ contains at least $v - 1$ edges, with equality if and only if $\Gamma$ is a tree.

Proof. Notice that in a connected graph each vertex is on at least one edge. We prove this result with induction to the number $v$ of vertices in the graph.

Clearly, for $v = 1$ the result is true. Now suppose $v > 1$ and suppose $\Gamma$ is a tree. Then removing a leaf of $\Gamma$, together with the unique edge on this leaf, yields a tree $\Gamma'$ with $v - 1$ vertices. By induction $\Gamma'$ contains $v - 2$ edges. Thus $\Gamma$ contains $v - 1$ edges.

Now suppose $\Gamma$ contains $\leq v - 1$ edges. Since $|E| \cdot 2 = 2 \cdot (v - 1) = \sum_{v \in V} \deg(v)$, we find at least one vertex, $x$ say, of degree 1. Removing this vertex and the unique edge on it from $\Gamma$ yields a connected graph $\Gamma'$ with $v - 1$ vertices and $\leq v - 2$ edges. By induction, $\Gamma'$ is a tree with $v - 2$ edges. But then, since the vertex $x$ has degree 1 in $\Gamma$, we also find $\Gamma$ to be a tree with $v - 1$ edges. $\square$

A spanning tree of a graph $\Gamma$ is a subgraph of $\Gamma$ which is a tree, containing all vertices and some edges of $\Gamma$. So the problem of Example 3.4 is equivalent to finding a spanning tree in a graph. Now we describe two search algorithms that in fact construct spanning trees.

Algorithm 3.6 [Depth First Search and Breadth First Search] Consider a finite connected graph $\Gamma$. Fix a vertex $v$ of $\Gamma$ and label it by $\text{Label} := 1$. The vertex $v$ will be the root of a spanning tree in $\Gamma$. We construct this tree now as follows.

While there is a labeled vertex that has a neighbor not in the tree, find the vertex, say $w$ with the highest label that has neighbors not in the tree. Add the edge on $w$ and one of its neighbors not yet in the tree to the tree, and label this neighbor by \text{Label} := \text{Label} + 1.

This algorithm is called “Depth First Search”. “Breadth First Search” is a similar algorithm. However, here the while loop reads a little bit differently:

While there is a labeled vertex that has an unlabeled neighbor, find the vertex, say $w$ with the smallest label that has neighbors not in the tree. Add the edge on $w$ and one of its neighbors not yet in the tree to the tree, and label this neighbor by \text{Label} := \text{Label} + 1.

Clearly both algorithms stop after at most $|\Gamma|$ steps and do indeed yield a tree. Actually they both construct a spanning tree. To see this we have to show that every vertex of the graph gets a label. Suppose not, then by connectivity of the graph, there is a path starting with a labeled vertex $w$ and ending with an unlabeled vertex $u$. walking through this path we will find a labeled vertex adjacent to an unlabeled one. However, this means the while-loop is not finished yet.

Actually the above algorithms imply the following theorem.

Theorem 3.7 Let $\Gamma$ be a finite connected graph, then $\Gamma$ contains a spanning tree.
Exercise 3.8 Find spanning trees in the following graph using both “Depth first search” and “Breadth first search”.

Exercise 3.9 Suppose $T$ is a spanning tree in a finite connected graph $\Gamma$. If $e$ is an edge of $\Gamma$ which is not in $T$, then there is an edge $d$ of $T$, such that replacing $d$ by $e$ in $T$ yields again a spanning tree of $\Gamma$. Prove this.

3.4. Networks

Example 3.8 Consider the set $A$ of (big) cities in the Netherlands. As we have seen in 1.13, the relation “there is a direct train connection between $a$ and $b$” defines a binary relation $R$ on this set. The transitive closure of this relation tells us whether we can travel by train from one city to another. It does not tell us what the best or fastest way to travel is. Therefore we need some extra information on the relation. For example, we do not only want to know that there is a train connection from Eindhoven to Tilburg, but we also want to know how long it takes the train to get from Eindhoven to Tilburg. So, for each $(a,b) \in R$ we want to know the time it costs to travel from $a$ to $b$. Such extra information can be encode by a so-called “cost function”. Having this information it is natural to ask for the fastest connection between two cities, i.e., to find the shortest path from $a$ to $b$.

A graph $\Gamma = (V,E)$ together with a cost function $\text{cost}$ on the set $E$ of edges, i.e., a map from $E$ to (usually )$\mathbb{R}$ (or any other set) is called a network. In such a network the length of a path $(v_1,\ldots,v_n)$ equals to sum $\sum_{i=1}^{n-1}\text{cost}(\{v_i,v_{i+1}\})$. The following algorithm due to Dijkstra describes a way to find a shortest path in a network.

Algorithm 3.9 [Dijkstra’s shortest path algorithm] Suppose $\Gamma = (V,E)$ is a finite connected graph with cost function $\text{cost} : E \to \mathbb{R}^+$ a positive valued cost function.
Given two elements \( s \) and \( t \) of \( V \), a shortest (i.e., of minimal cost) path from \( s \) to \( t \) can be found as in the following algorithm.

In the process of the algorithm we will keep track of the sets Done and Remaider and some partial maps DefiniteDistance, EstimatedDistance and Predecessor.

Initiate the algorithm by setting \( \text{Done} := \{s\} \), \( \text{Remainder} := V \setminus \text{Done} \). The distances are set by \( \text{DefiniteDistance}(s) := 0 \) and \( \text{EstimatedDistance}(r) := \infty \) for all \( r \in \text{Remainder} \).

While \( \text{Remainder} \) contains the element \( t \), determine for all elements \( r \) in \( \text{Remainder} \) the \( \text{EstimatedDistance}(r) \), being the minimum of \( \text{DefiniteDistance}(d) + \text{cost}(d, r) \), where \( d \) runs through the set of elements \( n \in \text{Done} \) with \( \{n, r\} \in E \). Moreover, set \( \text{Predecessor}(r) \) to be one of the elements \( d \) for which this minimum is attained.

For those elements \( r \in \text{Remainder} \) for which \( \text{EstimatedDistance}(r) \) is minimal, we can decide that their distance to \( s \) is actually equal to \( \text{EstimatedDistance}(r) \). So, for those \( r \) we set \( \text{DefiniteDistance}(r) = \text{EstimatedDistance}(r) \), and we remove these points from \( \text{Remainder} \) and add them to \( \text{Done} \).

Since in each step of the While-loop at least one element is added to \( \text{Done} \), the algorithm will terminate and we will find the minimal length \( \text{DefiniteDistance}(t) \) of a path from \( s \) to \( t \). Moreover, with the help of \( \text{Predecessor} \) we even can find a path realizing this minimal length.

Sometimes one is not interested in finding a shortest path in a network, but a longest. Notice that this question only makes sense if we are dealing with a network without cycles, or more in particular, when we are dealing with a directed network without cycles. The following variation on Dijkstra’s shortest path algorithm yields a solution to that problem.

Algorithm 3.10 [Dijkstra’s longest path algorithm] First we notice that we may assume that there are no cycles in the network. For otherwise the problem does not make sense.

Again we set \( \text{Done} := \{s\} \), \( \text{Remainder} := V \setminus \text{Done} \). The distances, however, are set by \( \text{DefiniteDistance}(s) := 0 \) and \( \text{EstimatedDistance}(r) := 0 \) for all \( r \in \text{Remainder} \).

While \( \text{Remainder} \) contains the element \( t \), determine for all elements \( r \) in \( \text{Remainder} \) which can only be reached by an edge starting in \( \text{Done} \) the \( \text{DefiniteDistance}(r) \), being the maximum \( \text{DefiniteDistance}(d) + \text{cost}(d, r) \), where \( d \) runs through the set of elements \( n \in \text{Done} \) with \( \{n, r\} \in E \). Moreover, set \( \text{Predecessor}(r) \) to be one of the elements \( d \) for which this maximum is attained.

We remove these points from \( \text{Remainder} \) and add them to \( \text{Done} \).

Since in each step of the While-loop at least one element is added to \( \text{Done} \), then algorithm will terminate, and we will find the maximal length \( \text{DefiniteDistance}(t) \) of a path from \( s \) to \( t \). Moreover, with the help of \( \text{Predecessor} \) we even can find a path realizing this maximal length.
Algorithm 3.11 [Kruskal’s Greedy Algorithm] Suppose $\Gamma$ is a network. To find a minimal spanning tree, we first sort the edges with respect to their weight (or cost). We start with the minimal edge and each time we add an edge of minimal weight to the graph which does not close a cycle. The resulting subgraph will be a spanning tree of minimal weight.

Algorithm 3.12 [Prim’s Algorithm] In Kruskal’s Greedy Algorithm 3.11 we first have to sort the edges. The following algorithm avoid this. Start with a vertex $v$ and put it into the set $T$. At this vertex choose a neighbor outside $T$ on an edge with minimal weight. Add the neighbor and edge to the subgraph $T$. Now choose a minimal edge among the edges with a vertex in $T$ and one outside $T$ a vertex outside $T$ and add it, together with its end points to $T$. Repeat this procedure till we have all vertices in $T$.

