Background material for the Course GCSCI-GA.3033-011 “Principles of Software Security”, New York University, CIMS, CS, Fall 2013

Patrick Cousot
CIMS, NYU

September 3, 2013
Chapter 2

Elements of Naïve Set Theory

We recall very basic mathematical concepts of set theory which are of constant use in formal methods in Computer Science. The material is elementary and can be skipped by the knowledgeable, or read very rapidly to fix notations, or read later when needed using the notation index.

2.1 Logic

2.1.1 Logical formulæ

For conciseness of explanations and proofs, we use logical notations as a metalanguage (i.e. logic is not our subject of study but is the formal language we use, in complement to English, to discuss our subject of study).

Logical statements such as “6 is even”, “6 is odd”, or “if a natural number is even then its successor is odd” are either true or false but not both. Following George Boole (Boole, 1847, Proposition 1, p. 27) such logical statements are written symbolically for conciseness.

For example, \(\text{even}(6)\) (meaning “6 is even”, which is true), \(\text{odd}(6)\) (meaning “6 is odd”, which is false), \(\forall n \in \mathbb{N} : \text{even}(n) \Rightarrow \text{odd}(n+1)\) (for all natural numbers \(n\), if \(n\) is even then \(n + 1\) is odd, which is true), where \(\text{even}(n) \triangleq \exists k : n = 2k\) (meaning than even is defined as being a multiple of 2) and \(\text{odd}(n) \triangleq \neg \text{even}(n)\) (meaning than odd is defined as not even).

Such symbolic logical formulæ include \(x, X, X, \mathcal{X}, \ldots\) (variables \(^1\)), \(0, 1, \ldots\) (constants), \(+, -, \ldots\) (operators and functions) to build terms, false (false), true (true), = (equality), \(\neq\).

---

\(^1\) i.e. abbreviates the latin expression “id est”, that is.

\(^2\) The typography provides information on the type of the variable. For example, \(x\) is an atomic element, not a set, \(X\) is un parameter set such as in a function or a quantifier, \(\mathcal{X}\) is a generic arbitrary set, \(\mathcal{X}\) is a classical set defined in Mathematics (\(\mathbb{N}\) for the natural numbers/non-negative integers, \(\mathbb{Z}\) for the integers, \(\mathbb{R}\) for the reals, \(\mathbb{R}^+\) for the non-negative reals, etc.), \(\mathcal{X}\) is a language that is a syntactic set, \(\mathcal{X}\) is a mathematical set whereas \(f\) is a mathematical function defined in the book and referenced in the index of notations, \(\mathcal{X}\) is a set of sets.
(disequality), ... (relations) to built atomic formulæ, ⇒ (implication), ⇔ (logical equivalence “if and only if” or “iff”), ∨ (disjunction), ∧ (conjunction), ¬ (not) to built propositional formulæ. These logical operations can all be defined in terms of a primitive Boolean operation nand given by the truth table (also called Cayley table (Cayley, 1854))

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P nand Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>

and the definitions ¬P \( \equiv \) P nand P, P \( \land \) Q \( \equiv \) ¬(P nand Q), P \( \lor \) Q \( \equiv \) ¬(¬P \( \land \) ¬Q), P nor Q \( \equiv \) ¬(P \( \lor \) Q), P xor Q \( \equiv \) (P \( \land \) ¬Q) \( \lor \) (¬P \( \land \) Q), P ⇒ Q \( \equiv \) ¬P \( \lor \) Q, P ⇔ Q \( \equiv \) (P ⇒ Q) \( \land \) (Q ⇒ P).

Symbolic logical formulæ also include quantifiers ∀ (for all), ∃ (exists), \( \exists ! \) (does not exist any), \( \exists ! ! \) (exists a unique), introduced by Augustus De Morgan ³, Charles Sanders Peirce ⁴, Gottlob Frege ⁵ and Giuseppe L. Peano ⁶ (De Morgan, 1840; Peirce, 1870; Peirce, 1885; Frege, 1879; Peano, 1895) to built predicates/logical formulæ. \( \exists x : P(x) \) stands for \( \exists x : P(x) \land \forall y : P(y) \Rightarrow x = y \) and \( \forall x : P(x) \) for \( ¬(\exists x : P(x)) \). We write \( \forall x : P(x) \) if and only if \( \forall x : x \in \mathcal{X} \Rightarrow P(x) \) and \( \exists x : x \in \mathcal{X} : P(x) \) for \( \exists x : x \in \mathcal{X} \land P(x) \). In particular for the empty set \( \emptyset \), \( \forall x \in \emptyset : P(x) \) is true while \( \exists x \in \emptyset : P(x) \) is false.

Logical formulæ \( P, Q, \ldots \in \mathcal{L} \) are sentences of a formal language \( \mathcal{L} \) based on these logical notations. A logical formula \( P \) is a statement/assertion describing properties of the entities designated by variables. Such a statement/assertion can either be valid/true or else invalid/false. So a logical formula must be understood in its context. It can be an hypothesis \( P \) or a conclusion \( Q \) of a theorem \( P \Rightarrow Q \), or a theorem itself \( T = P \Rightarrow Q \). For such a logical formula \( T \) to be believed true, a proof must be done, which is discussed in chapter ⁵.

Examples of logical formulæ of \( \mathcal{L} \) are false ⇒ true (which is true ⁷), \( \forall x \in \mathbb{R} : (x + 1)^2 = x^2 + 2x + 1 \) (which is true), \( \exists x : (x + 1)^2 = 0 \) (which is true when \( x \in \mathbb{Z} \) is an integer and false when \( x \in \mathbb{N} \) is a natural number that is a non-negative integer), \( \forall x : \lfloor x \rfloor \neq \text{nil} \) (a list of one element is not empty, which is true).

### 2.1.2 Quantification

Notice that \( \forall x : \forall y : P(x, y) \) if and only if \( \forall y : \forall x : P(x, y) \) and \( \exists x : \exists y : P(x, y) \) if and only if \( \exists y : \exists x : P(x, y) \). However \( \forall x : \exists y : P(x, y) \) and \( \exists y : \forall x : P(x, y) \) are using “Some” for \( \exists \) and “Every” for \( \forall \)

³ using “Some” for \( \exists \) and “Every” for \( \forall \)
⁴ using \( \sum \) for \( \exists \) and \( \prod \) for \( \forall \)
⁵ using a two-dimensional diagrammatic notation.
⁶ using the modern \( \exists \).
⁷ The intuition is that when we have reached an absurdity, then everything goes.
CHAPTER 2. ELEMENTS OF NAÏVE SET THEORY

not the same. For example $\forall x \in \mathbb{Z} : \exists y \in \mathbb{Z} : x + y = 0$ is true (choose $y = -x$) but $\exists y \in \mathbb{Z} : \forall x \in \mathbb{Z} : x + y = 0$ is false (since e.g. if such an $y$ exists then for $x = -y + 1$ we would have $x + y = (-y + 1) + y = 1 \neq 0$, a contradiction\(^8\)).

2.1.3 Negation

The negation $\neg P$ of $P$ is its "contrary" (or "opposite"). Formally $P \lor \neg P = \text{true}$, $P \land \neg P = \text{false}$ (so $\neg (P \land \neg P)$ is true), and $( (P \lor Q) \land \neg (P \land Q) ) \Rightarrow (Q = \neg P)$ so the negation is unique. It follows that $\neg \neg P = P$, $\neg (P \lor Q) = \neg P \land \neg Q$, so $\neg (\exists x \in \mathcal{X} : P(x)) = \forall \mathcal{X} \neg P(x) = \exists \mathcal{X} : \neg P(x)$ and $\neg (P \land Q) = \neg P \lor \neg Q$, so $\neg (\forall \mathcal{X} : P(x)) = \exists \mathcal{X} \neg P(x) = \exists \mathcal{X} : \neg P(x)$.

2.1.4 Variable renaming

A variable in a logic formula can be free/global (not under the scope of a quantifier) or bound/local (under the scope of a quantifier). For example in formula $\forall x : \exists y : x = z + y$, $x$, $y$ are bound while $z$ is free. In formula $\exists y : x = z + y$, $y$ is bound while $x$, $z$ are free, $x$, $y$, $z$ are all free in $x = z + y$.

Renaming the bound variables, does not change the meaning of the formula. For example $\forall x : \exists y : x = z + y$ is the same as $\forall a : \exists b : a = z + b$. This is similar to changing the name of the parameters uniformly in the body of the procedure. In both cases, there is a danger of variable capture for example renaming $x$ into $z$ would yield $\forall z : \exists y : z = z + y$ which is different from $\forall x : \exists y : x = z + y$ where $z$ is free/global. Variable capture can always be avoided by renaming with fresh variables, not appearing in the formula to be renamed.

2.1.5 Conventions

We write $X \triangleq D$ to mean that $X$ is defined to be $D$ and so $X = D$ onwards. $D$ may depend upon $X$ but then precautions have to be taken to ensure that $X$ is well-defined (that is does exist and is unique), which is discussed in chapter 7, and then in greater details in part ??.

To stick with usual conventions in Computer Science, when $P \Rightarrow Q$, we say that $P$ is stronger than $Q$ and that $Q$ is weaker than $P$ (Dijkstra and Scholten, 1990). We define the conditional if $\text{true}$ then $a$ else $b$ $\triangleq a$ and if $\text{false}$ then $a$ else $b$ $\triangleq b$.

In order to define more precisely the logical language $\mathcal{L}$ we would have to define the individual constants, variables, functions, relation, and predicates/logical formula that we use. We assume this is well known for basic mathematical structures such as naturals $\mathbb{N}$, integers $\mathbb{Z}$, ..., sets (as reviewed in section 2.2), .... As far as Computer Science structures are concerned, the notations will be introduced on the fly (and summarized in the notation index).

---

\(^8\) e.g. abbreviates the latin expression “exempli gratia”, for example or example given.

\(^9\) This is a proof by reductio ad absurdum or by contradiction discussed insection 5.11.
However, a choice has to be made on what can be quantified upon in existential and universal quantification, which is discussed in chapter 3.

## 2.2 Sets

The basic notion $x \in S$ in set theory is that an element $x$ belongs to set $S$ which can only be true or false. Its negation $x \notin S \triangleq \neg(x \in S)$ means that element $x$ does not belong to the set $S$. Using this primitive notion, one can then define

- $\{ x \mid P(x) \}$ denotes the set $S$ such that $\forall x : x \in S \iff P(x)$ where $P(x)$ is a predicate in $x$ hence can only be true or false. $\{ x \mid x \in X \land P(x) \}$.