Proposition 3.13 Kruskal’s Greedy Algorithm and Prim’s Algorithm provide us with a minimal spanning tree.

Proof. Let $T_m$ be the tree obtained after $m$ steps in Kruskal’s or Prim’s algorithm. With induction on $m$ we prove that there exists a minimal spanning tree $T$ containing $T_m$. In particular, taking $m$ to be the number of vertices in $\Gamma$, we find that the algorithms indeed yield a minimal spanning tree.

For $m = 1$ the graph $T_m$ is just one vertex and hence contained in any minimal spanning tree. Suppose $T_m \subseteq T$ for some minimal spanning tree $T$. Then in the next step of the algorithm we add an edge $\{v, w\}$ to $T_m$ to obtain $T_{m+1}$. If $\{v, w\}$ is also an edge of $T$, then clearly $T_{m+1} \subseteq T$. Thus assume $T$ does not contain $\{v, w\}$. This implies that $\{v, w\}$ with some vertices of $T$ forms a cycle $C$. Inside that cycle there is an edge $\{v', w'\} \neq \{v, w\}$ with $v' \in T_m$ but $w' \notin T_m$.

By the choice of $\{v, w\}$ in the algorithm, we have $\text{cost}(\{v, w\}) \leq \text{cost}(\{v', w'\})$. Hence, replacing the edge $\{v', w'\}$ in $T$ by $\{v, w\}$ we obtain again a spanning tree, $T'$ say, with $\text{cost}(T) \geq \text{cost}(T')$. In particular, $T'$ is also a minimal spanning tree containing $T_{m+1}$.

Exercise 3.10 Use Dijkstra’s algorithm to find both the shortest path from $s$ to $t$.

If we assume that one can only walk through the network from left to right, then what is the longest path from $s$ to $t$?

Exercise 3.11 Apply both Kruskal’s Greedy Algorithm and Prim’s Algorithm to find a minimal spanning tree in the following network.

Exercise 3.12 The diameter of a graph is the maximal distance between any of its vertices. Describe an algorithm that determines the diameter of a finite connected graph.
4. Order Relations

4.1. Partial Orders

Definition 4.1 A relation $\subseteq$ on a set $P$ is called an order if it is reflexive, antisymmetric and transitive. That means that for all $x, y$ and $z$ in $P$ we have:

- $x \subseteq x$;
- if $x \subseteq y$ and $y \subseteq x$, then $x = y$;
- if $x \subseteq y$ and $y \subseteq z$, then $x \subseteq z$.

The pair $(P, \subseteq)$ is called a partially ordered set, or for short, a poset.

Two elements $x$ and $y$ in a poset $(P, \subseteq)$ are called comparable if $x \subseteq y$ or $y \subseteq x$. The elements are called incomparable if $x \nsubseteq y$ and $y \nsubseteq x$.

If any two elements $x, y \in P$ are comparable, so we have $x \subseteq y$ or $y \subseteq x$, then the relation is called a linear order.

Example 4.2

- The identity relation $I$ on a set $P$ is an order.

  - On the set of real numbers $\mathbb{R}$ the relation $\leq$ is an order relation. For any two numbers $x, y \in \mathbb{R}$ we have $x \leq y$ or $y \leq x$. This makes $\leq$ into a linear order. Restriction of $\leq$ to any subset of $\mathbb{R}$ is again a linear order.

  - Let $P$ be the power set $\mathcal{P}(X)$ of a set $X$, i.e., the set of all subsets of $X$. Inclusion $\subseteq$ defines a partial order on $P$. This poset contains a smallest element $\emptyset$ and a largest element $X$. Clearly, $\subseteq$ defines a partial order on any subset of $P$.

  - The relation “Is a divisor of” $|\,$ defines an order on the set of natural numbers $\mathbb{N}$. We can associate this example to the previous one in the following way. For each $a \in \mathbb{N}$ denote by $D(a)$ the set of all divisors of $a$. Then we have

    $$a \mid b \Leftrightarrow D(a) \subseteq D(b).$$

  - On the set $P$ of partial functions from a set $A$ to a set $B$ we can define a partial order $\subseteq$ as follows. For $f, g \in P$ we have $f \subseteq g$ if and only if the graph of $f$ is contained in the graph of $g$, which means that $f \subseteq g$ where we consider $f$ and $g$ as subsets of $A \times B$.

  - On the set $P$ of partitions of a set $X$ we define the relation “refines” by the following. The partition $\Pi_1$ refines $\Pi_2$ if and only if each $\pi_1 \in \Pi_1$ is contained in some $\pi_2 \in \Pi_2$. The relation “refines” is a partial order on $P$.

    Notice, for the corresponding equivalence relations $R_{\Pi_1}$ and $R_{\Pi_2}$ we have $\Pi_1$ refines $\Pi_2$ if and only if $R_{\Pi_1} \subseteq R_{\Pi_2}$.
• If \( \subseteq \) is an order on a set \( P \), then \( \sqsubseteq \) also defines an order on \( P \). Here \( x \sqsupseteq y \) if and only if \( y \subseteq x \). The order \( \sqsubseteq \) is called the dual order of \( \subseteq \).

• Consider a finite tree and it a distinguished vertex \( r \) called the root of the tree. On the set of vertices of the tree we define the relation \( \subseteq \) as follows: \( x \subseteq y \) if and only if there is a path from \( r \) to \( y \) containing \( x \). This defines an order.

• Let \( V \) be a vector space. By \( P(V) \) we denote the set of all linear subspaces of \( V \). Containment \( \subseteq \) defines an order on \( P(V) \).

**Definition 4.3** If \( \subseteq \) is an order on the set \( P \), then the corresponding directed graph with vertex set \( P \) and edges \((x,y)\) where \( x \subseteq y \) is acyclic (i.e., contains no cycles).

If we want to draw a picture of the poset, we usually do not draw the whole graph. Instead we only draw an edge between \( x \) and \( y \) from \( P \) with \( x \subseteq y \) if there is no \( z \), distinct from both \( x \) and \( y \), for which we have \( x \subseteq z \) and \( z \subseteq y \). This graph is called the Hasse diagram for \((P,\subseteq)\). Usually pictures of Hasse diagrams are drawn in such a way that two vertices \( x \) and \( y \) with \( x \subseteq y \) are connected by an edge going upwards. For example the Hasse diagram for the poset \((\{1,2,3\},\subseteq)\) is drawn as below.

```
Definition 4.4 A chain of elements in \((P,\subseteq)\) is a sequence \((x_1,x_2,\ldots,x_n)\) of elements with \( x_i \neq x_{i+1} \) and \( x_i \subseteq x_{i+1} \). The elements in a chain are pairwise comparable. An antichain is a sequence \((x_1,x_2,\ldots,x_n)\) of pairwise incomparable elements.

An interval \([a,b]\), with \( a,b \in P \) is a the subset \( \{x \in P \mid a \subseteq x, x \subseteq b\} \). If, for \( a,b \in P \) with \( a \subseteq b \), the interval \([a,b]\) contains two points, then we call \( a \) and \( b \) neighbors. We say \( b \) covers \( a \) or \( a \) is covered by \( b \).
```
4.5 [New posets from old ones] There are various ways of constructing new posets out of old ones. We will discuss some of them. In the sequel both $P$ and $Q$ are posets with respect to some order, which we usually denote by $\sqsubseteq$, or, if confusion can arise, by $\sqsubseteq_P$ and $\sqsubseteq_Q$.

- If $P'$ is a subset of $P$, then $P'$ is also a poset with order $\sqsubseteq$ restricted to $P'$. This order is called the induced order on $P'$.
- $\sqsupset$ induces the dual order on $P$.
- Let $S$ be some set. On the set of maps from $S$ to $P$ we can define an ordering as follows. Let $f : S \to P$ and $g : S \to P$, then we define $f \sqsubseteq g$ if and only if $f(s) \sqsubseteq g(s)$ for all $s \in S$.
- On the cartesian product $P \times Q$ we can define an order as follows. For $(p_1, q_1), (p_2, q_2) \in P \times Q$ we define $(p_1, q_1) \sqsubseteq (p_2, q_2)$ if and only if $p_1 \sqsubseteq p_2$ and $q_1 \sqsubseteq q_2$.
- A second ordering on $P \times Q$ can be obtained by the following rule. For $(p_1, q_1), (p_2, q_2) \in P \times Q$ we define $(p_1, q_1) \sqsubseteq (p_2, q_2)$ if and only if $p_1 \sqsubseteq p_2$ and $p_1 \neq p_2$ or if $p_1 = p_2$ and $q_1 \sqsubseteq q_2$. This order is called the lexicographic order on $P \times Q$.

Of course we can now extend this inductively to direct products of more than two sets.

**Exercise 4.1** Let $|$ denote the relation “is a divisor of” defined on $\mathbb{Z}$. Even if we let 0 be a divisor of 0, then this does not define an order on $\mathbb{Z}$. Prove this.

**Exercise 4.2** Let $|$ denote the relation “is a divisor of”. This relation defines an order on the set $D = \{1, 2, 3, 5, 6, 10, 15, 30\}$ of divisors of 30. Draw the Hasse diagram.

Draw also the Hasse diagram of the poset of all subsets of $\{2, 3, 5\}$. Compare the two diagrams. What do you notice?

**Exercise 4.3** Let $\sqsubseteq$ denote an order relation on a finite set $P$. By $H$ we denote the relation defining adjacency in the Hasse diagram of $\sqsubseteq$. Prove that $\sqsubseteq$ is the transitive reflexive closure of $H$.

**Exercise 4.4** Let $m, n \in \mathbb{N}$. By $\Pi_m$ we denote the partition of $\mathbb{Z}$ into equivalence classes modulo $m$. What is a necessary and sufficient condition on $n$ and $m$ for $\Pi_m$ to be a refinement of $\Pi_n$.

**Exercise 4.5** Suppose $\sqsubseteq_P$ and $\sqsubseteq_Q$ are linear orders on $P$ and $Q$, respectively. Show that the lexicographical order on $P \times Q$ is also linear.

**Exercise 4.6** Show that the relations as defined in 4.5 are indeed orders.
4.2. Lattices

**Definition 4.6** Let \((P, \sqsubseteq)\) be a partially order set and \(A \subseteq P\) a subset of \(P\). An element \(a \in A\) is called the *largest element* or *maximum* of \(A\), if for all \(a' \in A\) we have \(a' \sqsubseteq a\). Notice that a maximum is unique, see Lemma 4.9 below.

An element \(a \in A\) is called *maximal* if for all \(a' \in A\) we have that either \(a' \sqsubseteq a\) or \(a\) and \(a'\) are incomparable.

Similarly we can define the notion of *smallest element* or *minimum* and *minimal element*.

If the poset \((P, \sqsubseteq)\) has a maximum, then this is often denoted as \(\top\) (top). A smallest element is denoted by \(\bot\) (bottom).

If a poset \((P, \sqsubseteq)\) has a minimum \(\bot\), then the minimal elements of \(P \setminus \{\bot\}\) are called the *atoms* of \(P\).

**Example 4.7**
- If we consider the poset of all subsets of a set \(S\), then the empty set \(\emptyset\) is the minimum of the poset, whereas the whole set \(S\) is the maximum. The atoms are the subsets of \(S\) containing just a single element.
- In the poset \((\mathbb{N}, |)\) there is no maximum. However, the minimum equals 1. The atoms are the prime numbers.
- In the poset \(P(V)\) of linear subspaces of a vector space \(V\), the trivial subspaces \(\{0\}\) and \(V\) are the minimum and maximum, respectively. The 1-dimensional subspaces are the atoms.

**Example 4.8** Notice, maximal elements need not be maxima. Indeed, if we consider the poset of partial maps from a set \(A\) to a set \(B\) where \(f \subseteq g\) if and only if the graph of \(f\) is contained in the graph of \(g\), see 4.2, then all maps from \(A\) to \(B\) are maximal. But (at least if \(B\) contains two or more elements) none of the maps is a maximum.

**Lemma 4.9** Let \((P, \sqsubseteq)\) be a partially order set. Then \(P\) contains at most one maximum and one minimum.

**Proof.** Suppose \(p, q \in P\) are maxima. Then \(p \sqsubseteq q\) as \(q\) is a maximum. Similarly \(q \sqsubseteq p\) as \(p\) is a maximum. But then by antisymmetry of \(\sqsubseteq\) we have \(p = q\). □

**Lemma 4.10** Let \((P, \sqsubseteq)\) be a finite poset. Then \(P\) contains a minimal and a maximal element.

**Proof.** Consider the directed graph associated to \((P, \sqsubseteq)\) and pick a vertex in this graph. If this vertex is not maximal, then there is an edge leaving it. Move along this edge to the neighbor. Repeat this as long as no maximal element is found. Since the
graph contains no cycles, we will never meet a vertex twice. Hence, as \( P \) is finite, the procedure has to stop. This implies we have found a maximal element.

A minimal element of \((P, \sqsubseteq)\) is a maximal element of \((P, \sqsupseteq)\) and thus exists also.

\(\Box\)

**Example 4.11** Notice that minimal elements and maximal elements are not necessarily unique. In fact, they do not even have to exist. In \((\mathbb{R}, \leq)\) for example, there is no maximal nor a minimal element.

**Algorithm 4.12** [Topological sorting] Given a finite poset \((P, \sqsubseteq)\), we want to sort the elements of \(P\) in such a way that an element \(x\) comes before an element \(y\) if \(x \sqsubseteq y\). This is called topological sorting. In other words, topological sorting is finding a map \(\text{ord}: P \to \{1, \ldots, n\}\), where \(n = |P|\), such that for distinct \(x\) and \(y\) we have that \(x \sqsubseteq y\) implies \(\text{ord}(x) < \text{ord}(y)\). We present an algorithm for topological sorting.

Suppose we are given a finite poset \((P, \sqsubseteq)\), then for each element \(p\) in \(P\) we determine the indegree, i.e., the number of elements \(q\) with \(q \sqsubseteq p\). While there are vertices in \(P\) with indegree 0, pick one of them, say \(q\), and set \(\text{ord}(q)\) to be the smallest value in \(\{1, \ldots, n\}\) which is not yet an image of some point. Now remove \(q\) from \(P\) and lower all the indegrees of the neighbors of \(q\) by 1.

Notice that, by Lemma 4.10, we will always find elements in \(P\) with indegree 0, unless \(P\) is empty.

**Example 4.13** Topological sort has various applications. For example consider a spreadsheet. In a spreadsheet various tasks depend on each other. In particular, some of the computations need input from other computations and therefore they can only be carried out after completion of the other computations. This puts a partial order on the set of tasks within a spreadsheet. By topological sort the task list can be linearized and the computations can be done in a linear order.

**Definition 4.14** Let \((P, \sqsubseteq)\) be a partially order set and \(A \subseteq P\) a subset of \(P\). An element \(p \in P\) is called an upper bound for \(A\) if \(a \sqsubseteq p\) for all \(a \in A\). It is called a lower bound for \(A\) if \(p \sqsubseteq a\) for all \(a \in A\). If the set of all upper bounds of \(A\) has a smallest element, then this element is called the supremum or least upper bound of \(A\). If it exists, it is denoted by \(\text{sup } A\).

Similarly the largest lower bound of \(A\) (if it exists) is called the infimum or greatest lower bound of \(A\) and is denoted by \(\text{inf } A\).

**Example 4.15**

- If \(S\) is a set and \(P = \mathcal{P}(S)\) the poset of all subsets of \(S\) with relation \(\subseteq\), then for any subset \(X\) of \(P\) the supremum \(\text{sup } X\) equals \(\bigcup_{x \in X} x\) and the infimum \(\text{inf } X\) equals \(\bigcap_{x \in X} x\).
• Consider the set \( \mathbb{N} \) of natural numbers with order relation \( | \), “is a divisor of”. Then the supremum of two elements \( a, b \in \mathbb{N} \) equals \( \text{lcm}(a, b) \). The greatest common divisor \( \text{gcd}(a, b) \) is the infimum of \( \{a, b\} \).

• Suppose \( X \) and \( Y \) are two sets and \( P \) is the set of all partial maps from \( X \) to \( Y \). We consider the order \( \sqsubseteq \) on \( P \) as in Example 4.2. If \( f \) and \( g \) are two distinct maps from \( X \) to \( Y \), then there is no supremum of \( f \) and \( g \), as there is no partial map having the same graph as both \( f \) and \( g \). But the infimum of \( f \) and \( g \) is the partial map whose graph is the intersection of the graph of \( f \) and \( g \).

**Definition 4.16** A poset \((P, \sqsubseteq)\) is called a lattice, if for all \( x, y \in P \) the subset \( \{x, y\} \) of \( P \) has a supremum and an infimum. The supremum of \( x \) and \( y \) is denoted by \( x \sqcup y \) and the infimum as \( x \sqcap y \).

**Example 4.17** Here are some examples of lattices we already encountered before.

• \((\mathbb{R}, \leq)\) is a lattice. If \( x, y \in \mathbb{R} \), then \( \sup\{x, y\} = \max\{x, y\} \) and \( \inf\{x, y\} = \min\{x, y\} \).

• If \( S \) is a set and \( P = \mathcal{P}(S) \) the poset of all subsets of \( S \) with relation \( \subseteq \), then \( P \) is a lattice with \( \sqcap = \cap \) and \( \sqcup = \cup \).

• The poset \((\mathbb{N}, |)\) of natural numbers with order relation \( | \) is a lattice with the least common multiple as \( \sqcup \) and the greatest common divisor as \( \sqcap \).

**Theorem 4.18** Let \((L, \sqsubseteq)\) be a lattice. Then for all \( x, y, z \in L \) we have the following:

i. \( x \sqcup x = x \) and \( x \sqcap x = x \);

ii. \( x \sqcup y = y \sqcup x \) and \( x \sqcap y = y \sqcap x \);

iii. \( x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z \) and \( x \sqcap (y \sqcap z) = (x \sqcap y) \sqcap z \);

iv. \( x \sqcup (x \sqcap y) = x \sqcap (x \sqcup y) = x \).