- The empty set $\emptyset$ such that no element belongs to $\emptyset$: $\forall x : x \notin \emptyset$ so $\emptyset = \{ x \mid \text{false} \}$. For example, the closed interval is empty $[a, b] = \emptyset$ when $b < a$;

- The set equality $S = S'$ meaning that the two sets have the same elements $\forall x : x \in S \iff x \in S'$; Observe that $\equiv$ corresponds to logical equivalence in that $\{ x \mid P(x) \} = \{ x \mid Q(x) \}$ if and only if $P \iff Q$;

- The set inclusion $S \subseteq S'$ meaning that all elements of $S$ are also elements of $S'$: $\forall x : (x \in S) \Rightarrow (x \in S')$. Observe that $\subseteq$ corresponds to implication in that $\{ x \mid P(x) \} \subseteq \{ x \mid Q(x) \}$ if and only if $P \Rightarrow Q$;

- The binary set union/join/least upper bound (lub) $S \cup S'$ which elements are those of either $S$ or $S'$: $\forall x : (x \in S \cup S') \Leftrightarrow (x \in S \lor x \in S')$ so that $S \cup S' = \{ x \mid x \in S \lor x \in S' \}$, so that union $\cup$ corresponds to disjunction $\lor$;

- The binary set intersection/meet/greatest lower bound (glb) $S \cap S'$ which elements are those in both $S$ and $S'$: $\forall x : (x \in S \cap S') \Leftrightarrow (x \in S \land x \in S')$ so that $S \cap S' = \{ x \mid x \in S \land x \in S' \}$, so that intersection $\cap$ corresponds to conjunction $\land$;

- The set complement $S \setminus S'$ which are the elements of set $S$ not in set $S'$: $\forall x : (x \in S \setminus S') \Leftrightarrow (x \in S \land x \notin S')$ so that $S \setminus S' = \{ x \mid x \in S \land x \notin S' \}$. When $S$ is understood form the context, we write $\neg S'$ to mean $S \setminus S'$ so that $\neg\{ x \mid P(x) \} = \{ x \mid \neg P(x) \}$ and set complement $\neg$ corresponds to logical negation $\neg$;

**Example 1.** The set operations are often illustrated using Leonhard L. Euler/John Archibald Venn’s diagrams (Euler, 1773; Venn, 1880). The interior of a patatoid作文。
CHAPTER 2. ELEMENTS OF NAÏVE SET THEORY

symbolically represents the elements of a set, while the exterior represents elements that are not members of this set.

- The powerset \( \varphi(S) \) which is the set of all subsets of \( S \): \( \forall X : X \in \varphi(S) \iff X \subseteq S \) so that \( \emptyset \in \varphi(S), S \in \varphi(S) \) and \( \varphi(S) = \{ X \mid X \subseteq S \} \).

The power set \( \varphi(S) \) of the set \( S = \{-1, 0, 1\} \) is represented by its Hasse diagram.\(^{12}\)

**Example 2.**

The vertices \( \bullet \) are labelled by an element \( X \) of \( \varphi(S) \) and there is an arc \( - \) (implicitly oriented upward) between two vertices labelled \( X, Y \in \varphi(S) \) if \( X \subseteq Y \) and \( X \not\subseteq Z \in \varphi(S) : X \subseteq Z \subseteq Y \) (i.e. \( Y \) covers \( X \)). So \( X \subseteq Y \) and only if \( X \) and \( Y \) are linked by zero or more arcs in the Hasse diagram.

Classical extensions of these basic notations include

- The enumerated set \( \{ x_1, \ldots, x_n \} \) denotes the set \( S \) such that \( \forall x : x \in S \iff \exists i \in \mathbb{N} : 1 \leq i \leq n \land x = x_i \). For \( n = 0 \), the set is the empty set \( \emptyset \). For \( n = 1 \), the set \( \{ x_1 \} \) is called a singleton;\(^{12}\)

• The closed interval \([a, b]\) \(\triangleq \{x \mid a \leq x \leq b\}\), (so that \([a, b] = \emptyset\) when \(b < a\));

• The left-opened interval \((a, b]\) \(\triangleq \{x \mid a < x \leq b\}\);

• The right-opened interval \([a, b)\) \(\triangleq \{x \mid a \leq x < b\}\);

• The opened interval \((a, b)\) \(\triangleq \{x \mid a < x < b\}\);

where the set to which \(x\) belongs (e.g. \(\mathbb{N}, \mathbb{Q}, \mathbb{R}\)) is left dependent on the context of the sentence where the notation is used.

The binary union and intersection operators can be extended to infinitely many sets such as \(\mathbb{N} = \bigcup\{\{i\} \mid i \in \mathbb{N}\} = \bigcup_{i \in \mathbb{N}} \{i\}\). Given a set of sets \(X\), we define

• The infinite set union/join \(\bigcup X\) which elements are those of some subset of \(X\): \(\forall x : (x \in \bigcup X) \iff (\exists Y : Y \in X \land x \in Y)\) so that \(\bigcup X = \{x \mid \exists Y : Y \in X \land x \in Y\} = \{x \mid \exists Y \in X : x \in Y\} = \bigcup_{Y \in X} Y\);

• The infinite set intersection/meet \(\bigcap X\) which elements are in all subsets of \(X\): \(\forall x : (x \in \bigcap X) \iff (\forall Y : Y \in X \Rightarrow x \in Y)\) so that \(\bigcap X = \{x \mid \forall Y : Y \in X \Rightarrow x \in Y\} = \{x \mid \forall Y \in X : x \in Y\} = \bigcap_{Y \in X} Y\).

### 2.3 Partial orders

Set inclusion is a partial order, that is (Schröder, 1890, Prinzip I, II, and Definition (1), pp. 168–184)\(^{13}\)

(a) reflexive \((\forall X : X \subseteq X)\).

(b) transitive \((\forall X, Y, Z : (X \subseteq Y \land Y \subseteq Z) \Rightarrow (X \subseteq Z))\).

(c) antisymmetric \((\forall X, Y : (X \subseteq Y \land Y \subseteq X) \Rightarrow (X = Y))\).

Moreover, \(X \subseteq Y \iff X \cap Y = X \iff X \cup Y = Y\), see exercise 2.26-2.

The powerset \(\wp(S)\) of at set \(S\), equipped with the \(\subseteq\) relation is a poset \(\langle S, \subseteq \rangle\) (which means that \(\subseteq\) is a partial order relation on \(S\)).

Strict inclusion \(X \subset Y \triangleq X \subseteq Y \land X \neq Y\) (also written \(\subset\)) is a strict partial order (which is strict \((\forall X, Y : (X \subseteq Y \land Y \subset X) \Rightarrow (X \neq Y))\) and transitive.

\(^{13}\) Ernst Schröder uses the same notation \(\subseteq\) both for \(\in\) and \(\subset\)!
2.4 Maximal/minimal elements, maximum/minimum

A **maximal element** of a set of sets $\mathcal{X} \subseteq \mathcal{P}(S)$ is a set $\mathcal{M} \in \mathcal{X}$ with no greater element in $\mathcal{X}$ that is such that $\forall X \in \mathcal{X} : \mathcal{M} \subseteq X$. In case there is only one maximal element in $\mathcal{X}$, it is called the **maximum** $\max \mathcal{X}$ of $\mathcal{X}$. The dual notions are those of **minimal elements** and **minimum** $\min \mathcal{X}$, if any.

2.5 Upper bound, Lower bound

An **upper bound** of a set of sets $\mathcal{X} \subseteq \mathcal{P}(S)$ (i.e. $\mathcal{X} \subseteq \mathcal{P}(\mathcal{S}))$ is a set $\mathcal{U} \in \mathcal{P}(S)$ such that $\forall X \in \mathcal{X} : X \subseteq \mathcal{U}$. A **lower bound** of a set of sets $\mathcal{X} \subseteq \mathcal{P}(S)$ is a set $\mathcal{L} \in \mathcal{P}(S)$ such that $\forall X \in \mathcal{X} : \mathcal{L} \subseteq X$. Let us define

\[
\begin{align*}
\text{ub}(\mathcal{X}) & \triangleq \{ \mathcal{U} \in \mathcal{P}(S) \mid \forall X \in \mathcal{X} : X \subseteq \mathcal{U} \} & \text{upper bounds of } \mathcal{X} \\
\text{lb}(\mathcal{X}) & \triangleq \{ \mathcal{L} \in \mathcal{P}(S) \mid \forall X \in \mathcal{X} : \mathcal{L} \subseteq X \} & \text{lower bounds of } \mathcal{X}
\end{align*}
\]

**Lemma 3.** $\forall \mathcal{X}, \mathcal{Y} \in \mathcal{P}(\mathcal{P}(S)) : \mathcal{X} \subseteq \text{lb}(\mathcal{Y}) \iff \mathcal{Y} \subseteq \text{ub}(\mathcal{X})$.

**Proof.** Let $\mathcal{X}, \mathcal{Y} \in \mathcal{P}(\mathcal{P}(S))$.

\[
\begin{align*}
\mathcal{X} & \subseteq \text{lb}(\mathcal{Y}) \\
\iff & \mathcal{X} \subseteq \{ \mathcal{L} \in \mathcal{P}(S) \mid \forall X \in \mathcal{Y} : \mathcal{L} \subseteq X \} \\
\iff & \forall \mathcal{L} \in \mathcal{X} : \forall X \in \mathcal{Y} : \mathcal{L} \subseteq X \\
\iff & \forall \mathcal{U} \in \mathcal{Y} : \forall X \in \mathcal{X} : X \subseteq \mathcal{U}
\end{align*}
\]

\[14\text{ equivalently, } \forall X \in \mathcal{X} : \mathcal{M} \subseteq X \iff \forall X \in \mathcal{X} : \neg (\mathcal{M} \subseteq X) \iff \forall X \in \mathcal{X} : \neg (\mathcal{M} \subseteq X) \lor \mathcal{M} = X) \iff \forall X \in \mathcal{X} : (\mathcal{M} \subseteq X) \Rightarrow (\mathcal{M} = X).\]
Given two sets \( X \) and \( Y \), \( X \subseteq Y \) if \( X \) is a subset of \( Y \). So it is called the upper bound. Then \( [\neg X, X] \) is called the greatest lower bound (glb) and \( [X, Y] \) the least upper bound (lub). It is the greatest lower bound so that the abstract properties are either \( \land \) or \( \lor \) for the glb and lub, respectively. It is the unique greatest lower bound. The maximum \( X \) is defined in terms of the glb and lub.

2.6 Least upper bound, greatest lower bound

The join \( \cup \mathcal{X} \in \wp(S) \) is an upper bound of \( \mathcal{X} \in \wp(\wp(S)) \). It is the least upper bound (lub) in that if \( \mathcal{U} \in \wp(S) \) is any upper bound of \( \mathcal{X} \) then \( \cup \mathcal{X} \subseteq \mathcal{U} \) is smaller or equal. By antisymmetry, the lub is unique and if \( \max \mathcal{X} \) exists then \( \cup \mathcal{X} = \max \mathcal{X} \).

The meet \( \cap \mathcal{X} \in \wp(S) \) is a lower bound of \( \mathcal{X} \). It is the greatest lower bound (glb) in that if \( \mathcal{L} \in \wp(S) \) is any lower bound of \( \mathcal{X} \) then \( \mathcal{L} \subseteq \cap \mathcal{X} \) is larger or equal. By antisymmetry, the glb is unique and if \( \min \mathcal{X} \) exists then \( \cap \mathcal{X} = \min \mathcal{X} \).

2.7 Lattice

Let \( S \) be a set and \( \mathcal{L} \subseteq \wp(\wp(S)) \) be a subset of \( \wp(S) \). As a subset of the poset \( \langle \wp(S), \subseteq \rangle \), \( \langle \mathcal{L}, \subseteq \rangle \) is a poset. Moreover, \( \langle \mathcal{L}, \subseteq \rangle \) is said to be

(a) a join-semi-lattice when lubs of finite subsets do exist in \( \mathcal{L} \) (or equivalently \( \forall X, Y \in \mathcal{L} : X \cup Y \in \mathcal{L} \));

(b) a meet-semi-lattice when glbs of finite subsets do exist in \( \mathcal{L} \) (or equivalently \( \forall X, Y \in \mathcal{L} : X \cap Y \in \mathcal{L} \));

(c) a lattice when it is both a join-semi-lattice and a meet-semi-lattice (or equivalently \( \forall X, Y \in \mathcal{L} : X \cup Y \in \mathcal{L} \wedge X \cap Y \in \mathcal{L} \)).
CHAPTER 2. ELEMENTS OF NAIVE SET THEORY

**Example 4.** A subset of a power set, which is a

\[
\begin{array}{cccc}
\text{poset} & \text{join-semi-lattice} & \text{meet-semi-lattice} & \text{lattice} \\
\end{array}
\]

A join-semi-lattice satisfies the following properties \(I^\lor\), \(C^\lor\), and \(A^\lor\), a meet-semi-lattice satisfies the properties \(I^\land\), \(C^\land\), and \(A^\land\), and a lattice satisfies the properties \(C^\lor\), \(A^\lor\), \(C^\land\), \(A^\land\), \(B^\lor\), and \(B^\land\).

\[
\begin{align*}
(I^\lor) & \quad X \cup X = X \\
(C^\lor) & \quad X \cup Y = Y \cup X \\
(A^\lor) & \quad (X \cup Y) \cup Z = X \cup (Y \cup Z) \\
(B^\lor) & \quad X \cap (X \cup Y) = X \\
\end{align*}
\]

\[
\begin{align*}
(I^\land) & \quad X \cap X = X \\
(C^\land) & \quad X \cap Y = Y \cap X \\
(A^\land) & \quad (X \cap Y) \cap Z = X \cap (Y \cap Z) \\
(B^\land) & \quad X \cup (X \cap Y) = X \\
\end{align*}
\]

Reciprocally a set \(\langle L, \cup \rangle\) equipped with a binary operation \(\cup\) satisfying \((I^\lor)\), \((C^\lor)\), and \((A^\lor)\) is a join-semi-lattice, \(\langle L, \cap \rangle\) satisfying \((I^\land)\), \((C^\land)\), and \((A^\land)\) is a meet-semi-lattice, and \(\langle L, \cup, \cap \rangle\) satisfying \((C^\lor)\), \((A^\lor)\), \((C^\land)\), \((A^\land)\), \((B^\lor)\), and \((B^\land)\) is a lattice where the partial order is defined a \(X \subseteq Y \iff X \cup Y = Y\) (and respectively or equivalently by \(X \cap Y = X\)).

The algebraic definition of lattices is due to Ernst Schröder (Schröder, 1890, Dritte Vorlesung, pp. 191–201) (using multiplication for \(\cap\) and addition for \(\cup\), after Boole and (Boole, 1847; Peirce, 1870)) and the axiomatization to Julius Wilhelm Richard Dedekind (Dedekind, 1931).

### 2.8 Complete lattice

A subset \(\mathcal{L} \in \wp(\wp(S))\) of \(\wp(S)\) is a **complete lattice** if and only if any subset \(\mathcal{X}\) of \(\mathcal{L}\) has a lub \(\bigcup \mathcal{X}\) and a supremum \(\bigcup \wp(\mathcal{S}) = \mathcal{S}\), and greatest lower bounds which are the lubs of the lower bounds that is \(\bigcap \mathcal{F} = \bigcup\{L \in \wp(S) \mid \forall X \in \mathcal{F} : L \subseteq X\}\) (see exercise 2.26-12 for more details).

**Example 5.** The following lattices are complete (but not Boolean).

---

\(^{15}\)This lattice is not *complete*.

\(^{16}\)from which \(I^\lor\) follows by \(X \cap X = X \cap (X \cup (X \cap Y))\) (by \(B^\land = X\) (by \(B^\lor\)) and dually for \(I^\land\).

\(^{17}\)i.e. not only the finite subsets as for join-semi-lattices but also the infinite subsets.
\[ \mathbb{N} = \{0, 1, 2, 3, \ldots \} \]
\[ 5 = \{0, 1, 2, 3, 4\} \]
\[ 4 = \{0, 1, 2\} \]
\[ 3 = \{0, 1, 2\} \]
\[ 2 = \{0, 1\} \]
\[ 1 = \{0\} \]
\[ 0 = \emptyset \]

2.9 Complete Boolean lattice

A subset \( \mathcal{L} \in \wp(\wp(S)) \) of \( \wp(S) \) is a complete Boolean lattice if and only if it is a complete lattice each subset \( \mathcal{X} \in \wp(\mathcal{L}) \) has a unique complement \( \neg \mathcal{X} \) such that \( \mathcal{X} \cap \neg \mathcal{X} = \emptyset \) and \( \mathcal{X} \cup \neg \mathcal{X} = \bigcup \mathcal{L} \).

Example 6. The powerset \( \wp(\mathcal{S}) \), \( \subseteq, \emptyset, \mathcal{S}, \bigcup, \bigcap, \neg \) where \( \neg \mathcal{X} \triangleq \mathcal{S} \setminus \mathcal{X} \) is a complete Boolean lattice.

The following complete lattice is also Boolean.

\[ \neg\{1\} = \{-1, 0\} \]
\[ \{0, 1\} = \neg\{-1\} \]
\[ \neg\{0, 1\} = \{-1\} \]
\[ \{1\} = \neg\{-1, 0\} \]
\[ \emptyset = \neg\{-1, 0, 1\} \]

2.10 Pairs, tuples, Cartesian product

Given two sets \( \mathcal{Y} \) and \( \mathcal{Y} \), the Cartesian product \( \mathcal{X} \times \mathcal{Y} \) is defined as \( \mathcal{X} \times \mathcal{Y} \triangleq \{\langle x, y \rangle \mid x \in \mathcal{X} \land y \in \mathcal{Y}\} \). Pairs \( \langle x, y \rangle \) can be formally defined using set theory (see exercise 2.26-9), but we rely on the intuition (for Computer Scientists, an array of two elements the first being \( x \) and the second \( y \)). Tuples are generalizations of pair to \( n \geq 1 \) components, \( \prod_{i=1}^{n} \mathcal{X}_i = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n \triangleq \{\langle x_1, x_2, \ldots, x_n \rangle \mid \forall i \in [0, n] : x_i \in \mathcal{X}_i\} \). \( \prod_{i=1}^{n} \mathcal{X} \) is written \( \mathcal{X}^n \) and extended to \( \mathcal{X}^0 \triangleq \{\emptyset\} \).
2.1.1 Relations

A (binary) relation on sets $X$ and $Y$ is a set $R \in \wp(X \times Y)$ \(^{18}\). So if $x \in X$ and $y \in Y$ either $\langle x, y \rangle \in R$ and we say that $x$ is related/connected to $y$ by $R$ (which we also write $R(x, y)$ or $x \ R \ y$ as in $x \leq y$ or $y \ R^{-1} x$ as in $y \geq x$)) or $\langle x, y \rangle \not\in R$ and we say that $x$ is unrelated/unconnected to $y$ by $R$ (which we also write $\neg R(x, y)$ or $x \not\ R \ y$ as in $x \not\leq y$). When $X = Y$, we say that $R \in \wp(X \times X)$ is a relation on $X$. For example, the identity relation on $X$ is $1_x \triangleq \{ \langle x, x \rangle \mid x \in S \}$.

Example 7. A relation can be also be represented by a graph or a matrix.

$$X = \{1, 2, 3\}$$
$$Y = \{a, b, c\}$$
$$R \in \wp(X \times Y)$$
$$R = \{\langle 1, a \rangle, \langle 2, a \rangle, \langle 2, c \rangle\}$$

Example 8 (Relational semantics of the assignment). Consider a program with two integer variables $x$ and $y$. Their value is a pair of integers $\langle x, y \rangle \in \mathbb{Z}^2$.

(a) The effect of the assignment $y := 1$ is $\{\langle x, y \rangle, \langle x, 1 \rangle\} \mid x, y \in \mathbb{Z}$ relating the values of the variables before and after the assignment and stating that $x$ is left unchanged and $y$ is assigned the value 1.

The effect of the assignment $x := x + 1$ can be formalized by the relation $\{\langle x, y \rangle, \langle x + 1, y \rangle\} \mid x, y \in \mathbb{Z}$ stating that $x$ is incremented while $y$ is left unchanged.

(b) If we consider machine integers between values $\min_int$ and $\max_int$ then the incrementation can be thought of as undefined in case of overflow so the semantics of assignment $x := x + 1$ is $\{\langle x, y \rangle, \langle x + 1, y \rangle\} \mid x \in [\min_int, \max_int] \land y \in [\min_int, \max_int]$.

(c) Most languages follow the hardware convention of using modular arithmetics in which case the semantics of assignment $x := x + 1$ is $\{\langle x, y \rangle, \langle x + 1, y \rangle\} \mid x \in [\min_int, \max_int] \land y \in [\min_int, \max_int] \cup \{\langle \max_int, y \rangle, \langle \min_int, y \rangle\} \mid y \in [\min_int, \max_int]$, at least on machines such that $\max_int + 1 = \max_int$.