**Proof.** The first two statements are obvious. To prove the third, we will show that \( x \sqcup (y \sqcup z) \) is the supremum of the set \( \{x, y, z\} \). Clearly, \( x \sqcup (y \sqcup z) \) is an upper bound for \( x \) and \( y \sqcup z \) and thus also for \( y \) and \( z \). Any other upper bound \( u \) of \( x \), \( y \) and \( z \), is also an upper bound of \( x \) and \( y \sqcup z \) and therefore an upper bound for \( x \sqcup (y \sqcup z) \). This implies that \( x \sqcup (y \sqcup z) \) is the smallest upper bound of \( x \), \( y \) and \( z \). But so is also \( (x \sqcup y) \sqcup z \) and we obtain the first statement of (iii). The second equation follows from the first by considering the dual poset.

It remains to prove (iv). Clearly \( x \) is an upper bound of \( x \sqcap y \) and \( x \sqcap y \). For any other upper bound \( u \) of \( x \) and \( x \sqcap y \) we clearly have \( x \sqsubseteq u \). Thus \( x \) equals the smallest common upper bound \( x \sqcup (x \sqcap y) \) of \( x \) and \( x \sqcap y \).

Dually we obtain \( x \sqcap (x \sqcup y) = x \).

\(\Box\)
Remark 4.19 Actually, if \( L \) is a set with two operations \( \sqcup \) and \( \sqcap \) from \( L \times L \to L \) satisfying (i) upt to (iv) of the previous Theorem, then the relation \( \sqsubseteq \) given by \( x \sqsubseteq y \) if and only if \( x = x \sqcap y \) defines a lattice structure on \( L \).

The standard example of a lattice is the lattice of all subsets of a set \( V \). All four properties listed above can easily be checked in this particular example. However not all laws carry over. The so-called distributive laws
\[
A \cap (B \cup C) = (A \cap B) \cup (A \cap C)
\]
and
\[
A \cup (B \cap C) = (A \cup B) \cap (A \cup C)
\]
for all \( A, B, C \subset V \), do not hold in general. A weaker version, however, is valid:

**Theorem 4.20** Let \((L, \sqsubseteq)\) be a lattice. Then for all \( x, y, z \in L \) we have the following:
\[
x \sqcup (y \cap z) \sqsubseteq (x \sqcup y) \cap (x \sqcup z)
\]
and dually,
\[
x \sqcap (y \sqcup z) \supseteq (x \sqcap y) \cup (x \sqcap z).
\]

**Proof.** Let \( x, y, z \in L \). Then we have that \( x \sqcup y \) is an upper bound for \( x \) and for \( y \). Furthermore, as \( y \cap z \sqsubseteq y \), we find that \( y \) is an upper bound for \( y \cap z \). Thus \( x \sqcup y \) is an upper bound for both \( x \) and \( y \cap z \). Hence we have \( x \sqcup (y \cap z) \subseteq x \sqcup y \). Similarly we find that \( x \sqcup (y \cap z) \subseteq x \sqcup z \). But then we find \( x \sqcup (y \cap z) \subseteq (x \sqcup y) \cap (x \sqcup z) \).

The second statement is just the dual of the first. \( \square \)

**Definition 4.21** Lattices in which we have for all \( x, y, z \)
\[
x \sqcup (y \cap z) = (x \sqcup y) \cap (x \sqcup z)
\]
and dually,
\[
x \sqcap (y \sqcup z) = (x \sqcap y) \cup (x \sqcap z)
\]
are called **distributive lattices**.

**Example 4.22** Consider the poset \((P(V), \subseteq)\) of all linear subspaces of a vector space \( V \). Let \( X, Y, Z \) be three distinct 1-dimensional subspaces inside a 2-dimensional subspace \( W \). Then \( X \sqcup (Y \cap Z) \) equals \( X \) whereas \((X \sqcup Y) \cap (X \sqcup Z) = W \cap W = W \). So, we have encountered a nondistributive lattice. (See Exercise 4.10.)

**Definition 4.23** A partial order in which every chain \( p_1 \sqsubseteq p_2 \sqsubseteq \ldots \) has a supremum is called a **complete partial order** (CPO).

A lattice in which every subset has a supremum and infimum, is called a **complete lattice**. Notice that a complete lattice has a maximal element.
Example 4.24 Every finite poset or lattice is complete.

Example 4.25 Consider the poset \((\mathcal{P}(V), \subseteq)\) of all subsets of a set \(V\). Then this poset is a lattice. It is also complete. Indeed, for any subset \(C\) of \(\mathcal{P}(V)\) the supremum is the union \(\bigcup_{c \in C} c\).

Example 4.26 Consider the poset of partial functions from a set \(A\) to a set \(B\). This poset is complete. Indeed, if we have a chain \(f_1 \subseteq f_2 \subseteq \ldots\), then this is a chain of subsets of \(A \times B\) with respect to inclusion. Hence \(f = \bigcup_{i \in \mathbb{N}} f_i\) is the supremum of the chain.

Example 4.27 The poset \((\mathbb{R}, \leq)\) is a lattice, but it is not complete. However, every interval \([a, b]\) with \(a < b\) is a complete lattice.

Theorem 4.28 Let \((P, \sqsubseteq)\) be a poset in which every subset has an infimum. Then every subset has also a supremum. In particular, \((P, \sqsubseteq)\) is a complete lattice.

Proof. Let \(A\) be a subset of \(P\). By \(B\) we denote the set of all upper bounds of \(A\). The infimum \(b\) of \(B\) is an upper bound of \(A\). Indeed, every element \(a\) is a lower bound for all the elements in \(B\) and thus also for the infimum of \(B\). In particular, we find \(b \in B\). The element \(b\) is the supremum of \(A\). \(\square\)

Exercise 4.7 Let \((P, \sqsubseteq)\) be a poset and \(A\) a subset of \(P\). Prove that an element \(a \in A\) is maximal if and only if for all \(x \in A\) we have \(a \sqsubseteq x\) implies \(a = x\).

Exercise 4.8 Let \((L, \sqsubseteq)\) be a lattice and \(x, y, z \in L\). Prove that

i. \(x \sqsubseteq y\) implies \(x \sqcup z \sqsubseteq y \sqcup z\).

ii. \(x \sqsubseteq z\) implies \(z \sqcup (x \sqcap y) = z\).

Exercise 4.9 In the figure below you see three diagrams. Which of these diagrams are Hasse diagrams? Which of these diagrams represents a lattice?

Exercise 4.10 Let \(V\) be a vector space. Show that the poset \((\mathcal{P}(V), \subseteq)\) is a complete lattice.

Exercise 4.11 Consider the poset of partial functions from a set \(A\) to a set \(B\) as in 4.26. This is a complete poset. Prove this.

Exercise 4.12 Is the poset \(\{\{1, 2, 3, 4, \ldots\}, |\) a complete lattice? How about \(\{0, 1, 2, 3, 4, \ldots\}, |\)?
Exercise 4.13 Suppose \((L, \sqsubseteq)\) is a lattice and \(a, b \in L\) with \(a \sqsubseteq b\). Prove that \(\sqsubseteq\) induces a lattice on the interval \([a, b]\).

Exercise 4.14 Let \((L, \sqsubseteq)\) be a lattice. If for all \(x, y, z \in L\) we have
\[
x \sqcup (y \cap z) = (x \sqcup y) \cap (x \sqcup z)
\]
then we also have for all \(x, y, z \in L\) that
\[
x \cap (y \sqcup z) = (x \cap y) \sqcup (x \cap z).
\]
Prove this. (Hint: use \(x \cap (y \sqcup z) = (x \cap (y \sqcup x)) \cap (y \sqcup z)\).)

5. Boolean Algebras

In the previous section we have encountered the operations \(\cap\) and \(\cup\) in a lattice \((L, \sqsubseteq)\). In this section we will have a closer look at these operations.

5.1. Definitions

We start with some examples.

Example 5.1 Let \(\Omega\) be a set and \(\mathcal{P}(\Omega)\) the set of all subsets of \(\Omega\). Then the intersection \(\cap\) and the union \(\cup\) are two binary operations on \(\mathcal{P}(\Omega)\). These operations \(\cap\) and \(\cup\) satisfy the following laws, where \(X, Y\) and \(Z\) are elements from \(\mathcal{P}(\Omega)\):

- commutativity: \(X \cap Y = Y \cap X\) en \(X \cup Y = Y \cup X\);
- associativity: \(X \cap (Y \cap Z) = (X \cap Y) \cap Z\) en \(X \cup (Y \cup Z) = (X \cup Y) \cup Z\);
- absorption: \(X \cup (X \cap Y) = X\) en \(X \cap (X \cup Y) = X\);
- distributivity: \(X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)\) and \(X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)\).
The empty set $\emptyset$ and complete set $\Omega$ are special elements with respect to these operations $\cap$ en $\cup$. For all $X \in \mathcal{P}(\Omega)$ we have:

- $X \cap \emptyset = \emptyset$;
- $X \cup \Omega = \Omega$.

Finally, for each element $X$ from $\mathcal{P}(\Omega)$ there exists a complement $X^c$ with:

- $X \cap X^c = \emptyset$ and $X \cup X^c = \Omega$.

**Example 5.2** For $n \in \mathbb{N}$ denote by $D(n)$ the set of positive divisors of $n$. Suppose $n$ is a product of distinct primes.