Semantics (a) corresponds to the understanding of programs as algorithms; (b) to an implementation that should corresponds to algorithm (a) but where execution is stopped when they differ; (c) corresponds to the understanding of programs as defined by the hardware on which they run. An even more specific semantics would fix the values of $\min_int$ and $\max_int$ for a family of processors (e.g. with hypothesis that $\min_int = -\max_int + 1$) or even a specific processor (with $\max_int = 32767$ for a 16 bits machine).

\(^{18}\)The notion of relation is attributed to De Morgan, Peirce and Frege (De Morgan, 1860; Peirce, 1870; Peirce, 1885; Frege, 1879).
We need a relation since an assignment may not always assign the same value to the variable. An example is the assignment \( x := ? \) of a random natural to the variable \( x \). Its relational semantics \((a)\) is \( \{ \langle x, y \rangle, \langle n, y \rangle \mid x, y \in \mathbb{Z} \land n \in \mathbb{N} \} \).

The domain of a relation \( R \in \wp(\mathcal{X} \times \mathcal{Y}) \) is \( \text{dom}(R) \triangleq \{ x \in \mathcal{X} \mid \exists y \in \mathcal{Y} : \langle x, y \rangle \in R \} \), its codomain, range or image is \( \text{img}(R) \triangleq \{ y \in \mathcal{Y} \mid \exists x \in \mathcal{X} : \langle x, y \rangle \in R \} \), and its field is \( \text{fld}(R) \triangleq \text{dom}(R) \cup \text{img}(R) \).

Example 9. For example 8, we have \( \text{dom}(R) = \{ 1, 2 \} \), \( \text{img}(R) = \{ a, c \} \), and \( \text{fld}(R) = \{ 1, 2, a, c \} \).

The inverse \( R^{-1} \) of a relation \( R \in \wp(\mathcal{X} \times \mathcal{Y}) \) is \( R^{-1} \triangleq \{ \langle x, y \rangle \mid \langle x, y \rangle \in R \} \) so that \((R^{-1})^{-1} = R \).

Example 10. For examples 8 and 9, the inverse relation is

\[
\mathcal{X} = \{ 1, 2, 3 \} \\
\mathcal{Y} = \{ a, b, c \} \\
R^{-1} \in \wp(\mathcal{Y} \times \mathcal{X}) \\
R^{-1} = \{ \langle a, 1 \rangle, \langle a, 2 \rangle, \langle c, 2 \rangle \} \\
\]

The composition of relations \( R_1 \in \wp(\mathcal{X} \times \mathcal{Y}) \) and \( R_2 \in \wp(\mathcal{Y} \times \mathcal{Z}) \) is \( R_1 \circ R_2 \in \wp(\mathcal{X} \times \mathcal{Z}) \) such that \( R_1 \circ R_2 = \{ \langle x, z \rangle \mid \exists y : \langle x, y \rangle \in R_1 \land \langle y, z \rangle \in R_2 \} \).

Example 11. For examples 8 and 10 the compositions \( R \circ R^{-1} \) and \( R^{-1} \circ R \) are

Using these notations we can express that a relation \( R \) on a set \( S \) is reflexive if and only if \( 1_S \subseteq R \), transitive if and only if \( R \circ R \subseteq R \), symmetric if \( R \subseteq R^{-1} \), and antisymmetric if \( R \cap R^{-1} \subseteq 1_S \).

Example 12 (Relational semantics of sequential assignments). The relational semantics of a sequence of assignments is the composition of their respective relational semantics. For example the relational semantics of incrementation \( x := x + 1 \) has been defined as \( S^p[x := x + 1] \triangleq \{ \langle x, y \rangle, \langle x + 1, y \rangle \mid x, y \in \mathbb{Z} \} \) in example 8.(a). The semantics of the sequential composition \( x := x + 1; x := x + 1 \) is then
2.12 (Reflexive) Transitive Closure

Let $R \in \varphi(S \times S)$ be a binary relation on a set $S$. The powers of $R^n$, $n \in \mathbb{N}$ are defined as $R^0 \triangleq 1_S$ is the identity relation on the set $S$, $R^1 = R$, and $R^{n+1} \triangleq R ; R^n$, which, by recurrence on $n$, is also $R^{n+1} \triangleq R^n ; R$, $n \in \mathbb{N}$.

The transitive closure of $R$ is $R^+ \triangleq \bigcup_{n \in \mathbb{N}^+} R^n$ while its reflexive transitive closure is $R^* \triangleq \bigcup_{n \in \mathbb{N}} R^n$ (Frege, 1884).

**Example 13.** For $R = \{(a, b), (b, c), (c, d)\}$ we have $(R^n = \emptyset$, for $n \geq 4)$

\[
\begin{array}{cccccc}
R^0 & R = R^1 & R^2 & R^3 & R^+ & R^* \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet
\end{array}
\]

2.13 Equivalence Relations

A relation $\equiv \in \varphi(X \times X)$ on a set $X$ is an equivalence relation if and only if it is reflexive $(\forall x \in X: x \equiv x)$, symmetric $(\forall x, y \in X: (x \equiv y) \Rightarrow (y \equiv x))$, and transitive $(\forall x, y, z \in X: (x \equiv y \land y \equiv z) \Rightarrow (x \equiv z))$.

\[
\begin{array}{c}
[a] = \{a, b, c\} \\
[d] = \{d, e\} \\
[f] = \{f\}
\end{array}
\]

\[
\begin{array}{c}
X = \{a, b, c, d, e, f\} \\
a \equiv b \equiv c \\
d \equiv e
\end{array}
\]
The equivalence class of \( x \in \mathcal{X} \) is \([x]_\equiv \triangleq \{ y \in \mathcal{X} \mid x \equiv y \}\). The quotient of \( \mathcal{X} \) by the equivalence relation \( \equiv \) is the set \( \mathcal{X}/_\equiv \triangleq \{ [x]_\equiv \mid x \in \mathcal{X} \} \) of all equivalence classes in \( \mathcal{X} \) for the equivalence relation \( \equiv \).

An equivalence class \([x]_\equiv\) can be named after any of its representative \( x \) since \( y, z \in [x]_\equiv \) implies \( x \equiv y \equiv z \) so \([x]_\equiv = [y]_\equiv = [z]_\equiv\). A set of class representatives is a subset \( R \subseteq \mathcal{X} \) of \( \mathcal{X} \) which contains exactly one element from each equivalence class (e.g. \( \{a, d, f\} \) in the above example). So, by definition, the set \( R \) of class representatives must cover all classes i.e. \( \{[x]_\equiv \mid x \in \mathcal{X}\} = \{[x]_\equiv \mid x \in R\} \) and there must be only one representative per class i.e. \( \forall x, y \in R: [x]_\equiv = [y]_\equiv \Rightarrow x = y\).

### 2.14 Partitions

Let \( \equiv \) be an equivalence relation on a set \( \mathcal{X} \). Any element \( x \in \mathcal{X} \) belongs to one equivalence class (by reflexivity) and only one (by symmetry and transitivity) so that \( \{[x]_\equiv \mid x \in \mathcal{X}\} \) is a partition of \( \mathcal{X} \) which blocks are the equivalence classes. By definition this means that blocks are disjoint \( ([x]_\equiv \cap [y]_\equiv \neq \emptyset ) \Rightarrow ([x]_\equiv = [y]_\equiv ) \) and cover \( \mathcal{X} \) that is \( \mathcal{X} = \bigcup [x]_\equiv \mid x \in \mathcal{X} \).

Reciprocally, if \( (B_i \in \wp(\mathcal{X}), i \in \Delta) \) is a partition of \( \mathcal{X} \) so \( i \neq j \Rightarrow B_i \cap B_j = \emptyset \) (disjointness) and \( \bigcup_{i \in \Delta} B_i = \mathcal{X} \) (cover) then the relation on \( \mathcal{X} \) defined by \( (x \equiv y) \triangleq \exists i \in \Delta : x, y \in B_i \) is an equivalence relation.

This correspondence between equivalences and partitions is made precise in theorem 136.

### 2.15 Functions

A function\(^{19}\) (application, correspondence, map, mapping, operator, transformation, etc) \( f \in \mathcal{X} \mapsto \mathcal{Y} \) is a relation between the so-called domain \( \mathcal{X} \) and codomain, range or image \( \mathcal{Y} \) which is functional which means that any element of the domain \( \mathcal{X} \) is related to at most one element of the codomain \( \mathcal{Y} \); \( \forall x, y, z : ((x, y) \in f \land (x, z) \in f) \Rightarrow (y = z) \). The unique element of \( \mathcal{Y} \) which is related to \( x \in \mathcal{X} \) by \( f \), if any, is traditionally written \( f(x) \) and is called the image of \( x \) by \( f \) or the value of \( f \) for \( x \).

The assignment of \( y \in \mathcal{Y} \) to a function \( f \in \mathcal{X} \mapsto \mathcal{Y} \) at point \( x \in \mathcal{X} \) is denoted \( f[x \leftarrow y] \). This is a function \( f[x \leftarrow y] \in \mathcal{X} \mapsto \mathcal{Y} \) everywhere equal to \( f \) but at point \( x \) where it is equal to \( y \). So \( f[x \leftarrow y](z) = f(z) \) for all \( z \in \mathcal{X} \setminus \{x\} \) while \( f[x \leftarrow y](x) = y \).

\(^{19}\) Gottfried Wilhelm von Leibniz is at the origin of the term of “function” in 1692, from Latin “functionem” (nominative “functio”), to achieve, to execute, to perform (von Leibniz, 1673 (published 1694)).
Example 14 (Environment assignment). Continuing example 8, if the program has many (integer) variables, then their state can be formalized by a so-called environment \( \rho \in \mathbb{V} \mapsto \mathbb{Z} \) mapping variables \( x \in \mathbb{V} \) to their integer value \( \rho(x) \in \mathbb{Z} \). The environment \( \rho \) is a function since a variable can have only one value at any time during the computation.

The effect of the constant assignment \( x := 1 \) is then \( \{ \langle \rho, \rho[x \leftarrow 1] \rangle \mid \rho \in \mathbb{V} \mapsto \mathbb{Z} \} \) where \( \rho(y) \) is the value of variables \( y \) before the assignment while \( \rho[x \leftarrow 1] \) describes the value of the variables after the assignment. The new value of \( x \) is 1 since \( \rho[x \leftarrow 1](x) = 1 \) while that of the other variables is left unchanged since \( \rho[x \leftarrow 1](y) = \rho(y) \) when \( y \neq x \).

The assignment of a random natural number \( x := ? \) would be \( \{ \langle \rho, \rho[x \leftarrow n] \rangle \mid \rho \in \mathbb{V} \mapsto \mathbb{Z} \land n \in \mathbb{N} \} \) since \( \rho(x) \) is a relation between the \( \mathbb{V} \) and \( \mathbb{Z} \) and \( n \in \mathbb{N} \) is a constant.