On $D(n)$ we can consider the operations $\gcd$ and $\lcm$. These operations are commutative, associative en distributive. Also the absorption laws hold, for $\lcm(x, \gcd(x, y)) = x$ and $\gcd(x, \lcm(x, y)) = x$. Special elements for these operations are 1 for $\gcd$ and $n$ for $\lcm$: $\gcd(1, x) = 1$ en $\lcm(x, n) = n$. Finally, to each $x \in D(n)$ we can associate a complement: $n/x$. For this complement we have: $\gcd(x, n/x) = 1$ and $\lcm(x, n/x) = n$.

Thus $D(n)$ together with the operations $\gcd$, $\lcm$, 1, $n$ and $x \mapsto n/x$ satisfies the same 7 laws as in the previous example.

**Definition 5.3** A set $V$ together with the binary operations $\sqcap$ en $\sqcup$, special elements $\bot$ and $\top$, called the bottom and top, respectively, and a unary operation $X \mapsto X^c$ (the complement of $v$) satisfying for all $X, Y \in V$:

- $\sqcap$ and $\sqcup$ are commutative, associative and distributive;
- the absorption laws: $X \sqcap (X \sqcap Y) = X$ and $X \sqcap (X \sqcup Y) = X$;
- $X \sqcap \bot = \bot$ en $X \sqcup \top = \top$;
- $X \sqcup X^c = \top$ en $X \sqcap X^c = \bot$,

is called a Boolean algebra.

**Remark 5.4** Boolean algebras are named after the British mathematician George Boole (1815–1864), who was the first to investigate the algebraic laws for the operations $\cup$ en $\cap$ on sets.

**Example 5.5** Let $V$ be the set of sequences of zeros and ones of length $n$. On $V$ we define an addition $\min$ and multiplication $\max$ as follows:

$$(a_1, \ldots, a_n)\min(b_1, \ldots, b_n) = (\min(a_1, b_1), \ldots, \min(a_n, b_n)).$$

$$(a_1, \ldots, a_n)\max(b_1, \ldots, b_n) = (\max(a_1, b_1), \ldots, \max(a_n, b_n)).$$
As special elements we have \( \bot = (0, \ldots, 0) \) and \( \top = (1, \ldots, 1) \). For each \( v \in V \) the element \( v^c \) is defined as the unique element with at coordinate \( i \) a 0, if and only if the coordinate \( i \) of \( v \) has value 1. These operation and elements turn \( V \) into a Boolean algebra.

The above example and Example 5.1 do not really differ. For, if \( \Omega \) is a set of \( n \) elements, then we can fix and order \( \omega_1, \ldots, \omega_n \) on these elements of \( \Omega \), and every subset \( X \) of \( \Omega \) can be identified with the sequence of zeros and ones with a one at position \( i \) if and only if \( \omega_i \in X \).

Addition \( \min \) corresponds to taking intersections, and multiplication \( \max \) with taking unions.

**Example 5.6** Suppose \( \Omega \) is an infinite set and \( B \) the Boolean algebra of subsets of \( \Omega \) as described in 5.1. By \( F \) we denote the set of subsets of \( \Omega \) with finitely many points or missing finitely many points form \( \Omega \). Then \( F \) contains the empty set \( \emptyset \) as well as the full set \( \Omega \). Moreover, \( F \) is closed under \( \cup \), \( \cap \) and \( ^c \). For, if \( X, Y \in F \) then also \( X \cup Y, X \cap Y \) and \( X^c \) are in \( F \). (Prove this!) The set \( F \) together with the elements \( \emptyset \) and \( \Omega \), and the operations \( \cup, \cap \) and \( ^c \) is also a Boolean algebra.

**Lemma 5.7** Let \((V, \cap, \cup, \bot, \top, v \mapsto v^c)\) be a Boolean algebra. Then \( V \) contains a unique bottom and a unique top element. Moreover, for each \( v \in V \) the complement \( v^c \) is the unique element \( w \) in \( V \) satisfying \( v \cup w = \top \) and \( v \cap w = \bot \).

**Proof.** Let \( v \) be an element in \( V \) with \( v \cap w = v \) for all \( w \in V \). Then we have \( v = v \cap \bot = \bot \). Thus the bottom element \( \bot \) is unique.

If \( v \) is an element in \( V \) with \( v \cup w = v \) for all \( w \in V \). Then we have \( v = v \cup \top = \top \). Thus also the top element \( \top \) is unique.

To prove uniqueness of the complement we first notice that for all \( v \in V \) we have:

\[
v \cup \bot = v \cup (v \cap \bot) = v, \quad v \cap \top = v \cap (v \cup \top) = v.
\]

Suppose, to prove uniqueness of the complement of \( v \), that \( v \cap w = \bot \) and \( v \cup w = \top \) for some element \( w \in V \). Then the above implies:

\[
w = v \cap \top = v \cap (v \cup v^c) = (w \cap v) \cup (w \cap v^c) = \bot \cup (w \cap v^c) = (v \cap v^c) \cap (w \cap v^c) = (v \cup w) \cap v^c = \top \cap v^c = v^c,
\]

and thus \( w = v^c \). \( \square \)
Exercise 5.1 Show that in a Boolean algebra we have:
\[ \bot^c = \top \quad \text{and} \quad \top^c = \bot. \]

Exercise 5.2 Suppose \( V \) with the operations \( \cap, \cup, \bot, \top \) and \( v \mapsto v^c \) is a Boolean algebra. Show that for all \( v \) and \( w \) in \( V \) we have:

i. \( (v^c)^c = v; \)

ii. \( (v \cap w)^c = v^c \cup w^c; \)

iii. \( (v \cup w)^c = v^c \cap w^c. \)

The last two rules are called the rules of De Morgan.

Exercise 5.3 Let \( n \in \mathbb{N} \) and denote by \( D(n) \) the set of divisors of \( n \). On \( D(n) \) we consider the operations as defined in Example 5.2. Does this always turn \( D(n) \) into a Boolean algebra? Give necessary and sufficient conditions on \( n \) to turn \( D(n) \) into a Boolean algebra.

Exercise 5.4 Prove that there is no Boolean algebra on three elements.

5.2. Examples from Logic and Electrical Engineering

Example 5.8 In logic we often make use of the following truth table:

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X &amp; Y</th>
<th>X \lor Y</th>
<th>\neg X</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>

the symbols \( X \) and \( Y \) can be interpreted as maps on a (not further specified) set taking the values \( True \) or \( False \). The first two columns give the various values \( X \) and \( Y \) can take.

From this table we easily check that \( \land, \lor \) and \( \neg \) define a Boolean algebra on the set \( \{False, True\}. \)

Example 5.9 In the design of electrical circuits one also often uses three basic gates, a NOT-gate, an AND-gate and an OR-gate, see [1].

The NOT-gate gives a signal if there is no input signal, and gives no signal if there is an input signal. The AND-gate accepts two signal and returns only a signal if there are indeed two signals coming in. The OR-gate accepts two signal and returns a signal if there is at least one signal coming in. Several of these gates are set together to
form the electrical circuit. In such a circuit we have \( n \) input channels and one output channel. We denote a signal at a channel by 1 and no signal by 0. The input is then a sequence of \( n \) zeros or ones. The output is just 1 or 0. We regard such a circuit as a map from the \( n \) input channels to the output channel, or from the set of 0, 1-sequences of length \( n \) to \( \{0, 1\} \).

For example, if we consider the a circuit with five input channels \( A, B, C, D \) and \( E \) as in the figure below, then we can think of \( A \) to represent the function which is projection on the \( A^{th} \)-coordinate, \( B \) is projection on the \( B^{th} \)-coordinate, etc. The AND-gate corresponds to taking the minimum, the OR-gate to taking the maximum. So \( A \) and \( B \) is minimum of the functions \( A \) and \( B \), and \( A \) or \( B \) is the maximum of the functions \( A \) and \( B \). The NOT-gate changes the value of a map from 1 to 0 or from 0 to 1.

The circuit displayed in the figure corresponds to the map

\[
\text{(not (A and B) or (C or D)) and (not E)}.
\]

The AND-gate, OR-gate and NOT-gate turn the set of electrical circuits into a Boolean algebra.

**Exercise 5.5** What is \( \bot \) and what is \( \top \) in the Boolean algebra of Example 5.8? The map \( X \land Y \) evaluates only in one case to \( True \). Construct three other maps that also evaluate only ones to \( True \).

**Exercise 5.6** What is \( \bot \) and what is \( \top \) in the Boolean algebra of Example 5.9? Can you prove that the AND-gate, OR-gate and NOT-gate turn the set of electrical circuits indeed into a Boolean algebra?
Exercise 5.7 The following electrical circuit represents the so-called Hotel switch. What does it do? Why is it called Hotel switch? Can you find another element in the Boolean algebra of electrical circuits representing this Hotel switch?

5.3. The Structure of Boolean Algebras

Proposition 5.10 Let $B = (V, \cap, \cup, \bot, \top, v \mapsto v^c)$ be a Boolean algebra. For $x, y \in V$ the following three statements are equivalent.

i. $x \cap y = x$;

ii. $x \cup y = y$;

iii. $\exists z \in V : y = x \cup z$.