The effect of the assignment \( x := x+1 \) is then \( \{ \langle \rho, \rho[x \leftarrow \rho(x) + 1] \rangle \mid \rho \in \mathbb{V} \mapsto \mathbb{Z} \} \) where \( \rho(y) \) is the value of variables \( y \) before the assignment while \( \rho[x \leftarrow \rho(x) + 1] \) describes the value of the variables after the assignment. The former value \( \rho(x) \) of \( x \) is incremented by 1 and assigned as the new value of \( x \). The value of the other variables is left unchanged. So after the assignment \( \rho[x \leftarrow \rho(x) + 1](x) = \rho(x) + 1 \) while \( \rho[x \leftarrow \rho(x) + 1](y) = \rho(y) \) when \( y \neq x \).

A function \( f \in \mathcal{X} \mapsto \mathcal{Y} \) can always be understood as the functional relation \( f = \{ \langle x, f(x) \rangle \mid x \in \mathcal{X} \} \) called its graph. We often write \( fx \) for \( f(x) \) in particular when \( f \in \mathcal{X} \mapsto (\mathcal{Y} \mapsto (\mathbb{Z} \mapsto \mathbb{R})) \) where \( fxyz \) stands for \( ((f(x))(y))(z) \) where \( fx = f(x) \in \mathcal{Y} \mapsto (\mathbb{Z} \mapsto \mathbb{R}) \) and \( fxyz = ((f(x))(y)) \in \mathbb{Z} \mapsto \mathbb{R} \) so \( fxyz \in \mathbb{R} \).

If \( f \in \mathcal{X} \mapsto \mathcal{Y} \) then we may want to specify the codomain \( \mathcal{Y}(x) \subseteq \mathcal{Y} \) for each argument \( x \) of \( f \) in which case we write \( f \in (x \in \mathcal{X} \mapsto \mathcal{Y}(x)) \) to mean that \( f \in \mathcal{X} \mapsto \bigcup_{x \in \mathcal{X}} \mathcal{Y}(x) \) and \( \forall x \in \mathcal{X} : f(x) \in \mathcal{Y}(x) \). In particular, for the product \( \prod_{i=1}^{n} \mathcal{X}_{i} \in (i \in [1, n] \mapsto \mathcal{X}_{i}) \). The unique empty function \( \bot \) has an empty graph hence is everywhere undefined. When \( \mathcal{Y} = \mathcal{X} \), we often call \( f \in \mathcal{X} \mapsto \mathcal{X} \) an operator or transformer on \( \mathcal{X} \).

We frequently curry functions \( f \in \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{Z} \) into \( f \in \mathcal{X} \mapsto \mathcal{Y} \mapsto \mathcal{Z} \) and uncurry \( f \in \mathcal{X} \mapsto \mathcal{Y} \mapsto \mathcal{Z} \) into \( f \in \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{Z} \) where \( f(x, y) = f(x)y \) so that the currying \( \text{curry}(f)(x, y) \triangleq f(x, y) \) and uncurrying \( \text{uncurry}(f)(x, y) \triangleq f(x)y \) operations are left implicit.
We write $\lambda F A$ set function $e$ where the body $A/l.sc/o.sc C/h.sc/u.sc/r.sc/c.sc/h.sc$.

### 2.16 Function Properties

A set function $F \in \varphi(\mathcal{X}) \mapsto \varphi(\mathcal{Y})$ is said to

- **increasing**: when $\forall X \in \varphi(\mathcal{X}), Y \in \varphi(\mathcal{Y}), : (X \subseteq Y) \Rightarrow (F(X) \subseteq F(Y))$ (i.e. larger parameters yield larger results)\(^{21}\). The set of increasing maps of $\varphi(\mathcal{X})$ into $\varphi(\mathcal{Y})$ is denoted $\varphi(\mathcal{X}) \hookrightarrow \varphi(\mathcal{Y})$;

- **decreasing**: when $\forall X \in \varphi(\mathcal{X}), Y \in \varphi(\mathcal{Y}), : (X \subseteq Y) \Rightarrow (F(X) \supseteq F(Y))$ (i.e. larger parameters yield smaller results)\(^{22}\). The set of decreasing maps of $\varphi(\mathcal{X})$ into $\varphi(\mathcal{Y})$ is denoted $\varphi(\mathcal{X}) \twoheadrightarrow \varphi(\mathcal{Y})$;

\(^{20}\)We often use greek letters, see appendix A.7 for the greek alphabet.

\(^{21}\) Increasing functions are also called covariant, isotone, order-preserving, and, unfortunately, monotone.

\(^{22}\) Decreasing functions are also called antitone, contravariant, or order-inversing.
CHAPTER 2. ELEMENTS OF NÀVE SET THEORY

monotone: when it is either increasing or else decreasing;

extensive: when $\forall X \in \varphi(\mathcal{X}) : X \subseteq F(X)$ \(^{23}\);

reductive: when $\forall X \in \varphi(\mathcal{X}) : F(X) \subseteq X$;

idempotent: when $\forall X \in \varphi(\mathcal{X}) : F(F(X)) = F(X)$;

itemfinite union/join/lub preserving: when $\forall X \in \varphi(\mathcal{X}), Y \in \varphi(\mathcal{Y}) : F(X \cup Y) = F(X) \cup F(Y)$;

(arbitrary) union/join/lub preserving: when $\forall S \in \varphi(\mathcal{X}) : F(\bigcup S) = \bigcup \{F(X) \mid X \in S\}$ \(^{24}\). The set of join preserving maps of $\varphi(\mathcal{X})$ into $\varphi(\mathcal{Y})$ is denoted $\varphi(\mathcal{X}) \rightarrow \varphi(\mathcal{Y})$. The map is said to be (arbitrary) union/join/lub inversing when $\forall S \in \varphi(\mathcal{X}) : F(\bigcup S) = \bigcap \{F(X) \mid X \in S\}$;

finite intersection/meet/glb preserving: when $\forall X \in \varphi(\mathcal{X}), Y \in \varphi(\mathcal{Y}) : F(X \cap Y) = F(X) \cap F(Y)$;

(arbitrary) intersection/meet/glb preserving: when $\forall S \in \varphi(\mathcal{X}) : F(\bigcap S) = \bigcap \{F(X) \mid X \in S\}$ \(^{25}\). The set of meet preserving maps of $\varphi(\mathcal{X})$ into $\varphi(\mathcal{Y})$ is denoted $\varphi(\mathcal{X}) \rightarrow \varphi(\mathcal{Y})$. The map is said to be (arbitrary) intersection/meet/glb inversing when $\forall S \in \varphi(\mathcal{X}) : F(\bigcap S) = \bigcup \{F(X) \mid X \in S\}$;

strict: when $F(\emptyset) = \emptyset$;

coorstrict: when $F(\mathcal{X}) = \mathcal{Y}$.

**Lemma 15.** An arbitrary join preserving map $F \in \varphi(\mathcal{X}) \rightarrow \varphi(\mathcal{Y})$ is strict.

**Proof.** The proof in $\langle \varphi(\mathcal{X}), \subseteq \rangle$ and $\langle \varphi(\mathcal{Y}), \subseteq \rangle$ is

\[
\begin{align*}
F(\emptyset) & = F(\bigcup \emptyset) & \text{\{note that } \emptyset \in \varphi(\mathcal{X})\} \\
& = \bigcup \{F(X) \mid X \in \emptyset\} & \text{\{def. } \bigcup \in \varphi(\mathcal{X}) \rightarrow \varphi(\mathcal{X}) \text{ and } \emptyset \in \varphi(\mathcal{X})\} \\
& = \bigcup \emptyset & \text{\{since } F \in \varphi(\mathcal{X}) \rightarrow \varphi(\mathcal{Y})\} \\
& = \emptyset & \text{\{since } X \in \emptyset \text{ is false and } \{F(X) \mid \text{false}\} = \emptyset\} \\
& \text{\{def. } \bigcup \text{\}}
\end{align*}
\]

\(^{23}\) Extensive functions are also called expanding, and, unfortunately, increasing.

\(^{24}\) that is $F(\bigcup S) = \bigcup F^S(\mathcal{S})$ using the image notation of section 6.1

\(^{25}\) that is $F(\bigcap S) = \bigcap F^S(\mathcal{S})$ using the image notation of section 6.1

33
Remark 16. Notice that in the definition of arbitrary lub/glb preservation $S$ can be empty or infinite while in the finite case $S = \{X, Y\}$ is not empty and always finite by finite composition of binary joins. It follows that an arbitrary lub-preserving has a behavior requirement on the empty set which is not requested for finite lub-preserving maps. For example if $A \in \wp(X) \setminus \{\emptyset\}$ then $F = \lambda X \in \wp(A) \cdot A \cup X$ is finite lub preserving since $A \cup (X \cup Y) = (A \cup X) \cup (A \cup Y)$ which extend to any $\Delta \neq \emptyset$ as $A \cup (\bigcup_{i \in \Delta} X_i) = \bigcup_{i \in \Delta} (A \cup X_i)$. However $F$ is not strict since $A \cup \emptyset = A \neq \emptyset$ so not arbitrary lub-preserving despite the fact that it is arbitrary non-empty lub-preserving.

Remark 17. These notions can be easily extended to functions with multiple parameters by fixing all but one parameter and considering that the function in this parameter has the requested property. For example, a function $f \in X_1 \times \ldots \times X_n \mapsto Y$ in increasing in its $i$-th parameter if and only if for all $x_1 \in X_1, \ldots, x_{i-1} \in X_{i-1}, x_{i+1} \in X_{i+1}, \ldots, x_n \in X_n$, the function $\lambda y \cdot f(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n)$ is increasing (in its unique parameter $y$).

2.17 Injections, Surjections, Bijections

A function $f \in X \mapsto Y$ is said to be injective/one to one when $\forall x, y \in X : (x \neq y) \Rightarrow (f(x) \neq f(y))$ (no two different elements can have same the image) or equivalently $\forall x, y \in X : (f(x) = f(y)) \Rightarrow (x = y)$, to be surjective/onto when $\forall y \in Y : \exists x \in X : f(x) = y$ (every element of the codomain is the image of at least one element of the domain), and to be bijective/one to one-onto/an isomorphism when it is both injective and surjective, on which case $X$ and $Y$ are said to be isomorphic (by $f$).

The inverse of a bijection $f \in X \mapsto Y$ is a singleton which necessarily unique element is called the inverse $f^{-1} \in Y \mapsto X$ of $f$ such that $\forall x \in X : f^{-1}(f(x)) = x$ and $\forall y \in Y : f(f^{-1}(y)) = y$.

2.18 Isomorphisms Between Ordered Structures

An isomorphism between sets need not preserve the partial orders defined on these sets. Here is a counter-example.
CHAPTER 2. ELEMENTS OF NAÏVE SET THEORY

\[ f(\emptyset) = \{0\} \]
\[ f(\{0\}) = \emptyset \]
\[ f(\{1\}) = \{0, 1\} \]
\[ f(\{0, 1\}) = \{1\} \]

An increasing isomorphism between sets need not preserve the partial order structure of these sets. Here is a counter-example.

However, an increasing isomorphism which inverse is also increasing does preserve the partial order structure of these sets, specifically the partial orders are the same, the lubs are the same, and the glbs are the same, up to the isomorphism.

The two order structures are the same up to a re-encoding of the elements of the structure. This will be made mathematically precise in section 8.9 where we study Galois isomorphisms.
2.19 Indexed Families

A family \( \langle x_i, \ i \in I \rangle \) of elements in \( \mathcal{X} \) indexed by a set \( I \) is a function \( x \in I \mapsto \mathcal{X} \) that maps any \( i \in I \) to \( x_i = x(i) \in \mathcal{X} \). We also write \( \langle x_i \in \mathcal{X}, \ i \in I \rangle \) to make clear that \( \forall i \in I : x_i \in \mathcal{X} \) as well as \( \langle x_i \in \mathcal{X}_i, \ i \in I \rangle \) to mean that \( \forall i \in I : x_i \in \mathcal{X}_i \). The class of all such families is written \( \langle \mathcal{X}_i, \ i \in I \rangle \). The index set \( I \) is often chosen to be the natural numbers \( \mathbb{N} \), or, when this is not large enough, ordinals (see chapter ??).

An indexed family \( x \in (\mathcal{X}_i, \ i \in I) \) can be abstracted to a set \( \{ x_i \mid i \in I \} \subset \bigcup_{i \in I} \mathcal{X}_i \) by ignoring duplicates \( x_i = x_j \) with \( i \neq j \). So indexed families allow for duplicates while sets don’t. So we prefer the notation \( \langle x_i, \ i \in I \rangle \) for indexed family rather than the classical but more ambiguous \( \{ x_i \}_{i \in I} \) which can be thought of as referring to a set.

2.20 Iteration

The powers \( f^n \in \mathcal{X} \mapsto \mathcal{X} \), \( n \in \mathbb{N} \) of a function \( f \in \mathcal{X} \mapsto \mathcal{X} \) are the identity \( f^0(x) \triangleq x \) for \( n = 0 \) and \( f^{n+1}(x) \triangleq f(f^n(x)) \) so that \( f^n(x) = f \circ f \circ \ldots \circ f \) (n times) is the application of \( f \) to argument \( x \), \( n \) consecutive times. The iterates of \( f \) from \( x \) are the indexed family \( \langle f^n(x), \ n \in \mathbb{N} \rangle \) that is \( x, f(x), f^2(x), \ldots, f^n(x), \ldots \). If \( \mathcal{X} \) is infinite, there are only three possibilities.

Divergence: \( \forall n \in \mathbb{N} : \forall k < n : f^n(x) \neq f^k(x) \).

Lasso: Otherwise, \( \exists n \in \mathbb{N} : \exists k < n : f^n(x) = f^k(x) \). If \( k < n - 1 \), we have a so-called lasso.

Fixpoint: If \( k = n - 1 \), we have a fixpoint since \( f^{n-1}(x) = f^n(x) = f(f^{n-1}(x)) \).
CHAPTER 2. ELEMENTS OF NAIVE SET THEORY

If \( \mathcal{X} \) is finite, divergence is impossible.

In case of a fixpoint, the iterates are ultimately stationary at the fixpoint \( \forall k \geq n - 1 : f^k(x) = f^{n-1}(x) = f(f^{n-1}(x)) \). So the limit \( \lim_{n \to \infty} f^n(x) \) is clearly this fixpoint \( f^{n-1}(x) = f(f^{n-1}(x)) \). Otherwise, the notion of limit is less clear and application dependent. For example, when \( F \in \mathcal{P}(\mathcal{X}) \mapsto \mathcal{P}(\mathcal{X}) \), we can also define \( F^\omega(\mathcal{X}) \triangleq \lim_{n \to \infty} F^n(\mathcal{X}) \triangleq \bigcup F^n(\mathcal{X}) \) (the limit \( F^\omega \) is also written \( F^\infty \)).

\[2.21\] Duality

In Mathematics, a duality dual translates concepts, theorems or mathematical structures \( x \) into other concepts, theorems or structures \( \text{dual}(x) \), in a one-to-one fashion, often (but not always) by means of an involution operation (\( \text{dual} \) is its own inverse) so that \( \text{dual}(\text{dual}(x)) = x \).

\[2.21.1\] Order-Reversing Duality

We frequently use the order-reversing duality where definitions, lemmatas, theorems, etc. have be established on a set \( \mathcal{P}(\mathcal{S}) \) using \( \subseteq, \emptyset, \cup, \cap, \subseteq \)-smaller, minimal, maximal, etc. and the dual definitions, lemmata, theorems, etc. consists in systematically replacing in these texts the symbols \( \subseteq, \emptyset, \cup, \cap, \subseteq \)-smaller, minimal, maximal, join preserving, meet preserving, etc. by the (order-reversing) duals, that is respectively by \( \supseteq, \mathcal{S}, \emptyset, \cap, \cup, \supseteq \)-smaller (i.e. \( \subseteq \)-greater), maximal, minimal, meet preserving, join preserving, etc. (see e.g. Schröder’s (1890, pp. 166–216) “Dualismus”). Because any definition in set-theory has an order-reversing dual, any proof of a lemma or theorem is also, by duality, a proof of the dual lemma or theorem. For example, the dual of lemma 15 is

**Lemma 18.** An arbitrary meet preserving map \( F \in \mathcal{P}(\mathcal{X}) \mapsto \mathcal{P}(\mathcal{Y}) \) is costrict.

**Proof.** The dual proof is the dual of the proof of lemma 15 hence it is in \( \langle \mathcal{X}, \supseteq \rangle \) and \( \langle \mathcal{Y}, \subseteq \rangle \) but \( \varphi(\mathcal{X}) \), in particular \( \emptyset \in \varphi(\mathcal{X}) \), is unchanged. So \( \bigcup \in \varphi(\mathcal{X}) \mapsto \varphi(\mathcal{X}) \) becomes \( \bigcap \in \varphi(\mathcal{X}) \mapsto \varphi(\mathcal{X}) \) by duality on the result of the operation.

\[
\begin{align*}
F(\mathcal{X}) &= F(\bigcap \emptyset) \\
&= \bigcap\{F(X) \mid X \in \emptyset\} \\
&= \bigcap \emptyset \\
&= \emptyset \\
&= \mathcal{Y}
\end{align*}
\]

\[26\] That is we assume that \( \omega = \infty = +\infty \land \forall n \in \mathbb{N} : n < \omega \).
Duality may have fixpoints *i.e.* dual$(x) = x$ in which case we say that the concept $x$ is *self-dual*. For example, the order-reversing dual of “increasing” for $f \in \wp(\mathcal{X}) \rightarrow \wp(\mathcal{Y})$ is “increasing”.

2.2.1.2 Complement Duality

A particular order reversing duality on $\wp(S)$ originates from complements $\bar{\mathcal{X}} \triangleq S \setminus \mathcal{X}$ since $\mathcal{X} \subseteq \mathcal{Y}$ if and only if $\bar{\mathcal{X}} \supseteq \bar{\mathcal{Y}}$, in which case we speak of *complement duality* where $\bar{\mathcal{X}}$ is the dual of $\mathcal{X}$. Functions are considered dual when $f(\bar{x}) = \bar{g(x)}$ so that $f$ and $\bar{f} \circ \bar{g}$ are complement dual. The join and meet operations are complement dual in this sense since by De Morgan’s laws (De Morgan, 1840) $\bigcap_{i \in \Delta} \mathcal{X}_i = \bigcup_{i \in \Delta} \bar{\mathcal{X}}_i$. This means that for every theorem of set theory there is an equivalent dual theorem (such as $\bar{\bar{\mathcal{X}}} \cap \mathcal{Y} = \bigcup_{i \in \Delta} \bar{\mathcal{X}}_i$ which is $\bigcap_{i \in \Delta} \bar{\mathcal{X}}_i = \bigcup_{i \in \Delta} \mathcal{X}_i$).

2.2.1.3 Semi-Duality

When considering a pair $(\wp(S_1), \wp(S_2))$, we can use duality on both powersets, or only on one of them, in which case we speak of “dualizing”. For example, if $f \in \wp(S_1) \rightarrow \wp(S_2)$ then the semi-dual of “$f$ is increasing” is “$f$ is decreasing” when considering the dual on $\wp(S_1)$ (or on $\wp(S_2)$) only. The semi-dual of “(arbitrary) join/meet preserving” is “(arbitrary) join/meet inversing”.

2.2.2 Pointwise Extension

Let $\mathbb{L}$ be a set. Given $\subseteq, \emptyset, \mathcal{X}, \cup, \cap, etc.$ on $\wp(\mathcal{X})$ their pointwise extensions to $\mathbb{L} \rightarrow \wp(\mathcal{X})$ are $\subseteq^L, \emptyset^L, \mathcal{X}^L, \cup^L, \cap^L$, such that $f \subseteq^L g \iff \forall \ell \in \mathbb{L} : f(\ell) = g(\ell)$, $\emptyset^L \triangleq \mathcal{X} \subseteq \mathbb{L} \cdot \emptyset$, $f \cup^L g \triangleq \mathcal{X} \subseteq \mathbb{L} \cdot f(\ell) \cup g(\ell)$, etc. We freely generalize results of sets to their pointwise extensions without repeating the proof of these results for all $\ell \in \mathbb{L}$. For example $\mathcal{X} \subseteq \mathcal{Y} \Rightarrow \mathcal{X} \cap \mathcal{Y} = \mathcal{X}$ and so $f \subseteq^L g \Rightarrow f \cap^L g = f$. Similarly, we sometimes use double pointwise extensions $\subseteq^L \subseteq^L, \emptyset^L \emptyset^L, \mathcal{X}^L \mathcal{X}^L, \cup^L \cup^L, \cap^L \cap^L$, etc. to $\mathbb{L} \rightarrow \mathbb{L'} \rightarrow \wp(\mathcal{X})$.