Proof. (i)$\Rightarrow$(iii). Suppose (i). Let $z = y$. Then

$$x \cup z = (y \cap x) \cup (y \cap \top) = y \cap (x \cup \top) = y \cap \top = y,$$

which proves (iii).

(iii)$\Rightarrow$(ii). Suppose (iii). Then with $z$ as in (iii):

$$y \cup x = x \cup z \cup x = x \cup z = y.$$

Hence (ii).

(ii)$\Rightarrow$(i). If (ii) holds, then, by the second absorption law,

$$y \cap x = (y \cup x) \cap x = x,$$

from which we deduce (i). $\square$

Definition 5.11 Let $B = (V, \cap, \cup, \bot, \top, v \mapsto v^c)$ be a Boolean algebra. Define the relation $\subseteq$ on $V$ by

$$x \subseteq y \iff x \cap y = x$$

for $x, y \in V$. (One of the three equivalent statements from 5.10.)
Lemma 5.12 Let $B = (V, \cap, \cup, \bot, \top, v \mapsto v^c)$ be a Boolean algebra. Then $\sqsubseteq$ defines a partial order on $V$ with minimum $\bot$ and maximum $\top$.

Proof. Let $x, y, z \in V$.

$\sqsubseteq$ is reflexive: Indeed, since $x = x \cap x$, we have $x \sqsubseteq x$.

$\sqsubseteq$ is antisymmetric: Suppose $x \sqsubseteq y \sqsubseteq x$. Then $x = x \cap y = y$, so $x = y$.

$\sqsubseteq$ is transitive: Suppose $x \sqsubseteq y \sqsubseteq z$. Then we have $x = y \cap x = (z \cap y) \cap x = z \cap (y \cap x) = z \cap x$, and hence $x \sqsubseteq z$.

We have proved $\sqsubseteq$ to be reflexive, antisymmetric and transitive. Hence it is a partial order.

Finally, since $\bot = \bot \cap x$ en $\top = \top \cup x$ (see 5.10), we find $\bot$ and $\top$ to be the minimum and maximum of $(V, \sqsubseteq)$. □

Definition 5.13 An element $x$ of a Boolean algebra is called an atom if there is no other element $y$ than $\bot$ and $x$ itself with $y \sqsubseteq x$.

By Proposition 5.10 an element $x$ different from $\bot$ in a Boolean algebra is an atom if and only if for all $y \neq \bot$ we have: if $x \cap y = y$, then $y = x$. If $a_1, \ldots, a_m$ are elements in a Boolean algebra, then

$$a_1 \cup \cdots \cup a_m$$

is well defined ($\cup$ is associative). Moreover, this expression is independent of the order of the $a_i$ ($\cup$ is commutative) and finally it is independent of the multiplicity of the $a_i$ (for, $a \cup a = a$). Therefore, we can express this the element $a_1 \cup \cdots \cup a_m$ also by

$$\bigsqcup_{1 \leq i \leq m} a_i.$$  

Similarly, also

$$a_1 \cap \cdots \cap a_m$$

is well defined, and we write it as

$$\bigcap_{1 \leq i \leq m} a_i.$$  

Theorem 5.14 [Representation Theorem of Finite Boolean Algebras] Let $B = (V, \cap, \cup, \bot, \top, v \mapsto v^c)$ be a finite Boolean algebra. Suppose $A$ is the set of atoms in $B$. Then every element $b \in V$ can be written as

$$b = \bigsqcup_{a \in A, a \sqsubseteq b} a.$$  

Moreover, this expression is unique, i.e., if $b = \bigsqcup_{a \in A'} a$ for some subset $A'$ of $A$, then $A' = \{ a \in A, a \sqsubseteq b \}$.  

37
Proof. Suppose the right hand side of
\[ b = \bigcup_{a \in A, a \sqsubseteq b} a \]
equals \( w \). Then by 5.10 we have \( w \sqsubseteq b \). Therefore
\[ b = (b \cap w) \sqcup (b \cap w^c) = w \sqcup (b \cap w^c). \]
We will show \( b \cap w^c = \perp \). For, then we have \( b = w \sqcup \perp = w \), and we are done.

If \( \alpha \in A \) satisfies \( \alpha \sqsubseteq b \cap w^c \), then
\[ \alpha \sqsubseteq b \] and \( \alpha \sqsubseteq \bigcap_{a \in A, a \sqsubseteq b} a^c. \]
In particular, \( \alpha \sqsubseteq \alpha^c \), so \( \alpha = \alpha \cap \alpha^c = \perp \). A contradiction. This implies that no element from \( A \) is less than or equal to \( b \cap w^c \). Since \( V \) is finite, there is for each element \( v \neq \perp \) an atom \( a \) with \( a \sqsubseteq v \): if \( v \) is not an atom, then there is an element \( v_1 \neq \perp \) with \( v_1 \sqsubseteq v \). Then \( v_1 \) is an atom, or there is an element \( v_2 \neq \perp \) with \( v_2 \sqsubseteq v_1 \). Etc. Conclusion: \( b \cap w^c = 0 \).

Remains to prove the uniqueness of this expression for \( b \). Suppose that \( b = \bigcup_{a \in A'} a \) for some subset \( A' \) of \( A \). Then clearly \( A' \subseteq \{ a \in A, a \sqsubseteq b \} \). If \( a_0 \in \{ a \in A \mid a \sqsubseteq b \} \setminus A' \), then
\[ a_0 = a_0 \cap b = a_0 \cap (\bigcup_{a \in A'} a) = \bigcup_{a \in A'} (a_0 \cap a) = \bigcup_{a \in A'} \perp = \perp. \]
A contradiction. \( \Box \)

The above theorem states that a finite Boolean algebra \( B \) can be identified with the Boolean algebra of subsets of \( A \), the set of atoms of \( B \). In the next section we will make this statement more precise.

Notice that the above result is not true for infinite Boolean algebras. Indeed, the Example 5.6 shows that an infinite Boolean algebra can not always be identified with the set of subsets of its atoms.

Example 5.15 Consider the Boolean algebra of electrical circuits with \( n \) input channels as described in Example 5.9. Then the AND-gate corresponds to \( \sqcap \) and the OR-gate to \( \sqcup \). In terms of maps from the set of 0,1-sequences of length \( n \) to \( \{0,1\} \), AND corresponds to taking the minimum of two functions. Hence, the map which sends everything to 0 is the minimal element \( \perp \) in this Boolean algebra. The atoms are the elements that take the value 1 only ones. Hence there are just as many atoms as there 0,1-sequences and that number is \( 2^n \). Now Theorem 5.14 implies that there are exactly \( 2 \times 2^n \) different elements in this Boolean algebra. But as there are also exactly \( 2 \times 2^n \) different maps from the set of 0,1-sequences to \( \{0,1\} \), this implies that each possible map can be realized by a circuit.
It remains the problem to find for a given map a circuit that realizes the map. Or even better, a small circuit doing so. This is still a difficult task. But our translation of the problem into a problem of Boolean algebras makes it easier to solve this question with the help of the computational power of computers.

**Exercise 5.8** Determine the atoms of all the Boolean algebras from 5.2, 5.1, 5.6, and 5.8.

**Exercise 5.9** Prove for elements \( a, b \) in a Boolean algebra:
\[
a \sqsubseteq b \iff b^c \sqsubseteq a^c.
\]

**Exercise 5.10** Prove, for elements \( x, y, z \) in a Boolean algebra,
\begin{enumerate}
  \item that \( x \sqsubseteq z \) and \( y \sqsubseteq z \) is equivalent to \( x \sqcup y \sqsubseteq z \);
  \item that \( z \sqsubseteq x \) and \( z \sqsubseteq y \) is equivalent to \( z \sqsubseteq x \sqcap y \);
  \item that \( x \sqcap y = \bot \) is equivalent to \( x \sqsubseteq y^c \). [hint: look at \( x \sqcap (y \sqcup y^c) \).]
\end{enumerate}

**Exercise 5.11** Let \( a \) be an atom in a Boolean algebra and suppose \( x, y \) are two arbitrary elements. Prove:
\begin{enumerate}
  \item \( a \sqsubseteq x^c \iff a \not\sqsubseteq x \). [hint: \( \Rightarrow 5.10 \) (ii); \( \Leftarrow 5.10 \) (iii).]
  \item \( a \sqsubseteq x \sqcup y \iff a \sqsubseteq x \) or \( a \sqsubseteq y \).
\end{enumerate}

**Exercise 5.12** Suppose \((V, \cap, \cup, 0, 1, v \mapsto v^c)\) is a Boolean algebra. Show that for all \( v \) in \( V \) we have:
\begin{enumerate}
  \item \( v \cap v = v \);
  \item \( v \cup v = v \).
\end{enumerate}

**Exercise 5.13** In Exercise 5.4 one should prove that there is no Boolean algebra on three elements. With the results of this section at hand, one can prove much more. What order can a finite Boolean algebra have?

6. **Morphisms**

6.1. **Morphisms of Relations**

**Definition 6.1** Suppose \( A \) and \( B \) are sets, \( R_A \) is a binary relation on \( A \) and \( R_B \) a binary relation on \( B \). A map \( \phi : A \to B \) is called a morphism (with respect to \( R_A \) and \( R_B \)) if and only if for all all \( a, a' \in A \) with \( aR_Aa' \) we also have \( \phi(a)R_B\phi(a') \).

A morphism \( \phi \) induces a map from \( R_A \) to \( R_B \). If \( \phi \) is bijective as well as the induced
map from $R_A$ to $R_B$, then it is called an isomorphism. We call $(A, R_A)$ and $(B, R_B)$
isomorphic if and only if there exists an isomorphism from $A$ to $B$ with respect to the
relations $R_A$ and $R_B$.

If $A = B$ and $R_A = R_B$, then we also call such a $\phi$ a morphism of $R_A$. A
isomorphism of $R_A$ is also called an automorphism.

**Example 6.2** Consider the poset $D(30)$ of all divisors of 30 with order relation $|$ and
the poset $\mathcal{P}((2, 3, 5))$ of all subsets of the set $\{2, 3, 5\}$ with order $\subseteq$. Then the map
$\phi: \mathcal{P}((2, 3, 5)) \rightarrow D(30)$ given by $\phi(X) = \prod_{x \in X} x$ is a morphism between the two
posets. (Notice that by definition we set $\prod_{x \in \emptyset} x = 1$.) The map is bijective and even
an isomorphism (Prove this!).

**Example 6.3** Consider the function $f: \mathbb{R} \rightarrow \mathbb{R}$ defined by $f(x) = e^x$. Then $f$ is a
morphism of the relation $\leq$ on $\mathbb{R}$. Indeed, if $x, y \in \mathbb{R}$ then $x \leq y$ implies $f(x) \leq f(y)$. Actually, any monotone non-decreasing function is a morphism of $\leq$.

**Example 6.4** Let $n \in \mathbb{Z}$ and consider the equivalence relation $\text{mod } n$ on $\mathbb{Z}$. For each
$a \in \mathbb{Z}$ the addition $\text{Add}_a$ with $a$ is a morphism. Indeed, if $x = y \text{ mod } n$, then also
$x + a = y + a \text{ mod } n$. Similarly multiplication with $a$ is also a morphism as, for $x, y \in \mathbb{Z}$
with $x = y \text{ mod } n$ we also have $a \cdot x = a \cdot y \text{ mod } n$.

**Example 6.5** Let $V$ be a finite set and consider the poset $\mathcal{P}(V)$ of all subsets of $V$.
For each $X \subseteq V$ denote by $|X|$ the cardinality of $X$. Then the map $|\cdot|: \mathcal{P}(V) \rightarrow \mathbb{N}$ is
a morphism of $\mathcal{P}(V)$ to $(\mathbb{N}, \leq)$. Indeed, if $X \subseteq Y$, then $|X| \leq |Y|$.

**Example 6.6** If $\Gamma$ and $\Delta$ are graphs then a map $\phi$ from the vertex set of $\Gamma$ to the
vertex set of $\Delta$ is called a (graph) morphism if edges of $\Gamma$ are mapped to edges of $\Delta$.
For example, consider $\Gamma$ and $\Delta$ to be the following graphs:
The map $\phi$ defined by

\[
\begin{align*}
a & \mapsto 1 \\
b & \mapsto 2 \\
c & \mapsto 3 \\
d & \mapsto 1 \\
e & \mapsto 2 \\
f & \mapsto 3
\end{align*}
\]

is a morphism from $\Gamma$ to $\Delta$.

The permutations $\psi = (a,b,c,d,e,f)$ and $\chi = (b,d)(a,e)$ of the vertices of $\Gamma$ are automorphisms of $\Gamma$.

**Proposition 6.7** If $\phi$ is a morphism and $\psi$ also, then $\psi \circ \phi$ is also a morphism.

The inverse of an isomorphism is an isomorphism.

**Exercise 6.1** Prove Proposition 6.7.

**Exercise 6.2** The posets $(D(1001), \mid)$ and $(D(30), \mid)$ are isomorphic. Give an isomorphism between these two posets.

**Exercise 6.3** Determine all 12 automorphism of the graph $\Gamma$ from Example 6.6.

**Exercise 6.4** Give an isomorphism between $(\mathbb{R}, \leq)$ and $(\mathbb{R}^+, \leq)$.

**Exercise 6.5** Let $X$ be a set and consider the poset $\{0,1\}^X$ of maps from $X$ to $\{0,1\}$ with order $\sqsubseteq$ defined by $f \sqsubseteq g$ if and only if $f(x) \leq g(x)$ for all $x \in X$. Give an isomorphism between the poset $\{0,1\}^X$ and $(\mathcal{P}(X), \subseteq)$.

### 6.2. Continuous Maps

In the previous sections we have considered not only relations on sets, but also some operations. For example, $\cap$ and $\sqcup$ are two relations which we have encountered in lattices and Boolean algebras. But also taking complements in Boolean algebras is an operation. In this subsection we will have a closer look at maps between lattices or CPOs that “respect” not only the relations but also these operations. First an example.

**Example 6.8** Consider the lattice $M$ of the following six subsets of $\{1,2,3,4\}$:

$\emptyset, \{1,2\}, \{1,2,3\}, \{1,2,3,4\}$ and $\{1,3\}$

with Hasse diagram

The subset $L$ consisting of

$\emptyset, \{1,2\}, \{1,2,3,4\}$ and $\{1,3\}$

41
is also a lattice. The identity map $\phi$ from $L$ to $M$ is a morphism of the posets on $L$ and $M$. However,

$$\phi(\{1, 2\} \cup_L \{1, 3\}) = \phi(\{1, 2, 3, 4\}) = \{1, 2, 3, 4\},$$

and

$$\phi(\{1, 2\}) \cup_L \phi(\{1, 3\}) = \{1, 2\} \cup_M \{1, 3\} = \{1, 2, 3\}.$$

Hence a poset morphism does not automatically respect the operation $\sqcup$.

Therefore we have the following definition.

**Definition 6.9** Let $L$ and $M$ be two lattices. Then a poset morphism $\phi : L \to M$ is called a lattice morphism if for all $x, y \in L$ we have

$$\phi(x \sqcap_L y) = \phi(x) \sqcap_M \phi(y)$$

and

$$\phi(x \sqcup_L y) = \phi(x) \sqcup_M \phi(y).$$

**Definition 6.10** Let $L$ and $M$ be two complete lattices or CPOs. A map $\phi : L \to M$ is called continuous if for every nonempty chain $C$ we have

$$\phi(\text{sup}(C)) = \text{sup}(\phi(C)).$$

**Example 6.11** Consider the lattice $\mathcal{P}(V)$ of all subsets of a set $V$. If $V$ is a finite set, then the map $|\cdot| : \mathcal{P}(V) \to \{0, \ldots, n\}$, where $n = |V|$, is a poset morphism. In general, its is not a lattice morphism. Indeed, if $X$ and $Y$ are two subsets $|X \cup Y| = \max(|X|, |Y|)$ if and only if $X \subseteq Y$ or $Y \subseteq X$. However, it is a continuous map.

**Exercise 6.6** Let $L$ and $M$ be two complete lattices or CPOs. Prove that a continuous map $\phi : L \to M$ is a poset morphism.

**Exercise 6.7** Consider the lattice $\mathcal{P}(V)$ of all subsets of a set $V$. Let $X$ be a fixed subset of $V$. Then the map $Y \mapsto X \cap Y$ is a continuous lattice morphism from $\mathcal{P}(V)$ to $\mathcal{P}(X)$. 
6.3. Fixed Point Theorems

We are now in a position to discuss a so-called Fix Point Theorem, which is of importance for theoretical computer science.

First we need some terminology. Consider a complete lattice or CPO \( L \) and a map \( \phi : L \to L \). By \( \text{FIX}(\phi) \) we denote the set

\[
\text{FIX}(\phi) = \{ x \in L \mid \phi(x) = x \}.
\]

The infimum of this set (if it exists) is denoted by \( \text{fix}(\phi) \). The set of dec-points is the set

\[
\text{DEC}(\phi) = \{ x \in L \mid \phi(x) \sqsubseteq x \}.
\]

Clearly \( \text{FIX}(\phi) \subseteq \text{DEC}(\phi) \). We will show that for continuous maps \( \phi \) the element \( \text{fix}(\phi) \) exists. Moreover, for all \( x \in \text{DEC}(\phi) \) we have \( \text{fix}(\phi) \sqsubseteq x \). So, \( \text{fix}(\phi) \) is also the infimum of \( \text{DEC}(\phi) \). If \( \phi \) is a continuous map, then we can even find \( \text{fix}(\phi) \) as \( \sup \{ \phi^i(\bot) \mid i \in \mathbb{N} \} \).

We state these results in the following theorem.