2.2.3 Sequences

Given a set $\mathbb{A}$, we let $\mathbb{A}^n \triangleq [0, n - 1] \rightarrow \mathbb{A}$ be the set of non-empty finite sequences $x_0, x_1, \ldots, x_{n-1}, n > 0$, of elements of the set $\mathbb{A}$ that is of functions $x \in [0, n - 1] \rightarrow \mathbb{A}$ such that $\forall i \in [0, n - 1] : x(i) = x_i$. So it is the indexed family $\langle x_i \in \mathbb{A}, i \in [0, n - 1] \rangle$. The non-empty finite sequences over $\mathbb{A}$ are $\mathbb{A}^+ \triangleq \bigcup_{n > 0} \mathbb{A}^n$. For $n = 0$, the empty sequence is $\epsilon \in \emptyset \rightarrow S$. The finite sequences over $\mathbb{A}$ are $\mathbb{A}^* \triangleq \bigcup_{n \in \mathbb{N}} \mathbb{A}^n = \mathbb{A}^+ \cup \{\epsilon\}$. An infinite sequence $x_0, x_1, \ldots, x_n, \ldots$ of elements of $S$ is $x \in \mathbb{N} \rightarrow S$ such that $\forall i \in \mathbb{N} : x(i) = x_i$. 38
CHAPTER 2. ELEMENTS OF NAÏVE SET THEORY

It is the indexed family \( \langle x_i \in S, \ i \in \mathbb{N} \rangle \). The length \( |x| \) of a finite sequence \( x \in A^n \) is \( n \) and that of an infinite sequence is \( \omega \) (such that \( \forall n \in \mathbb{N} : n < \omega \)).

2.24 Characteristic Function

Let \( S \in \mathcal{S} \) be a set of the universe \( \mathcal{U} \) containing all the sets we are interested in (see section 3.4). The characteristic function of \( S \) is \( f_S \in \mathcal{U} \mapsto \mathbb{B} \) such that \( \forall x \in \mathcal{U} : f_S(x) \triangleq (x \in S) \) so that \( S = \{ x \in \mathcal{U} | f_S(x) \} \).

Inversely, a Boolean function \( f \in \mathcal{U} \mapsto \mathbb{B} \) uniquely defines the subset \( S = \{ x \in \mathcal{U} | f(x) \} \). This establishes a bijection between \( \mathcal{S} \) and \( \mathcal{U} \mapsto \mathbb{B} \).

We often call a Boolean-valued function \( P \in \mathcal{U} \mapsto \mathbb{B} \) a predicate and consider it, up to this bijection, as the property \( \{ x \in \mathcal{U} | P(x) \} \).

2.25 Bibliography

Enderton (1972) provides an introduction to first-order logic whereas Andrews (2002) includes higher order logics.

There are many introductory books to naïve set theory, such as (Halmos, 1960; Johnstone, 1987; Levy, 1979; Mendelson, 1964; Monk, 1969). See (Dubreil and Dubreil-Jacotin, 1939; Ore, 1942; Whitman, 1946) for more specific results on equivalence relations. Iteration in section 2.20 originates from the successive approximations of Newton (Newton, 1669; Newton, 1671 (published 1736)), Carl Friedrich Gauss (Gauss, 1823), and Joseph Liouville (Liouville, 1837) to compute fixpoints.

The project of Nicolas Bourbaki was to rigorously formulate (almost) all Mathematics in set theory (Bourbaki, 1939), which can also be the case for Computer Science.

For example, specification languages such as B (Abrial, 1996), Event-B (Abrial, 2010), or the Vienna Development Method VDM (Jones, 1990) can be understood as extensions of Set Theory where objects and proof methods pertinent to Computer Science are rigorously defined and developed into a hopefully clear, full, and explicit presentation.

2.26 Exercises

Exercise 2.26-1 Following (Peirce, 1870), let \( F \) be the set of Frenchmen, \( M \) the set of men, \( A \) the set of animals, \( E \) be a set of emperors, \( C \) be a set of conquerors, \( L \) be a set of lovers, and \( V \) be a set of violinists. Express the following properties in set theory.

(a) “Every Frenchman is a man, but there are men besides Frenchmen” (Peirce, 1870, p. 324);
(b) “Every Frenchman is a man, without saying whether there are any other men or not” (Peirce, 1870, p. 324);
(c) “From every Frenchman being a man and every man being an animal; [we can infer] that every Frenchman is an animal” (Peirce, 1870, p. 324);

(d) “I shall take involution in such a sense that $X^Y$ will denote everything which is an $X$ for every individual of $Y$. Let us write this relation as $X \times Y$ (instead of Peirce’s $X^Y$). Express “That which is emperor or conqueror of every Frenchman” (Peirce, 1870, p. 332)\(^{27}\);

(e) “That which is emperor of every Frenchmen and conqueror of all the rest” (Peirce, 1870, p. 332);

(f) “The things which are lovers to nothing but French violinists are the things that are lovers to nothing but Frenchmen and lovers to nothing but violinists” (Peirce, 1870, p. 354)\(^{28}\);

(g) “To say that a person is both emperor and conqueror of the same Frenchman is the same as to say that, taking any class of Frenchmen whatever, this person is either an emperor of some one of this class, or conqueror of some one among the remaining Frenchmen.” (Peirce, 1870, p. 355).

Exercise 2.26-2  Prove that $\mathcal{X} \subseteq \mathcal{Y} \iff \mathcal{X} \cap \mathcal{Y} = \mathcal{X} \iff \mathcal{X} \cup \mathcal{Y} = \mathcal{Y}$. □

Exercise 2.26-3  Prove that if $X \in \wp(\mathcal{U})$ and $\mathcal{Y} \in \wp(\wp(\mathcal{U}))$ then $X \cap \cup \mathcal{Y} = \cup\{X \cap Y \mid Y \in \mathcal{Y}\}$ and $X \cup \cap \mathcal{Y} = \cap\{X \cup Y \mid Y \in \mathcal{Y}\}$. For example if $\mathcal{Y} = \{Y_1, Y_2\}$ this is $X \cap (Y_1 \cup Y_2) = (X \cap Y_1) \cup (X \cap Y_2)$. □

Exercise 2.26-4  Prove De Morgan law if $\mathcal{Y} \in \wp(\wp(\mathcal{U}))$ then $\neg \cup \mathcal{Y} = \cap\{-Y \mid Y \in \mathcal{Y}\}$ where $\neg X \triangleq \mathcal{U} \setminus X$. Similarly, $\neg \cap \mathcal{Y} = \cup\{-Y \mid Y \in \mathcal{Y}\}$. □

Exercise 2.26-5  Show that a partial function $f \in \mathcal{X} \not\rightarrow \not\mathcal{Y}$ defines an equivalence relation on $\mathcal{X}$. □

Exercise 2.26-6  Given $R_1 \in \wp(\mathcal{X} \times \mathcal{Y})$ and $R_2 \in \wp(\mathcal{Y} \times \mathcal{Z})$, calculate $(R_1 ; R_2)^{-1}$. □

Exercise 2.26-7  Following remark 17, prove that the composition $; \,$ of relations is $\subseteq$-increasing in each of its parameters. □

\(^{27}\) Probably an allusion to Louis-Napoléon Bonaparte, who was the first President of France and next the last Emperor of the French until 4 September 1870.

\(^{28}\) Possibly an allusion to the famous Belgian violin virtuoso Charles Auguste de Bériot, lover of the acclaimed singer, “La Malibran”. 

40
CHAPTER 2. ELEMENTS OF NAÏVE SET THEORY

Exercise 2.26-8  Let \( \mathcal{X} \in \wp(\wp(\mathcal{X})) \) be a subset of the power set of a set \( \mathcal{X} \) such that if \( X, Y \in \mathcal{X} \) then \( X \cup Y \in \mathcal{X} \) (\( (\mathcal{X}, \subseteq) \) is called a poset and \( (\mathcal{X}, \subseteq, \cup) \) a join-lattice).

Prove that \( f : \mathcal{X} \mapsto \mathcal{X} \) is \( \subseteq \)-increasing if and only if for all \( X, Y \in \mathcal{X} \), \( f(X) \cup f(Y) \subseteq f(X \cup Y) \). Express the dual.

In set theory, there is no need for atomic elements since every entity can be defined in terms of the empty set and operations on sets. Here is an example for pairs.

Exercise 2.26-9  Kuratowski (1921) proposed to encode pairs as \( \langle a, b \rangle \triangleq \{\{a\}, \{a, b\} \} \) (where \( a \) and \( b \) are sets). Show that if \( p = \langle a, b \rangle \) then \( a = \bigcup \cap p \) and \( b = \bigcup \{x \in \bigcup p \mid \bigcup p \neq \bigcap p \Rightarrow x \notin \bigcap p \} \).

Exercise 2.26-10  If \( \mathcal{X}, \mathcal{Y} \in \wp(S) \) then prove that \( \mathcal{X} \subseteq \mathcal{Y} \iff \mathcal{X} \cap \mathcal{Y} = \mathcal{X} \iff \mathcal{X} \cup \mathcal{Y} = \mathcal{Y} \iff \neg \mathcal{X} \cup \mathcal{Y} = S \iff \mathcal{X} \cap \neg \mathcal{Y} = \emptyset \).

Exercise 2.26-11  Let \( R, T \in \wp(\mathcal{X} \times \mathcal{X}) \) be binary relations on a set \( \mathcal{X} \) such that \( R \subseteq T \). Prove that if \( T \) is transitive then \( R^+ \subseteq T \).

Exercise 2.26-12  Let \( \langle \mathcal{L}, \sqsubseteq \rangle \) be a poset such that any subset \( X \in \wp(\mathcal{L}) \) has a lub \( \bigcup X \). Prove that this implies that any subset \( X \in \wp(\mathcal{L}) \) has a glb \( \bigcap X \). State the dual proposition.
Chapter 3

Complements on Logic

We have grouped in this short chapter 3 more advanced topics on the logical mathematical foundations that can be used in Computer Science. This chapter 3 is more related to the foundations/philosophy of Mathematics and can be skipped at first reading.

3.1 First and Second-Order Logic

In second order logic (e.g. Wilhelm Ackermann and David Hilbert (Ackermann and Hilbert, 1928, Der engere Prädikatenkalkül, pp. 49–105)), quantification can be on sets (predicates), functions, and relations. For example, von Leibniz’s Law (also called von Leibniz principle of the identity of indiscernibles (von Leibniz, 1686 (published by H. Lestienne 1907), Section IX)) states that two objects that have exactly the same properties are equal ∀x, y : (x = y) ⇔ (∀P : x ∈ P ⇔ y ∈ P) so that they are different if and only if they can be distinguished by some property ∀x, y : (x ≠ y) ⇔ (∃P : (x ∈ P ∧ y ∉ P) ∨ (x ∉ P ∧ y ∈ P)). So the quantifications ∀P : . . . and ∃P : . . . are on all possible properties P of objects.

In first order logic, quantification is restricted to variables that range over a set (e.g. Hermann Weyl (Weyl, 1918), (Ackermann and Hilbert, 1928, Der erweiterte Prädikatenkalkül, pp. 106–153)). For example one cannot write ∀P : ∀x : x ∈ P ∨ x ∉ P in first-order-logic since sets and properties cannot be quantified over.

This makes a difference on what can be expressed and proved with the logic, a point that we illustrate below in the rest of the next section 3.2.

3.2 First and Second-Order Deductive System

A logic must specifying how to make formal proofs (see chapter 5 and exercises ?? and ??) which is formalized by a deductive system. It contains axioms (such as ∀x : x /∈ ∅ in set theory) and inference rules (such as modus ponens ∀P, Q : (P ∧ P ⇒ Q) ⇒ Q written ∀P, Q : \( \frac{P, P \Rightarrow Q}{Q} \))
in second-order logic.

In first-order logic, quantification is not allowed on predicates so than one must use axiom and inference rule schemata, in which one or more schematic/metalinguistic variables \( \phi, \psi, \text{ etc.} \) appear, and can be substituted by any formula of the logic\(^1\). For example the first order logic modus ponens schema is 
\[
\frac{\phi, \phi \Rightarrow \psi}{\psi}
\]
where \( \phi \) and \( \psi \) can be substituted by any first-order formula. This generates countably infinitely many axioms and inference rules, one for each possible instantiation of the schematic/metalinguistic variables \( \phi, \psi, \text{ etc.} \)

### 3.3 Models

It turns out that second-order logic is strictly more expressive than first-order logic, even when using schemata. For example second-order Dedekind/Peano arithmetic (Dedekind, 1893; Peano, 1889) has only one standard model \((0 \cdot 1 \cdot 2 \cdot 3 \cdots)\) whereas first-order Peano arithmetic (where recurrence on naturals is expressed with an inference rule schema) has non-standard models \((0 \cdot 1 \cdot 2 \cdot 3 \cdots -1' \cdot -2' \cdot -3' \cdots ... ... \cdot -2'' \cdot -1'' \cdot 0'' \cdot 1'' \cdot 2'' \cdots ... \)).

So the logic \( \mathcal{L} \) that we use is a second-order logic, which is more expressive, as a formal language for expressing properties of mathematical entities and structures, concepts, and rules of reasoning.

### 3.4 Universe

We rely on the existence of a universe \( \mathcal{U} \) (also called domain of discourse or underlying set) containing all entities required to describe concepts manipulated by computers and their computations, including nonterminating ones.

The universe \( \mathcal{U} \) is assumed to contain

(a) \( A \in \mathcal{U} \) for given atomic elements \( A \triangleq \{ x \in \mathcal{U} : \forall y \in \mathcal{U} : y \notin x \} \) which can be elements of sets but are not sets themselves including the empty set \( \emptyset \), the natural numbers \( \mathbb{N} \), the integers \( \mathbb{Z} \), the rationals \( \mathbb{Q} \), the reals \( \mathbb{R} \), \( -\infty \), \( +\infty \), lexemes and values of programming languages such as \( := \) and \( \text{nil} \), etc\(^2\);

(b) If \( X \in \mathcal{U} \) then \( X \subseteq \mathcal{U} \);

(c) If \( X \in \mathcal{U} \) then \( \wp(X) \in \mathcal{U} \);

---

\(^1\) possibly with syntactic restrictions e.g. on the occurrences of free variables. An example is the Zermalo-Frankel axiom schema of predicative separation \( \forall x : \exists y : \forall z : (z \in y) \leftrightarrow (z \in x \land \phi(z)) \) where \( \phi \) is any formula such that variable \( y \) is not free in \( \phi \), which asserts the existence of a subset of a set if that subset can be defined without reference to the entire universe of sets (thus avoiding a potential circularity).

\(^2\)In set theory atomic elements are called ur-elements but are actually not needed since they can easily be modeled in terms of the empty set and set operations only.
CHAPTER 3. COMPLEMENTS ON LOGIC

(d) If $\Delta \in \mathcal{U}$ and the $X_i \in \mathcal{U}$ are sets for all $i \in \Delta$ then the Cartesian product $\prod_{i \in \Delta} \{X_i \mid i \in \Delta\} \in \mathcal{U}$ and the join $\bigcup_{i \in \Delta} \{X_i \mid i \in \Delta\} \in \mathcal{U}$.

The set of all sets of the universe $\mathcal{U}$ is $\mathcal{S} \triangleq (\mathcal{U} \setminus \mathcal{A}) \cup \{\emptyset\}$.

3.5 Interpretation of the Logic

Contrary to formal mathematical logic, we use second-order logic with the particular interpretation/model/semantics of the formulæ relative to the universe $\mathcal{U}$.

In particular, we always specify the set $\mathcal{S} \in \mathcal{S}$ on which elements are quantified as in $\forall x \in \mathcal{S} : P(x)$ or $\exists x \in \mathcal{S} : P(x)$. When $\mathcal{S}$ is generic, we understand that $\mathcal{S} \in \mathcal{S}$ is any set of the universe $\mathcal{U}$ (so e.g. $\forall x : P(x)$ is $\forall x \in \mathcal{U} : P(x)$). Since this convention can lead to heavy notations, we may omit the set and then we implicitly refer to the universe. For example, $\forall X : \forall X : (X \in X) \Rightarrow (X \subseteq \bigcup X)$ stands for $\forall X \in \mathcal{S} : \forall X \in \mathcal{S} : (X \in X) \Rightarrow (X \subseteq \bigcup X)$.

The usual operations for logical properties have the following set-theoretic interpretation in the set $\mathcal{U}$.

<table>
<thead>
<tr>
<th>Logic</th>
<th>Set theory</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>atomic formulæ of $\mathcal{L}$</td>
<td>$\mathcal{A}$</td>
<td>atoms</td>
</tr>
<tr>
<td>formulæ of $\mathcal{L}$</td>
<td>$\mathcal{S}$</td>
<td>sets</td>
</tr>
<tr>
<td>$\Rightarrow$</td>
<td>$\subseteq$</td>
<td>implication/subset inclusion</td>
</tr>
<tr>
<td>$\text{false}$</td>
<td>$\emptyset$</td>
<td>false/empty set</td>
</tr>
<tr>
<td>$\text{true}$</td>
<td>$\mathcal{U}$</td>
<td>true/whole set</td>
</tr>
<tr>
<td>$\lor$</td>
<td>$\cup$</td>
<td>disjunction/union</td>
</tr>
<tr>
<td>$\land$</td>
<td>$\cap$</td>
<td>conjunction/intersection</td>
</tr>
<tr>
<td>$\neg$</td>
<td>$\neg$</td>
<td>negation/complement ($\neg X = \mathcal{U} \setminus X$)</td>
</tr>
</tbody>
</table>

3 so the universe contains in particular the set of natural numbers $\mathbb{N}$, the set of integers $\mathbb{Z}$, the set of rationals $\mathbb{Q}$, the set of reals $\mathbb{R}$, all subsets of a sets hence its powerset, all relations, function graphs, etc.

4 All our reasonings are within the universe $\mathcal{U}$, which is a set. This avoids problematic concepts and paradoxes in set theory such as the “set” of everything which includes the “set” of all sets, a concept that leads to contradictions as in “the set of all elements that do not belong to that set” $\mathcal{S} = \{x \mid x \notin \mathcal{S}\}$ (which is analogous to the liar paradox that is the statement “this sentence is false.”). The purpose of set theory is to solve these contractions, see e.g. Monk (1969). Then $\mathcal{U}$ can be chosen as any large enough set.

5 so contrary to proof theory (Hilbert, 1928; Pohlers, 2008; Schütte, 1960; Troelstra and Schwichtenberg, 2000), we are not interested in the study of the purely syntactic manipulations of formulæ and deduction methods as in found in interactive theorem provers. Moreover, contrary model theory (Chang and Keisler, 1992), we do not consider all possible models/interpretations/semantics for logics, but a fixed one chosen to be suitable to discuss Computer Science concepts.
\begin{align*}
\forall x : P(x) &\mid \{ x \mid P(x) \} = \mathcal{U} \quad \text{universal quantification} \\
\exists x : P(x) &\mid \{ x \mid P(x) \} \neq \emptyset \quad \text{existential quantification}
\end{align*}

By these definitions $x \in \bigcup \emptyset = \text{false}$ so $\bigcup \emptyset = \emptyset$ and $x \in \bigcap \emptyset = \{ x \in \mathcal{U} \mid \text{true} \} = \mathcal{U}$. Similarly $\bigcap S = \emptyset$ and $\bigcup S = \mathcal{U}$. It follows that $\langle S, \subseteq, \emptyset, \mathcal{U}, \cap, \cup, \neg \rangle$ is a complete Boolean lattice of sets (so where $\neg X = S \setminus X$).

Because our use of logic is for the specific set-theoretic interpretation in the universe $\mathcal{U}$, the two notations are essentially interchangeable and we may use indifferently one or the other, or even a mix of the two notations, depending on what we think is the more “natural” in some specific context (such as $\forall x \in \mathbb{Z} : \exists y \in \mathbb{Z} : x + y = 0$ better than $(\mathcal{U} \setminus \mathbb{Z}) \cup \{ x \mid \neg((\mathbb{Z} \cap \{ y \mid x + y = 0 \}) = \emptyset) \} = \mathcal{U}$).

### 3.6 Bibliography

Examples of possible universes $\mathcal{U}$ are \textsc{Willard Van Orman Quine}’s New Foundations (Forster, 1995), \textsc{John von Neumann}’s hierarchy of sets (von Neumann, 1928), the small sets in \textsc{Samuel Eilenberg} and \textsc{Saunders Mac Lane}’s category theory (Eilenberg and Lane, 1945; Mac Lane, 1998) and \textsc{Alexander Grothendieck}’s universe in toposes (Streicher, 2006). Inside this universe, we use naïve set theory (Halmos, 1960).

(Manna, 1970) anticipated that all interesting properties of computer systems can be expressed in second order logic.

### 3.7 Exercises

**Exercise 3.7-1** When do you think of the definition $S \overset{\Delta}{=} \{ x \in \mathcal{U} \mid x \not\in S \}$?

**Exercise 3.7-2** The definition of the universe $\mathcal{U}$ in section 3.4, does not mention intersections. Prove that if $\Delta \in \mathcal{U}$ and the $X_i \in \mathcal{U}$ are sets for all $i \in \Delta$ then $\bigcap_{i \in \Delta} X_i \in \mathcal{U}$. 

46