**Theorem 6.12 [Fixed Point Theorem]** Suppose \( L \) is a CPO. If \( \phi : L \to L \) is a continuous map then \( \text{fix}(\phi) \) exists. Moreover, we have

\[
\text{fix}(\phi) = \text{inf}(\text{DEC}(\phi)) = \sup \{ \phi^i(\bot) \mid i \in \mathbb{N} \}.
\]

**Proof.** Suppose \( L \) is a complete lattice or CPO and \( \phi : L \to L \) is a continuous map. Then \( \bot \sqsubseteq \phi(\bot) \), thus \( C := \{ \phi^i(\bot) \mid i \in \mathbb{N} \} \) is a chain in \( L \). This chain has a supremum, \( s \) say. The image of the chain \( C \) under \( \phi \) equals the chain \( \{ \phi^{i+1}(\bot) \mid i \in \mathbb{N} \} \), which has the same supremum as \( C \). Hence, \( \phi(s) = s \in \text{DEC}(\phi) \).

Let \( x \in \text{DEC}(\phi) \). Then, as \( \bot \sqsubseteq x \) we also have \( \phi(\bot) \sqsubseteq \phi(x) \sqsubseteq x \). With induction we can even show that \( \phi^i(\bot) \sqsubseteq x \) for all \( i \in \mathbb{N} \). But then \( s \sqsubseteq x \). Hence \( s \sqsubseteq \text{inf}(\text{DEC}(\phi)) \), but since \( s \in \text{DEC}(\phi) \), we even have \( s = \text{inf}(\text{DEC}(\phi)) \) and also \( s = \text{inf}(\text{FIX}(\phi)) \).

**Example 6.13** Consider the interval \([0, 1]\) of \( \mathbb{R} \) with order \( \leq \). This is a CPO. Let \( f : [0, 1] \to [0, 1] \) be a non-decreasing and analytically continuous function. Then \( f \) is also continuous as a map of posets. Hence the above Fixed Point Theorem implies that there is an \( x \in [0, 1] \) with \( f(x) = x \). The point may be found by computing \( f(0), f^2(0), f^3(0), \ldots \).

**Example 6.14** This fixed point theorem is used in computer science to have a theoretical sound definition of recursively defined functions. We explain this by an example. (See also [2].) Consider the set \( P \) of partial maps from \( \mathbb{N} \) to \( \mathbb{N} \). This is a complete partial order as we have seen in Example 4.26. The function \( f : \mathbb{N} \to \mathbb{N} \) defined by \( f(n) = n! \) is in this set. On a computer this function \( f \) is often defined recursively by:

\[
f(0) = 1; \ f(n + 1) = (n + 1) \cdot f(n).
\]
In fact, as a computer can only calculate the value of $f$ in a finite interval, it only knows partial maps. Or better, a chain of partial maps. Indeed, if $f$ is a partial map from $\mathbb{N}$ to $\mathbb{N}$, known to the computer, then it also knows $\phi(f)$, the partial map given by

$$
\phi(f)(z) = 1 \text{ if } z = 0
$$

$$
\phi(f)(z) = z \cdot f(z - 1) \text{ if } z - 1 \in \text{Domain}(f)
$$

$$
\phi(f)(z) \text{ is undefined if } z - 1 \notin \text{Domain}(f) \text{ and } z \neq 0.
$$

Clearly $\phi$ is a morphism. But, as we will prove below, it is also continuous. Indeed, let $C$ be a chain in $P$ with supremum $c$. Notice that the graph of $c$ is the union of all the graphs from the elements in $C$. Then, as $\phi$ is a morphism, $\phi(c)$ is an upperbound of $\phi(C)$. But is it also the smallest upperbound? Let $d$ be the supremum of the chain $\phi(C)$. So $d \subseteq \phi(c)$. So, $d$ is the partial map whose graph is the union of all the graphs in $\phi(C)$. Suppose $(z, \phi(c)(z))$ is a point on the graph of $\phi(c)$. If $z = 0$, then this is the point $(0, 1)$ which is also on the graph of $d$. If $z \neq 0$, then $z - 1$ is in the domain of $c$ and thus in the domain of some element $c_0 \in C$. But then

$$
\phi(c)(z) = z \cdot c(z - 1) = z \cdot c_0(z - 1) = \phi(c_0)z.
$$

So, $(z, \phi(c)(z))$ is also on the graph of $\phi(c_0)$. This implies that the graph of $\phi(c)$ is contained in the graph of $d$. But then $\phi(c) \subseteq d$ and by antisymmetry of $\subseteq$ we have $d = \phi(c)$.

Now the function $f$ can be defined as the unique element $\text{fix}(\phi)$.

Of course, we can apply the same procedure to many recursively defined functions. For example, the map $g : \mathbb{N} \to \mathbb{N}$ recursively defined by

$$
g(0) = 1; g(z) = (z^2 + 1) \cdot g(z - 1),
$$

44
is the smallest fixed point of the continuous morphism $\psi : P \to P$ given by

$$
\psi(f)(z) = \begin{cases} 
1 & \text{if } z = 0 \\
(z^2 + 1) \cdot f(z - 1) & \text{if } z - 1 \in \text{Domain}(f) \\
\text{undefined} & \text{if } z - 1 \notin \text{Domain}(f) \text{ and } z \neq 0.
\end{cases}
$$

An application of the above Fixed Point Theorem is the following theorem:

**Theorem 6.15 [Cantor and Bernstein]** Suppose $A$ and $B$ are two sets. If there exist two injective maps $f : A \to B$ and $g : B \to A$, then there exists a bijection $h : A \to B$.

**Proof.** Of course, $g^{-1}$ is a bijective map from $g(B) \subseteq A$ to $B$. We will partition the set $A$ into two sets $X \subseteq g(B)$ and its complement $X^c$, such that the map $h : A \to B$ given by $h(a) = g^{-1}(a)$ for $a \in X$ and $h(a) = f(a)$ for $a \in X^c$ is map we are looking for.

Such a map $h$ is injective if and only if $f(X^c) \cap g(X) = \emptyset$. It is surjective if and only if $X = g(f(X^c)^c)$.

We can find such a set $X$ as a fixed point of the map $\phi : \mathcal{P}(g(B)) \to \mathcal{P}(g(B))$ given by

$$
\phi(Y) = g(f(Y^c)^c).
$$

The mapping $\phi$ is a morphism. Indeed if $Z \subseteq Y$ then

$$
Z^c \supseteq Y^c \\
f(Z^c) \supseteq f(Y^c) \\
f(Z^c)^c \subseteq f(Y^c)^c \\
g(f(Z^c)^c) \subseteq g(f(Y^c)^c) \\
\phi(Z) \subseteq \phi(Y)
$$

That $\phi$ is continuous follows easily, since it respects taking intersections or unions. The fixed point theorem now guarantees that there exists an $X$ as wanted. Moreover, it also tells us how to find it. Namely, it is the supremum of the chain

$$
\emptyset \subseteq \phi(\emptyset) \subseteq \phi^2(\emptyset) \subseteq \cdots.
$$

□

**Example 6.16** In the proof of the above theorem we not only showed that there is a bijection $h$ from $A$ to $B$, but we also proved the existence of a bijection $h$ and a subset $X \subseteq g(B)$ such that $h|_X = g|_X^{-1}$ and $h|_{X^c} = f|_{X^c}$. 45
Let us given an example. Suppose $f, g : \mathbb{N} \to \mathbb{N}$ are defined by $f(n) = 2n = g(n)$. Now the set $X$ can be found as the supremum of the chain

$$\emptyset \subseteq \phi(\emptyset) \subseteq \phi^2(\emptyset) \subseteq \cdots,$$

where $\phi(Y) = g(f(X)^c)$.

Now

$$\phi(\emptyset) = \{n \in \mathbb{N} \mid \text{ord}_2(n) = 1\},$$
$$\phi^2(\emptyset) = \{n \in \mathbb{N} \mid \text{ord}_2(n) = 1, 3\},$$
$$\phi^3(\emptyset) = \{n \in \mathbb{N} \mid \text{ord}_2(n) = 1, 3, 5\},$$

and so on.

So, $X$ is the set of all $n \in \mathbb{N}$ with $\text{ord}_2(n)$ odd.

**Exercise 6.8** Consider the sequence $a_n, n \in \mathbb{N}$ defined by $a_1 = 1$ and $a_{n+1} = \sqrt{a_n + 1}$. Use the Fixed Point Theorem to determine $\lim_{n\to\infty} a_n$. (hint: consider the function $f : [0, 2] \to [0, 2]$ defined by $f(x) = \sqrt{1 + x}$.)

**Exercise 6.9** Consider the map $\phi$ as given in Example 6.16. Prove (for example by induction) that for $n > 0$ we have $\phi^n(\emptyset)$ equals $\{m \in \mathbb{N} \mid \text{ord}_2(m) = 1, 3, 5, \ldots, 2n-1\}$.

**Exercise 6.10** Let $f : \mathbb{N} \to \mathbb{N}$ be a defined by $f(n) = 2n$ and $g : \mathbb{N} \to \mathbb{N}$ be a defined by $f(n) = 3n$. Then both $f$ and $g$ are injective maps from $\mathbb{N}$ to $\mathbb{N}$.

Find a bijection $h : \mathbb{N} \to \mathbb{N}$ and a set $X \subseteq 3\mathbb{N} = g(\mathbb{N})$ such that $h|_X = g|_X^{-1}$ and $h|_{X^c} = f|_{X^c}$.

**References**

